

Tactile sensing for Robot Learning: Development to Deployment

Raunaq Mahesh Bhirangi

CMU-RI-TR-24-61

August 2024

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Abhinav Gupta, Co-Chair

Carmel Majidi, Co-Chair

Deepak Pathak

Lerrel Pinto, *New York University*

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Keywords: Tactile Sensing, Robotics, Machine Learning

For the people who raised me and the ones I've found along the way

Abstract

The role of tactile sensing is widely acknowledged for robots interacting with the physical environment. However, few contemporary sensors have gained widespread use among roboticists. This thesis proposes a framework for incorporating tactile sensing into a robot learning paradigm, from development to deployment, through the lens of ReSkin – a versatile and scalable magnetic tactile sensor. By examining design, integration, policy learning and representation learning in the context of ReSkin, this thesis aims to provide guidance on the implementation of effective sensing systems for robot learning.

We begin by proposing ReSkin – a low-cost, compact, and diverse platform for tactile sensing. We develop a self-supervised learning technique that enables sensor replaceability by adapting learned models to generalize to new instances of the sensor. Next, we investigate the scalability of ReSkin in the context of dexterous manipulation: we introduce the *D’Manus*, an inexpensive, modular, and robust platform with integrated large-area ReSkin sensing, aimed at satisfying the large-scale data collection demands of robot learning.

Based on the learnings from the development of ReSkin and the *D’Manus*, we propose AnySkin – an upgraded sensor tailored for robot learning that further reduces variability in response across sensor instances. AnySkin is as easy to integrate as putting on a phone case, eliminates the need for adhesion and demonstrates enhanced signal consistency. We deploy AnySkin in a policy learning setting for precise manipulation, demonstrate improved task performance when augmenting camera information, and exhibit zero-shot transfer of learned policies across sensor instances.

Going beyond sensor design and deployment, we explore representation learning for sensors including but not limited to ReSkin. Sensory data is typically sequential and continuous; however, most research on existing sequential architectures like LSTMs and Transformers focuses primarily on discrete modalities such as text and DNA. To address this gap, we propose Hierarchical State Space (HiSS) models, a conceptually simple and novel technique for continuous sequence-to-sequence prediction (CSP). HiSS creates a temporal hierarchy by stacking structured state-space models on top of each other, and outperforms state-of-the-art sequence models such as causal Transformers, LSTMs, S4, and Mamba. Further, we introduce CSP-Bench, a new benchmark for CSP tasks from real-world sensory data. CSP-Bench aims to address the lack of real-world datasets available for CSP tasks, providing a valuable resource for researchers working in this area.

Finally, we conclude by summarizing our takeaways throughout the journey of ReSkin from development to deployment, and outline promising directions for bringing tactile sensing into the fold of mainstream robotics research.

Acknowledgments

First and foremost, I extend my deepest gratitude to my advisors, both official and unofficial: Abhinav Gupta, Carmel Majidi, Vikash Kumar, Tess Hellebrekers, and Lerrel Pinto. When I began my PhD, I had only a vague idea of my end goal—becoming a full-stack roboticist. If I’ve made any progress toward that ideal, it is thanks to Abhinav, who encouraged me to venture into the relatively unexplored territory of tactile sensing within the context of robot learning. Carmel has been unwavering in his support throughout this journey, guiding me when I worked within his area of expertise and trusting my process as I delved deeper into robot learning. Vikash has been, and continues to be, an inspiration for his profound understanding of every aspect of robotics, from design and development to real-world robot learning. Their support and belief in me have profoundly shaped my research trajectory.

However, this thesis would not exist without the indispensable support of Tess and Lerrel, who have been my advisors in all but name. Tess taught me everything I know about sensors, helped bridge the gap between sensor design and deployment, and consistently removed the many obstacles inherent in hardware development. Beyond research, she has been a pillar of support and guidance, motivating me through the lowest points of my PhD and encouraging me to persevere when I was ready to abandon research altogether. Lerrel adopted me as an advisee when my PhD felt aimless and uninspired, rekindling the passion that drew me to robotics in the first place and restoring my joy in research. He is an inspirational researcher who has profoundly influenced my own research approach. I am eternally grateful to both Tess and Lerrel for being central figures in making this mentally and emotionally challenging endeavor feel rewarding and fulfilling.

Much of the work presented in this thesis would not have been possible without my collaborators—Abigail DeFranco, Jacob Adkins, Venkatesh Pattabiraman, Chenyu Wang, and Siddhant Halder. I want to give a special shout-out to Venkatesh, who has been a consistent partner in crime in numerous exciting research projects during the final year of my PhD and continues to inspire me with his boundless enthusiasm and tireless work ethic.

From the very beginning, my thesis has been an inherently interdisciplinary endeavor, and I have been fortunate to receive invaluable support from four different labs throughout this journey. I owe my deepest gratitude to Zach Patterson, Drew Sabelhaus, Yunsik Ohm, Mason Zadan, Michael Vinciguerra, Dinesh Patel, Nolen Keays, Akhil Padmanabha, and all other members of the Soft Machines Lab. As a complete novice in wet lab experiments, their guidance and assistance at every step have been indispensable. I would also like to extend my heartfelt thanks to Sudeep Dasari, Shikhar Bahl, Yufei Ye, Homanga Bhardhwaj, Helen Jiang, Kenneth Marino, Victoria Dean, Samantha Powers, Jianren Wang, Unnat Jain, Gaoyue Zhou, Kenneth Shaw, Murtaza Dalal, Russell Mendonca, Mohan Kumar, and all other members of the interwoven Gupta and LEAP labs. Their groundbreaking work in robot learning has inspired much of my own research, and their willingness to discuss ideas and provide invaluable feedback has been immensely helpful. During the final phase

of my research this past year, I have been incredibly fortunate to be part of the Generalizable Robotics and AI Lab at NYU. I am deeply grateful to Irmak Guzey, Haritheja Etukuru, Mahi Shafiullah, Jeff Cui, Nikhil Bhattasali, Siddhant Haldar, Ulyana Piterbarg, Ademi Adeniji, Mara Levy, Enes Erciyas, Aadhithya Iyer, Zu Wang, and all other lab members. They have welcomed my wildest ideas, offered insightful feedback, and created a lab environment that I have genuinely looked forward to being in every day.

Meta has played a crucial role in my research journey, providing many of the resources that made my work possible. I am especially grateful for the Meta AI Mentorship program, which has been instrumental in supporting my research during the last two years of my PhD. I would also like to express my heartfelt thanks to Giri Anantharaman, Mustafa Mukadam, Chris Paxton, Priyam Parashar, Arjun Lakshmipathy, Jessica Hodgins, Sehee Min, Jay Vakil, Susan Andrews, Trayce, and all my other friends at FAIR Pittsburgh, whose support and ideas have been invaluable to my research efforts.

No words can fully convey the impact of my parents, Shilpa and Mahesh Bhirangi, on my journey thus far. They have always encouraged me to strive for excellence and have worked tirelessly to open doors for me that I didn't even know existed. This thesis is as much a product of their efforts to provide me with the best possible opportunities as it is of my own. Their unwavering love and support have enabled me to make the most of those opportunities. I am also deeply thankful for the guidance and encouragement of my extended family and community, and the entire Bhirangi clan that has cheered me on every step of the way.

I am endlessly grateful for the incredible friends I've found over the years, who, despite living miles away, always made time for me and helped me stay grounded. Roshail is a friend with whom I share an inexplicable, deep cosmic connection, and I've been incredibly fortunate to be able to rely on his judgment whenever I doubted my own. Tejas has been my partner in crime, joining me in countless mischiefs, embracing my juvenile humor, and sharing in the excitement of our adventures and Civilization VI games. His unwavering support has been a constant in my life. Aarti has been the wise and dependable presence I can always turn to. Aakriti and Basuhi have brought warmth to many holidays, where we've wrapped ourselves in blankets, watched anime, and indulged in delicious food together. Last, but certainly not the least, Vani has been a constant source of joy and personal growth since the day we met. Her unwavering belief in me has lifted me during my lowest moments, allowing me to find the self-compassion I often lacked. Every day, she inspires me with the effort and intention she pours into everything she does, while somehow always showing up for the people she loves.

Living in a foreign country, 12,000 kilometers away from family, was never going to be easy, but I was fortunate to find a wonderful set of friends who made Pittsburgh feel like a second home. Alex Stephens, Ben Freed, and Swapnil Pande have been the best roommates anyone could ask for, making coming home after a long day something I looked forward to every single day. Alex has cooked some of the best meals I've ever had and has always inspired me with his ability to blend

childlike curiosity with exceptional engineering and artistic sensibility to create beautiful scientific projects. Ben, my oldest friend in Pittsburgh, has been my partner in crime at nearly every party and has been the perfect sounding board for discussing everything from science to politics. Swapnil has been one of my closest friends throughout this journey, offering support through the lowest points of my PhD and making the isolation during the pandemic so much more bearable.

In the interest of brevity, I won't elaborate on all the wonderful people who have forever etched Pittsburgh into my heart, but I would like to thank Abhijat Biswas, Abigail Breinfeld, Alex Baikovitz, Ananya Rao, Arkadeep Chaudhury, Ben Eisner, Ben Newman, Ceci Morales, Charvi Rastogi, Dan McGann, Danny Vedova, Daphne Chen, David Neiman, Gokul Swamy, Hans Kumar, Helen Jiang, Jason Zhang, Jonathan Luiten, Mansi Sood, Mateo Guaman Castro, Michelle Zhao, Mrinal Verghese, Naman Gupta, Pragna Mannam, Ravi Pandya, Rohan Deshpande, Shikib Mehri, Sudharshan Suresh, Tanmay Shankar, Tejas Srinivasan, Thomas Weng, Tito Babatunde, Varsha Hari and Vivian Shen. Tanmay's chai deserves a separate shout-out for being the tastiest and most consistent source of caffeine in my life throughout the last four years. I am also deeply grateful to the Jadhav and Pande families for always welcoming me into their homes and treating me to delicious, home-cooked Indian food whenever I missed home.

I spent the last year of my PhD working at NYU in New York, and I was able to avoid the upheaval of moving to a new city thanks to the sweetest roommate, Irmak Guzey. I am incredibly grateful to have quickly found a wonderfully close-knit group of friends in the Hell's Kitchen Gang: Aditya Shankar, Yamini Bansal, Dimitris Kalimeris, Priyank Parikh, and Prithika Vageeswaran.

Finally, I would like to thank the incredible administrative staff at Carnegie Mellon who helped me stay on top of my academic requirements and navigate the complexities of immigration: Suzanne Muth, BJ Fecich, and Jean Harpley. I am also grateful to the, sadly former, RoboOrg for creating many of the fun experiences that enriched my graduate school years.

Contents

1	Introduction	1
1.1	Tactile sensors for Robotics	1
1.2	Deep Learning and Sensors	2
1.3	Multimodal Policy Learning	3
1.4	Thesis Outline	4
2	ReSkin: versatile, replaceable, lasting tactile skins	5
2.1	Introduction	6
2.2	Background	7
2.3	Design and Fabrication	9
2.4	Experimental Setup	10
2.5	Single Sensor Model – Decoding Magnetic Flux to Contact Characteristics	11
2.6	Adapting to New Sensors – MultiSensor Model + Self-supervised Learning	12
2.6.1	Results	13
2.7	ReSkin in Action	15
2.8	Conclusion	16
3	D’Manus: A dexterous hand with large-area sensing	17
3.1	Introduction	18
3.2	Related Work	19
3.2.1	Dexterous Hands and data-driven learning	19
3.2.2	Tactile sensing	19
3.3	Platform and System Details	21
3.3.1	The Hand: Construction and Interfacing	22
3.3.2	Large-area Exteroceptive Sensing: ReSkin	22
3.3.3	Control and Proprioceptive Sensing	22
3.3.4	Software	23
3.4	Experiments	23
3.5	Tactile Perception: Data and Modeling	24
3.5.1	Data Collection	24
3.5.2	Model Learning	26
3.6	Results	26
3.6.1	Dexterity of the <i>D’Manus</i>	26
3.6.2	Tactile Perception: Material Identification	26

3.6.3	Perceptive Generalization: Softness and Texture	27
3.6.4	Tactile Bin Sorting	29
3.6.5	Robustness and Reliability	30
3.7	Conclusions and Limitations	30
4	AnySkin: Tailoring tactile skins for robot learning	31
4.1	Introduction	32
4.2	Related Work	33
4.2.1	Tactile sensing	33
4.2.2	Replaceability for Tactile Sensors	33
4.2.3	Visuotactile Policy Learning	34
4.3	Background	34
4.3.1	ReSkin: replaceable magnetic tactile skins	34
4.3.2	BAKU: transformer architecture for multimodal learning	34
4.4	Fabrication	35
4.4.1	Mold design	35
4.4.2	Magnetic elastomer fabrication	35
4.5	Experiments	36
4.5.1	Experimental Setup	36
4.5.2	Results	38
4.6	Conclusion	39
5	Hierarchical state space models for continuous sequence-to-sequence modeling	41
5.1	Introduction	42
5.2	Related Work	43
5.2.1	Sequence-to-sequence prediction for sensory data	43
5.2.2	Hierarchical Modeling	44
5.2.3	Data for Continuous Sequence Prediction	44
5.3	Background	44
5.3.1	Sequence-to-sequence Prediction	44
5.3.2	Deep State Space Models	45
5.4	CSP-Bench: A Continuous Sequence Prediction Benchmark	45
5.4.1	Touch Datasets	46
5.4.2	Curated Public Datasets	48
5.5	Hierarchical State-Space Models (HiSS)	48
5.5.1	Data Preparation and Sampling	49
5.5.2	Model Architecture	49
5.5.3	Training details	50
5.6	Experiments and Results	50
5.6.1	Performance of Flat models on CSP-Bench	52
5.6.2	Improving CSP Performance with HiSS	52
5.6.3	Does HiSS Simply do Better Downsampling?	52
5.6.4	Effect of Chunk Size on Performance	52
5.6.5	Effect of Sensory Preprocessing on Performance	54

5.6.6	How Does HiSS Perform on Smaller Datasets?	54
5.6.7	Failure on TotalCapture	55
5.7	Conclusion and Limitations	55
6	Conclusion and Future Prospects	57
A	Appendix for Chapter 5	61
A.1	ReSkin fabrication details	61
A.1.1	Circuitry	61
A.1.2	OnRobot Gripper Tips	62
A.2	Model architectures and Training	62
A.2.1	Flat Architectures	62
A.2.2	Hierarchical architectures	62
A.2.3	Hyperparameters	62
A.3	Experimental Setup and Data Collection details	64
A.3.1	ReSkin: Onrobot Gripper on a Kinova JACO Arm	64
A.3.2	Xela: Allegro Hand on a Franka Emika Panda Arm	66
A.4	Ablations	68
A.4.1	Data Preprocessing	68
A.4.2	Smaller Datasets	69
A.5	TotalCapture Preprocessing	70
	Bibliography	71

List of Figures

- 2.1 A) ReSkin is easy to fabricate and the size of a penny, enabling a wide range of applications. B) Robot gripper using tactile feedback from ReSkin sensors to hold a blueberry without squishing it. C) Dog shoe with an embedded ReSkin sensor; (inset) visualization of sensor measurements. D) Contact localization on a new ReSkin sensor using our self-supervised adaptation procedure. E) Contact localization on a ReSkin curated into a fabric sleeve as a 2in x 4in contiguous skin. F) ReSkin sensor as a fingertip sensor to record forces and contacts while folding a dumpling 6
- 2.2 ReSkin is replaceable! 7
- 2.3 A) Experimental setup for data collection with Dobot Magician, ATI Nano 17 (inset), and six sensor boards streaming to a control computer. B) Mold for curing elastomer along with magnet holders. C) Two types of circuit boards – rigid and flexible – designed for ReSkin. 9
- 2.4 Variation in magnetic field over time and across different sensors. Each tick on the x-axis corresponds to a component of the magnetic field measured by the sensor. While the general trends for individual sensors overlap, there is still obvious variation across samples. 11
- 2.5 Model performance with increasing number of interactions 12
- 2.6 Self-supervised adaptation works with lesser adaptation data as well as training data 14

- 3.1 The *D’Manus* – a low-cost, 10 DoF, reliable prehensile hand with ReSkin [13] sensing. 18
- 3.2 **Anatomy of the *D’Manus* hand:** The *D’Manus* is actuated at joint level using Dynamixel XM430-210 smart actuators. ReSkin sensors are integrated with the fingertips and the palm. Each fingertip sensor is comprised of 8 magnetometers while the palm sensor consists of 32 magnetometers for a total of 56 magnetometers. Sensor and motor interfacing components are housed in the core of the hand. 20
- 3.3 Simulated *D’Manus* 23
- 3.4 **Data collection setup:** Tactile data is collected by placing the object on the palm and executing a human-scripted interaction policy for motor babble. 25
- 3.5 **Sample ReSkin data:** Visualization of tactile data from two of the fingers while interacting with the loofah in Fig. 3.4. 25
- 3.6 Model architecture 26

3.7	Material coverings for Material Identification Task: Uncovered, small bubble wrap, large bubble wrap, corrugated cardboard, silicone sponge and combination of materials	27
3.8	Illustration of the <i>D’Manus</i> grasping different objects with a variety of grasps [43, 156]	27
3.9	Datasets used for Softness and Texture Identification Models	28
4.1	AnySkin is easy to integrate with a range of end effectors	33
4.2	BAKU architecture used in our experiments	35
4.3	Fabrication procedure for AnySkin	36
4.4	Experimental setup for visuotactile policy learning consisting of three fixed cameras, one wrist camera, and an AnySkin sensor on one of the gripper tips . . .	37
4.5	Progression of the plug insertion task used in policy learning experiments	37
5.1	CSP-Bench is a publicly accessible benchmark for continuous sequence prediction on real-world sensory data. We show that Hierarchical State Space Models (HiSS) improve over conventional sequence models on sequential sensory prediction tasks.	42
5.2	Hidden Markov Model for a two-sensor system. X is a data-generating process. Sensor, S , and output, Y , are two observable processes.	45
5.3	CSP-Bench is comprised of six datasets. Three datasets – ReSkin Marker Writing, ReSkin Intrinsic Slip and XELA Joystick Control are tactile datasets collected in-house on two different robot setups as demonstrated above. Three other datasets – RoNIN [71], VECtor [48] and TotalCapture [138] are curated open-source datasets.	46
5.4	(Left) Flat SSM directly maps a sensor sequence to an output sequence. (Right) HiSS divides an input sequence into chunks which are processed into <i>chunk features</i> by a low-level SSM. A high-level SSM maps the resulting sequence to an output sequence.	49
A.1	Circuitry	61
A.2	Gripper Tips with ReSkin	62
A.3	Marker Writing Frames (Top): The gripper tips hold the marker and bring it in contact with the paper before the sequence starts. The arm maneuvers the marker to execute eight strokes on the paper. Intrinsic Slip Frames (Middle): The gripper tips hold the box to start the sequence, and slip through the robot workspace with different orientations. Joystick Control Frames (Bottom): After the sequence begins, the hand holds the joystick, controlling its movement through various positions.	64
A.4	Boxes in the Dataset	65
A.5	End-effector Workspace on the Box, & Local Co-ordinate System	66
A.6	Extreme3D Pro Joystick & Co-ordinate System	67

List of Tables

2.1	ReSkin is the only sensor that satisfies all the requirements for learning approaches	8
2.2	The single-sensor baseline performs poorly, failing to capture variability across sensors. Our self-supervised adaptation significantly improves prediction accuracy as well as MSE in xy , F	14
3.1	Cost breakdown for the D’Manus	21
3.2	Operational Details for the D’Manus	23
3.3	The <i>D’Manus</i> can distinguish between different materials purely using tactile feedback (Sec. 3.6.2). Further, models trained for softness and texture classification generalize to interactions with unseen objects (Sec. 3.6.3).	27
3.4	Comparison of models trained using data from different components of the hand.	29
4.1	Policy performance on the plug insertion task with different input sets.	38
5.1	Summary of all the modalities present in CSP-Bench. Modalities used for training are <i>italicized</i> . In addition to the data used for training models, we also release synchronized video and robot kinematics data to facilitate further research in CSP problems.	47
5.2	Comparison of MSE prediction losses for flat and HiSS models on CSP-Bench. Reported numbers are averaged over 5 seeds for the best performing models. MW: Marker Writing, IS: Intrinsic Slip, R: RoNIN, V: VECtor, JC: Joystick Control, TC: TotalCapture	51
5.3	Performance comparison with (a) downsampled inputs, (b) low pass filter on input sequences, and (c) fewer training samples	53
5.4	Effect of chunk size on performance of HiSS models	54
A.1	Hyperparameters for flat architectures	63
A.2	Hyperparameters for low-level models used in hierarchical architectures	63
A.3	Dimensions of Boxes in the Dataset	67
A.4	Comparison of MSE prediction losses for flat and HiSS models on CSP-Bench when passing the input sequences through a low-pass filter. Reported numbers are averaged over 5 seeds for the best performing models. MW: Marker Writing, IS: Intrinsic Slip, JC: Joystick Control, TC: TotalCapture	68

A.5 Comparison of MSE prediction losses for flat and HiSS models on CSP-Bench when using a fraction of the training dataset. Reported numbers are averaged over 5 seeds for the best performing models. MW: Marker Writing, IS: Intrinsic Slip, JC: Joystick Control, TC: TotalCapture 69

Chapter 1

Introduction

Sensing devices play a vital role in enabling robots to comprehend and respond to their surroundings effectively. Among these sensors, tactile sensors are particularly important as they provide robots with a sense of touch, allowing them to handle objects with precision, detect obstacles or hazards, adjust their grip on objects, and manipulate their environment effectively. Despite significant progress in this field over the years, a universally applicable tactile sensing solution for robots remains elusive. This is primarily due to the complex, multi-step processes of developing, integrating, and learning from tactile sensors – each step presenting its own set of challenges. This thesis provides a recipe for addressing these challenges through the lens of ReSkin, a magnetic tactile sensing solution focused on durability, scalability and robustness for robot learning applications.

1.1 Tactile sensors for Robotics

The criticality of tactile feedback to human dexterity [77, 78, 79] has long inspired numerous investigations into robotic tactile sensors since the earliest days of robotics [67, 91]. Over the years, miniaturization and rapid prototyping have expedited the development of tactile sensors relying on a range of transduction technologies [52]. Resistive [128, 147] and piezoresistive [12, 129] sensors measure applied pressure through the change of electrical resistance due to the deformation of a material between two electrodes. Capacitive sensors [53, 133] similarly rely on conditioning circuits that measure the change in capacitance resulting from the deformation of the sensor in order to capture the interaction characteristics. More recently, optical sensors [40, 90, 146, 152] that use a camera in conjunction with an elastomeric material, and capture physical interactions with the environment through a series of images of the deforming elastomer, have emerged as a high resolution alternative for tactile sensing. Other solutions use MEMS devices [105, 132] and piezoelectric materials [36, 159] as transduction mechanisms for recording physical contact information.

However, many of these sensing solutions suffer numerous drawbacks that preclude their adoption as commodity sensors for robotics. With the exception of optical sensors, each sensing technology requires direct electrical connections between the circuitry and soft elastomer. While the integration of soft elastomers has enhanced tactile sensors across the board by improving

contact conformity, this unintended coupling increases costs and complicates integration. Optical sensors overcome this pitfall by separating the sensing electronics (camera) from the sensing interface (elastomer), but require a clear line-of-sight between camera and elastomer severely restricting their form factor and increasing design complexity. Elastomeric interfaces, by virtue of being soft, degrade much faster than the associated electronics and need to be frequently replaced. However, replaceability and consistency in sensor response are characteristics that are seldom discussed in the context of soft sensors. Furthermore, the complex fabrication procedures associated with soft sensors make them difficult to manufacture at scale and increase variability in response across sensor instances.

In light of these shortcomings, the focus of this thesis is primarily on tactile sensing using magnetic elastomers [69, 70]. The use of magnetic transduction allows ReSkin circuitry that measures the signal to be completely independent and detached from the magnetic elastomer that serves as the sensing interface. This affords our sensors a range of advantages from low cost and scalability to variable form factors enabling sensorization of surfaces of diverse shapes and sizes. The simplicity and repeatability of our fabrication process further reduces the variability in sensor response across different instance of the magnetic elastomer skins, minimizing disruption due to replacement of the elastomer and strengthening our case as a general purpose tactile sensor for robotics.

Tactile Sensing and Robot Hands

Analogous to tactile sensing, the versatility of the human hand has inspired long-standing efforts to emulate its capabilities with a robotic hand [10, 89, 103]. The complexity of building these devices results in most contemporary solutions such as the Shadow Hand [82, 139] and the Allegro Hand (Wonik Robotics) being extremely expensive ($> \$25,000$), brittle, and difficult to repair. These pitfalls are at odds with the demands of data-driven robotics, the emerging class of algorithms for robot control, that rely on large amounts of data, and in turn hardware that is inexpensive and robust to the vagaries of large-scale data collection. Solutions like the LEAP Hand [124] and the Trifinger Hand [151] have sought to plug this gap by creating inexpensive, versatile and easy to assemble robotic hands. However, while tactile sensing has been widely acknowledged to be central to human dexterity [78, 79], none of these solutions provide a scalable integration of tactile sensing at a reasonable price point ($< \$50,000$). The *D'Manus* – an open-sourced hand with integrated large-area sensing fills this crucial void in the robot hand landscape. Furthermore, the hand is fully 3D-printable, has a palm to aid dexterity unlike [2, 151], critical adduction and abduction capabilities unlike the Allegro, and is at least $10\times$ less expensive than most commercial alternatives.

1.2 Deep Learning and Sensors

Most real world control systems, such as wind turbine condition monitoring [130], MRI recognition [84] and inertial odometry [4, 98], often process noisy sensory data to deduce environmental states. Conventional sensor response modeling largely relies on analytical techniques to model the relationship between the raw measured quantity (such as resistance, capacitance, magnetic

flux) and the quantity of interest (force, torque, inertial measurements) [96, 120]. While analytical modeling is useful in mapping the measured transduction to interpretable quantities such as force or contact location, it is often cumbersome and/or requires restrictive assumptions that do not fully model the behavior of the sensor [61]. Computational techniques like Finite Element Analysis [97], while effective, can be extremely slow and preclude real time use of the sensors they model. Advances in rapid prototyping and manufacturing technology have fuelled an increase in the pace and diversity of sensor development, as well as a demand for indirect modeling techniques that enable real-time deployment of these sensors. Machine Learning has emerged as a viable solution to this problem, by enabling implicit sensor modeling without the need to explicitly model the complex physical phenomena driving the transduction mechanism [32, 75].

However, as recent research in deep learning for vision and language has shown impressive capabilities across tasks involving these modalities [1, 41], capable ML models for sensory data are few and far between [86, 154]. Deep learning solutions that do show promising results on sensory data continue to be sensor-specific [71, 153] studies. This is the result of a catch-22 in sensor learning: lack of consolidated, labelled datasets of sensory data, the resulting lack of research in neural architectures that are good at processing sensory data, and therefore, a lack of an understanding of the extent of capabilities of sensory systems that would inform and prompt the collection of more data. To tackle this problem, we propose a two-part solution: CSP-Bench – a benchmark of six sensory datasets for continuous sequence prediction, and Hierarchical State Space Models – a neural architecture adept at sequential reasoning over continuous sensory data, based on incorporating temporal hierarchies into structured state space models like S4 and Mamba. We draw from success stories in vision and language [85] have demonstrated the importance of prudent neural architecture choices and inductive biases in learning effective representations for learning-based reasoning. We show that on six sensory prediction tasks across three different sensors, HiSS outperforms conventional sequence modeling architectures like causal Transformers, LSTMs, S4 and Mamba.

1.3 Multimodal Policy Learning

As roboticists tackle problems of robots operating in unstructured environments, robot learning, especially with the advent of deep learning, has emerged as an especially promising solution. Impressive capabilities for grasping [161], manipulation of articulated objects [42, 107] as well as bimanual manipulation [162] have been made possible by integrating cutting edge neural architectures [68, 117, 142] for vision with advances in density estimation [30, 93] and imitation learning algorithms. However, keeping in line with the consistent theme in this chapter, robot learning models that incorporate tactile sensing are conspicuously scarce. Analysis for learning precise, complex skills that require reasoning about physical interactions with the environment are largely restricted to simulation [28, 92] with little discussion on transferring policies to the real world. Alternative approaches often involves convoluted, unrealistic camera setups to get around the lack of tactile sensing [3, 5]. In this thesis, we perform a controlled study of multimodal policy learning with vision and tactile sensory data and cross-validate the criticality and effectiveness of all the available modalities for learning effective robot policies for precise manipulation.

1.4 Thesis Outline

The rest of this thesis document is structured as follows: Chapter 2 introduces ReSkin – a magnetic tactile skin, its capabilities, and the potential of learned sensor models, Chapter 3 introduces the D’Manus: an open-source dexterous hand design with integrated large-area sensing, Chapter 4 introduces an upgraded self-adhering sensor skin design and demonstrates replaceability in policy learning, while Chapter 5 talks about a new benchmark and a novel learning architecture for sequential modeling of tactile and other sensory data. We highlight takeaways and future prospects in Chapter 6.

Chapter 2

ReSkin: versatile, replaceable, lasting tactile skins

Bhirangi, R., Hellebrekers, T., Majidi, C., & Gupta, A. (2021, October). Reskin: versatile, replaceable, lasting tactile skins. In 5th Annual Conference on Robot Learning.

TH designed the procedure for fabricating ReSkin. RB was responsible for the fabrication of skins, setting up experiments, and training prediction models. RB and TH were jointly responsible for experiment design and analysis of results.

Abstract

Soft sensors have continued growing interest because they enable both passive conformal contact and provide active contact data from the sensor properties. However, the same properties of conformal contact result in faster deterioration of soft sensors and larger variations in their response characteristics over time and across samples, inhibiting their ability to be long-lasting and replaceable. ReSkin is a tactile soft sensor that leverages machine learning and magnetic sensing to offer a low-cost, diverse and compact solution for long-term use. Magnetic sensing separates the electronic circuitry from the passive interface, making it easier to replace interfaces as they wear out while allowing for a wide variety of form factors. Machine learning allows us to learn sensor response models that are robust to variations across fabrication and time, and our self-supervised learning algorithm enables finer performance enhancement with small, inexpensive data collection procedures. We believe that ReSkin opens the door to more versatile, scalable and inexpensive tactile sensation modules than existing alternatives. Videos of experiments and fabrication, code and learned models can be found on <https://reskin.dev>.

2.1 Introduction

In recent years, AI has advanced significantly from large-scale recognition to defeating human players in games. But surprisingly current approaches still struggle at one task: dexterous manipulation. While babies, from a young age, can perform several challenging manipulation tasks, robots continue to struggle even with simple tasks. Why is that? We believe a significant bottleneck in dexterous manipulation is the lack of practical solutions to tactile sensing. From collecting large-scale rich contact data in the wild for learning models to building individual tactile sensors for robot fingers and hand surfaces, current tactile sensing solutions lack on multiple dimensions and fail to scale up.

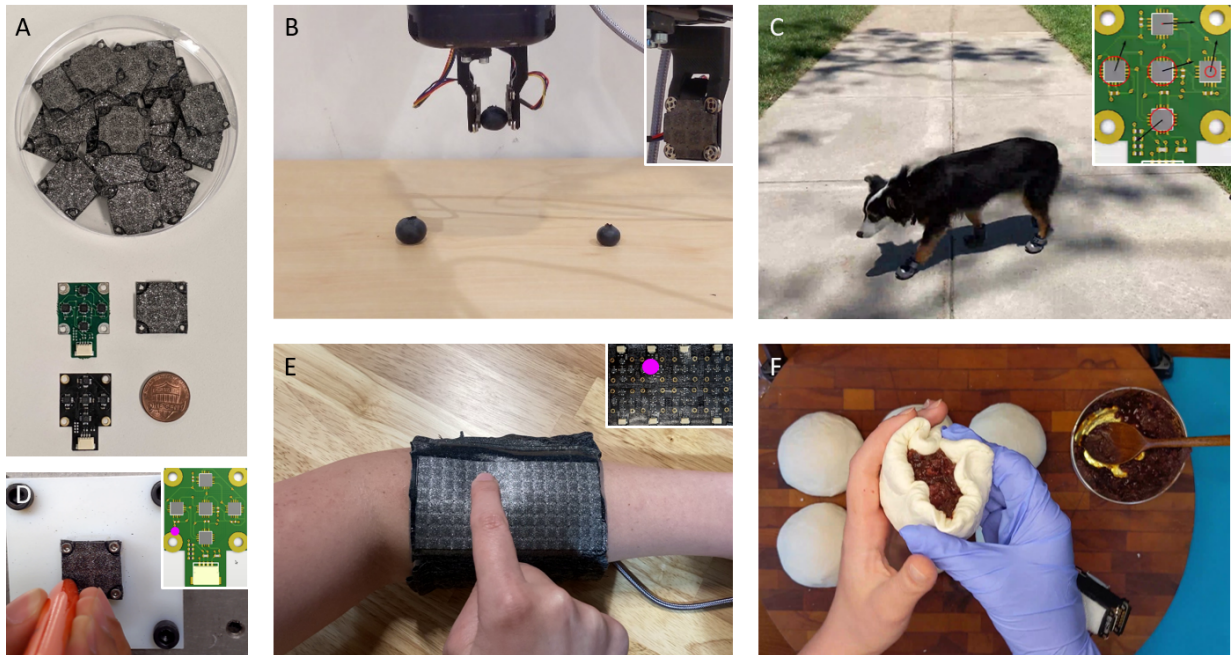


Figure 2.1: A) ReSkin is easy to fabricate and the size of a penny, enabling a wide range of applications. B) Robot gripper using tactile feedback from ReSkin sensors to hold a blueberry without squishing it. C) Dog shoe with an embedded ReSkin sensor; (inset) visualization of sensor measurements. D) Contact localization on a new ReSkin sensor using our self-supervised adaptation procedure. E) Contact localization on a ReSkin curated into a fabric sleeve as a 2in x 4in contiguous skin. F) ReSkin sensor as a fingertip sensor to record forces and contacts while folding a dumpling

In the context of robotics and AI, good tactile skins aim to provide: (a) conformal contact for stable grasping/manipulation; (b) accurate compression and shear force measurements; (c) high force (<0.1 N) and temporal resolution (>100 Hz); and (d) large surface area coverage (>4 cm²) with good spatial resolution for sensing at all contact points. For practical usage, good tactile sensors should also prioritize being (e) compact and versatile, (f) inexpensive, and (g) long-lasting. Current solutions for tactile sensing have not been able to address all of these needs. For example, vision-based tactile sensors are often bulky, expensive, and slow to respond (30-60 Hz) [90, 95]. Resistive and capacitive soft sensors require many connections that lead to early failure and

integration challenges [7, 145]. Commercial sensing options, such as BioTac, are expensive (>\$1000) and available in limited form factors. Rigid tactile sensors, such as force-sensitive resistors, lack the soft, deformable surface that is advantageous for object/environment interaction. Above all, while there has been a plethora of work focused on fingertip sensing, all-over sensing skins are much less studied.

There are two primary reasons why sensing skins have not been practical solutions for tactile sensing: (a) first, there is a direct trade-off between the soft materials that enable conformal contact and their ability to perform well over time. The exact properties that make soft sensors ideal for dexterous manipulation, make them degrade easily during robotic tasks; (b) but more importantly, even skins with durable lasting materials require data-driven modeling which generally fails to generalize from one sensor to another. Therefore, any replacement of skin requires relearning the model which is impractical (hence, limiting experiments to one sensor only [70]).

We propose ReSkin – an inexpensive (<\$30), replaceable, compact, versatile and long-lasting tactile soft skin. ReSkin is composed of soft magnetized skin and a flexible magnetometer-based sensing mechanism. Any deformation of the skin caused by normal/shear forces is read via distortions in magnetic fields. These distortions can be mapped back to estimate the contact points and forces on the original skin using a learned machine learning model. The ReSkin design is compact (2-3mm thick) and long-lasting (our ML models perform accurate predictions even beyond 50K interactions). ReSkin is versatile – the skin and the sensor mechanism can be used anywhere from robot hands to objects to gloves, arm sleeves and even dog paws. ReSkin has high temporal (up to 400Hz) and spatial resolution (1mm with 90% accuracy). But what makes ReSkin the ideal tactile skin is the ability to replace an old skin with a new skin as if you are peeling off an old band-aid and putting a new one on. Our learned models perform strongly even on new skins out-of-the-box but can be further adapted to high precision and resolution using a self-supervised calibration technique. We believe ReSkin has the ability to collect contact data in the wild, provide robust tactile perception capabilities to our robot (See Figure 2.1) and effectively make tactile perception a first-class citizen among its peers (pixels and sound).

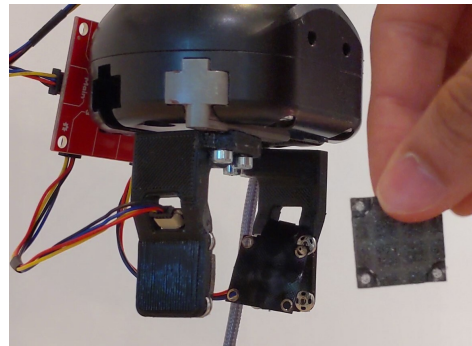


Figure 2.2: ReSkin is replaceable!

2.2 Background

Soft sensing skins provide tactile or proprioceptive information without affecting the underlying mechanics of the system. Machine learning approaches have recently been shown to effectively parse soft sensor data for robotic grasping [18, 19, 34], proprioception [141], and object classification [33], among many others [32, 125, 155]. Recent work in this area has relied on all types of sensing principles to collect and infer information in all types of form factors [17, 66, 160]. For example, resistive networks patterned onto a knit glove have been used to collect tactile grasping dataset and classify objects using neural networks [133]. Capacitive soft skins can be scaled up to

larger areas by sampling at different interrogation frequencies to detect position via SVMs with just one interface [128]. Multi-modal sensing skins have also been shown to improve the ability to discern between 8 types of applied deformation using neural networks [81] and static or dynamic inputs [111]. Data-driven approaches are becoming more common over traditional modeling due to the unpredictable material properties that introduce both non-stationary responses and non-linear behaviors over time, especially considering the dynamic interactions and unconstrained environments robots may encounter.

Unlike capacitive[128], resistive[133], and piezoelectric[12] soft sensing, magnetic and optical soft sensors do not require direct electrical connections between the circuitry and elastomer. This is ideal to keep cost down, as the elastomeric interface degrades much faster than the accessory electronics. It also simplifies the replacement process by not requiring the user to disconnect and reconnect individual wires. While optical sensors provide high spatial resolution data, they also require a clear line of sight between the camera and elastomer to observe deformations[40, 146]. The camera’s depth-of-focus puts a hard limit on the minimum distance to the elastomer surface leading to relatively bulky sensor modules. In contrast, magnetic sensing benefits greatly from minimizing the distance between sensor and elastomer, allowing for a much more compact tactile sensor. In addition, the small form factor of magnetometers, as compared to cameras, enables compatibility across more diverse form factors for the tactile sensor. We demonstrate these key benefits by integrating the magnetic skin onto an arm sleeve, glove, dog shoe and a robot end-effector. In each case, the elastomer is removable while the circuitry stays in place. While there have been a number of attempts towards developing a large area skin - piezoresistive fabrics [12, 17, 143], rigid taxels [29] as well as optical sensors[140], they often lack shear sensing capabilities[12, 29, 143], conformal contact[29] and/or a scalable fabrication process[12]. ReSkin, however, is uniquely positioned to satisfy all of these requirements, and has the potential to be a scalable solution for all-over sensing skin.

	ReSkin	DIGIT	GelSlim	BioTac	RSkin
Type	Magnetic	Optical	Optical	MEMS	Piezoresistive
Frequency	400Hz	60 Hz	60 Hz	100 Hz	?
Variable Form Factor	✓	✗	✗	✗	✓
Thickness <3mm	✓	✗	✗	✗	✓
Low Cost	✓	✓	✓	✗	✗
Easily replaceable	✓	✓	✓	?	✗
Area coverage	✓	✗	✗	✗	✓
Durable (>50k contacts)	✓	?	✗	✓	?

Table 2.1: ReSkin is the only sensor that satisfies all the requirements for learning approaches

The underlying principle for magnetic sensing is that an applied deformation is measured as a change in magnetic flux readings by nearby magnetometers. However, we still need to learn or estimate the mapping function that decodes change in magnetic flux into contact force position and magnitude. Several works on soft sensors have used neural networks for sensor characterizations [63, 136], but these models are often trained on single sensor prototypes, and do not necessarily transfer to new copies of the sensor. Then, the end-user is required to collect and sometimes

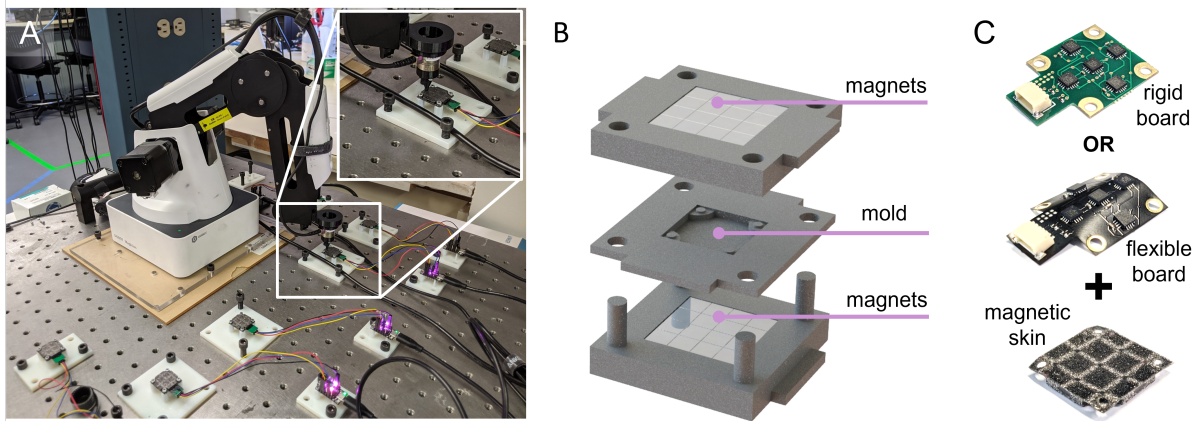


Figure 2.3: A) Experimental setup for data collection with Dobot Magician, ATI Nano 17 (inset), and six sensor boards streaming to a control computer. B) Mold for curing elastomer along with magnet holders. C) Two types of circuit boards – rigid and flexible – designed for ReSkin.

label their own data for each sensor, which additionally requires access expensive, specialized equipment [63, 80].

In this paper, we systematically perform an experimental analysis of the proposed magnetic tactile sensor. First, we extensively study the characterization of a single sensor over time. We demonstrate that one can learn a quite accurate data-driven model to map magnetic flux changes to contact force location and magnitude. We also demonstrate that the skins are long-lasting. However, models trained on one sensor fail to generalize to other sensors or to different circuit board designs. Our first insight is to exploit multi-sensor learning: learn a more generalizable model by using data from a larger number of sensors. While this leads to significant improvement, it still falls well short of training and testing on the same sensor. Inspired by recent work in self-supervised learning, we also present a simple self-supervised calibration procedure which learns to adapt the multi-sensor model to a particular sensor using just a couple of hundred pokes on the skin. Our self-supervised approach is inspired from several works in slow feature learning [55, 76] and contrastive learning [27, 121, 147].

2.3 Design and Fabrication

The sensing principle for ReSkin relies on relative distance changes between embedded magnetic microparticles in an elastomer matrix and a nearby magnetometer. The use of magnetic microparticles allows the skin to be molded into many shapes and thicknesses. When the magnetic composite is deformed by applied force, the magnetometer reports changes in magnetic flux in its X-, Y-, and Z- coordinate system [69]. For an overall sensing area of 20mm x 20mm (Figure 2.3), we measure magnetic flux changes using 5 magnetometers. Four magnetometers (MLX90393; Melexis) are spaced 7mm apart around a central magnetometer. All 3D-printed molds, circuit board files, bill of materials, and libraries used have been publicly released and opensourced on the website.

Magnetic Elastomer Fabrication. Our fabrication technique takes advantage of strong edge-effects of permanent magnets. We magnetize the composite over a 4x4 grid of much smaller cube magnets (0.25in; AppliedMagnets). Additionally, we apply a more uniform magnetic field during curing by placing the grid of magnets both above and below the sample (Figure 2.3B). Magnetic microparticles (MQP-15-7 (~ 80 mesh); Magnequench) and 2-part polymer (Dragonskin 10 NV; Smooth-On) are mixed in a 2:1:1 ratio and poured into a 3D-printed mold. The mold is placed in a vacuum chamber for 3 minutes to remove air bubbles before placing magnets above and below the sample. The sample cures at room temperature (approx 24°C) for at least 3 hours before being removed.

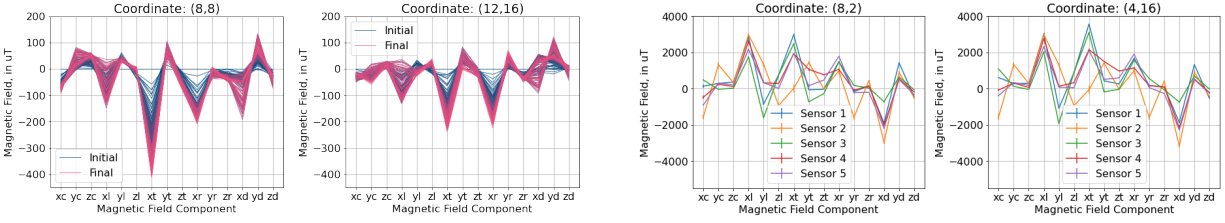
Circuit Board. We use two variants of the circuit board for the experiments and demos presented in this paper – a rigid board and a flexible board illustrated in Figure 2.3C. The circuit includes a 4-pin connector (JST-SH; Molex) that transfers 20 values of magnetometer data (Temp, B_X , B_Y , B_Z for 5 chips) to a microcontroller (Trinket M0; Adafruit) at approximately 400 Hz. The microcontroller processes and transmits this data over USB to be read over serial from a central computer. To allow for easy replacement without damaging the board, we avoid the use of permanent adhesive for attaching the sensing skin. Skins are secured using four screws (M2-6; for the rigid boards), or using four sewable snaps (size 4/0, Dritz; for the flexible boards). The bottoms of the rigid boards are first insulated by applying a thin layer of polymer (nitrocellulose).

2.4 Experimental Setup

Our goal is to perform experimental analysis of the proposed sensor, ReSkin, and the learned mapping between the magnetic field measurements from the sensor (\mathbf{B}) and the planar location ($\mathbf{x} = (x, y)$) and magnitude of the applied force (\mathbf{F}). We want to analyze how ReSkin performs on the different desired attributes – the accuracy of contact force estimation, spatial resolution of contact prediction, robustness to wear and tear and how model performance varies across different skin instances. For all the experiments, we use the data collection apparatus shown in Figure 2.3. The circuit board along with the skin is fixed to a 3D printed mount and streams 4 values, (Temp, B_X , B_Y , B_Z), measurements for each of the five magnetometers at 400 Hz. A hemispherical indenter fastened to the end of a Dobot Magician robot is used to apply forces at different locations on the skin. The indenter also encases a 3-axis F/T sensor (Nano17; ATI) that streams force data at 1kHz.

We restrict ourselves to quasi-static measurements and analysis for the results presented in the following sections, unless stated otherwise. The world coordinate frame used in these experiments is defined such that the xy-plane is aligned with the base of the robot as shown in Figure 2.3. To collect data, we first specify a location for the robot to move to such that the indenter makes contact with the skin. We then record five measurements from the sensor board. The specified xy-location is used as the ground truth label for the location of the force. The normal force measured by the Nano17 is used as the ground truth label for the magnitude of the applied force.

The indentations are made in a snake-like pattern along a 9x9 grid (excluding 4 points per corner; a total of 65 indentations) of size 16cm x 16cm shown in Figure 2.3. During each iteration, we do a single pass at each of 6 depths from 0.2mm to 1.2mm, for a total of 390 indentations. We collect data over multiple iterations.



(a) Variation of magnetic field over time for a single sensor at two coordinates, over 10,000 indentations. The first measurement is subtracted from other measurements to better illustrate degree of variation.

(b) Variation in magnetic field at two different points across five different sensors. Each line corresponds to the average magnetic field measured over 10,000 indentations for a particular sensor

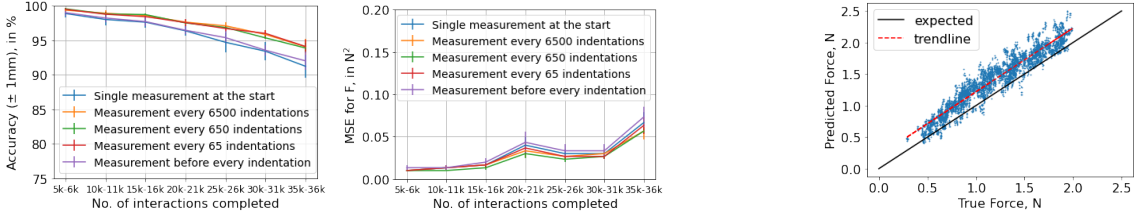
Figure 2.4: Variation in magnetic field over time and across different sensors. Each tick on the x-axis corresponds to a component of the magnetic field measured by the sensor. While the general trends for individual sensors overlap, there is still obvious variation across samples.

2.5 Single Sensor Model – Decoding Magnetic Flux to Contact Characteristics

Our first experiment is to evaluate the accuracy of the mapping from magnetic flux \mathbf{B} to contact force location and magnitude prediction. Our five-layer multilayer perceptron (MLP) architecture for the mapping function is: $\mathbf{B}(15) \rightarrow \text{MLP}+\text{ReLU}(200) \rightarrow \text{MLP}(200) \rightarrow \text{MLP}(40) \rightarrow \text{MLP}+\text{ReLU}(200) \rightarrow \text{MLP}+\text{ReLU}(200) \rightarrow xyF(3)$. The change in magnetic field resulting from deformation is used as the input to our model. The third activation layer is the bottleneck feature layer. We use $feat(\cdot)$ to represent this 3-layer feature extraction network. Our loss function is L2-loss on (x, y, F) . We define the accuracy of contact localization as the fraction of points whose x and y predictions are both within ± 1 mm of their true label. We collect a total of 50K samples and use a random 45K for training and 5K for test. On this simple experiment, we get MSE for location and force is $0.037 \pm 0.014 \text{ mm}^2$ and $0.005 \pm 0.002 \text{ N}^2$ respectively, with a contact localization accuracy of $99.58 \pm 0.34\%$.

To demonstrate the shear sensing capability of the sensor, we perform another experiment. Instead of the quasi-static setup explained in Sec. 2.4, data is collected dynamically by indenting the skin to a certain depth and dragging it along the length of the sensor. We move in straight lines along x and y directions at intervals of 2 mm to cover the entire area of the sensor. The network architecture is the same as described in the previous paragraph, predicting (x, y, F_x, F_y, F_z) instead of just (x, y, F_z) . On this experiment, we get MSE for F_{xy} : $0.0011 \pm 0.0002 \text{ N}^2$, without compromising on prediction of normal force (MSE: $0.003 \pm 0.001 \text{ N}^2$) or contact location (MSE: $0.085 \pm 0.006 \text{ mm}^2$).

Of course, the above setup is not realistic since training and test data is unlikely be sampled randomly from the same distribution. Instead, we use a more practical setting, training on an initial K samples and testing on the samples that come after. As the elastomer goes through multiple cycles of compression and retraction, we see a drift in the properties of the elastomer. This is evidenced by the variation in the recorded magnetic field shown in Figure 2.4a. Therefore, it is critical to analyze how the learned model behaves with respect to time.



(a) Accuracy of contact localization and MSE for force predictions, as the number of interactions with the skin increases

(b) Scatter plot of predicted force and the actual force applied, after 45,000 interactions.

Figure 2.5: Model performance with increasing number of interactions

Since we learn a sensor model that uses the change in magnetic field as input, we would need to record the magnetic field before and after contact occurs. Depending on the application scenario in which the sensor is deployed, it may often be easier to collect calibrating no-load magnetic field measurements at regular intervals. Here, we design an experiment to quantify the effect of the frequency of this measurement on learned sensor models. We collect data from 50,000 indentations on a single sensor. We train a neural network to predict contact location and force, using the first 5,000 indentations as the training set. This model is then evaluated on test sets comprising 1000 indentations after every 5000 indentations, to understand the degree and rate of domain shift. The results of this experiment are shown in Figure 2.5.

Based on Figure 2.5a, we observe that prediction errors are higher and increase faster when we only make a single no-load measurement at the start. This can be attributed to the drift in the elastomer response over time, which can be offset to a certain extent by more frequent no-load measurements. Errors are also higher when no-load measurements are updated before every contact. This could be a result of overfitting to the training data, since later experiments were seen to significantly benefit from updating zero measurements just before every contact. Furthermore, Figure 2.5b indicates that the model, on average, overestimates the applied force. This overestimation can be attributed to the softening of the elastomer as the number of interactions increases.

2.6 Adapting to New Sensors – MultiSensor Model + Self-supervised Learning

Our goal is to provide a simple, replaceable tactile sensor. To achieve this, it is imperative that any learned models acting on sensor measurements generalize to new sensor boards and skins. We demonstrate the generalizability of our learned sensor response model in the following sections. Our models predict the contact normal force (F) and location($\mathbf{x} = (x, y)$) using the change in magnetic field measured by the magnetometers(\mathbf{B}). However, the cheap and easy fabrication method for ReSkin comes with a significant degree of variability in the sensor response. Figure 2.4b demonstrates the variation in raw magnetic field resulting from an identical indentation across different sensors. So, how can we learn a model that generalizes to new sensors and even

new circuit boards?

We use two techniques to help improve generalization for new skins and PCBs. First, instead of using data from a single sensor, we use data from multiple sensors to train our mapping function. This allows the model to see more diverse data in training and learn a more generalizable mapping function. Additionally, we apply a feature regularization component (self-supervised loss) to our loss function. This component is a triplet loss computed in feature space as follows:

$$\mathcal{L}_{\text{triplet}} = \max(0, \|feat(\mathbf{B}_a) - feat(\mathbf{B}_p)\|^2 - \|feat(\mathbf{B}_a) - feat(\mathbf{B}_n)\|^2), \quad (2.1)$$

where \mathbf{B}_a , \mathbf{B}_p and \mathbf{B}_n are three datapoints with corresponding contact locations \mathbf{x}_a , \mathbf{x}_p and \mathbf{x}_n , such that $\|\mathbf{x}_a - \mathbf{x}_p\| < \|\mathbf{x}_a - \mathbf{x}_n\|$, ie. \mathbf{x}_a is closer to \mathbf{x}_p than \mathbf{x}_n . Subscripts a , p and n refer to anchor, positive and negative samples respectively. This loss encourages points that are closer on the skin to be closer to each other in feature space. It acts as a regularizer while also enabling us to use the self-supervised adaptation procedure described in the following paragraph.

Note that this self-supervised loss does not require ground-truth contact location or force readings and therefore can be leveraged to further improve performance on new sensor boards and skins. A new user can collect their own unlabeled dataset, which can be indexed without requiring explicit labels. For instance, the user can use the tip of a pen to indent the sensor skin in a straight line and incrementally index these points as they move along the line. Triplets of points can now be sampled along this line, and the indices can be used to order the pairs within each triplet by distance. Our multi-sensor learned model can then be fine-tuned using these triplets to minimize the triplet loss. At every training step, we sample a batch from the original training data, and an equal-sized batch of triplets (sampled with replacement) from the unlabeled dataset. The former is used to minimize the original loss function, while the latter is only used to minimize the triplet loss.

2.6.1 Results

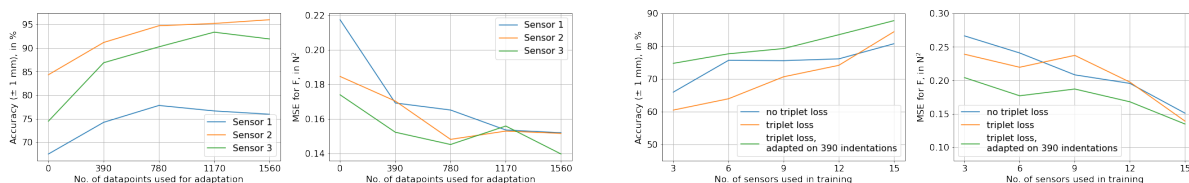
We compare four model approaches. Our baseline model is trained on one sensor and tested on a different sensor. The other three are multi-sensor models – (a) trained without the triplet loss, (b) trained with the triplet loss, and (c) trained with the triplet loss along with self-supervised adaptation.

For the following comparisons, we collect data of 10,000 indentations each from 18 different skins. We use a set of 6 sensor boards and 18 skins: each board appearing thrice in the dataset, each time with a different skin on top. For the multi-sensor models, we perform 6-fold cross-validation, with the held out test set corresponding to three different skins on a particular sensor board each time. For the self-supervised adaptation, we use different subsets of the unseen sensor data in the adaptation step, to qualitatively illustrate the effect of the quantity of data used for adaptation. For the single-sensor model, we individually train on 3 different sensors and test on 9 other sensors.

Based on Table 2.2, we see that the multi-sensor models do significantly better than the single-sensor model. Training on a larger set of sensors allows the neural network model to generalize better. Further, we see that adding the triplet loss slightly affects performance. This small drop could be attributed to the additional constraint on the feature space resulting from the triplet loss. However, the self-supervised adaptation gives us a sizeable improvement over the

Model	Accuracy, in %	MSE_{xy} , in mm^2	MSE_F , in N^2
Single-sensor	25.24 ± 10.12	6.453 ± 3.363	0.420 ± 0.149
Multi-sensor without triplet loss	84.43 ± 12.88	0.733 ± 0.707	0.155 ± 0.025
Multi-sensor with triplet loss	81.03 ± 12.86	0.756 ± 0.718	0.155 ± 0.030
Multi-sensor with triplet loss, adapted using 390 indentations	87.00 ± 11.81	0.514 ± 0.601	0.142 ± 0.025

Table 2.2: The single-sensor baseline performs poorly, failing to capture variability across sensors. Our self-supervised adaptation significantly improves prediction accuracy as well as MSE in xy , F



(a) Self-supervised adaptation improves significantly even with small quantities of adaptation data

(b) Self-supervised adaptation leads to even larger performance gain with fewer sensors in training data

Figure 2.6: Self-supervised adaptation works with lesser adaptation data as well as training data

model predicting without adaptation. Note that the adaptation procedure also results in improved force prediction performance.

To further investigate the effectiveness of our self-supervised adaptation procedure, we look at how increasing the quantity of data used for adaptation correlates with model performance. As can be seen from Figure 2.6a, there is a significant improvement in performance between no self-supervision and using 780 points for self-supervision. However, performance seems to plateau as we increase the data beyond this point, which indicates that our model adapts quickly with small amounts of unlabeled data. Finally, Figure 2.6b shows the effect of the number of sensors in the training data on model performance. We use data from three unseen sensors as the test set. As expected, performance improves as the amount of training data increases. Further, we observe that the self-supervised adaptation, while always doing better than the other models, offers a significant improvement in performance when using fewer sensors in training.

Generalizing models with manual indentations Since the analysis of our self-supervised adaptation technique is performed in a very controlled scenario we perform another, less controlled, experiment which is likely to be closer to the end user’s application setting. We manually indent the sensor 325 times and run self-supervised adaptation using the triplet loss. To test the effectiveness of the adaptation, we collect test data using the original experimental setup to evaluate the adapted models. After adaptation we see improvements in accuracy, from 79.86% to 84.84%, MSE_{xy} from 0.676 mm^2 to 0.489 mm^2 and MSE_F from 0.268 N^2 to 0.192 N^2 , clearly demonstrating the effectiveness of our proposed method even outside a controlled environment.

Generalizing models to a different sensor board type In order to demonstrate the effectiveness of our self-supervised adaptation scheme, we now adapt a model learned using rigid sensor boards

on a flexible sensor board. Note that this is a significantly harder adaptation problem since the distance between skin and circuit board is 80% lesser in case of the flexible board. We see an average contact localization accuracy of 75% with MSE error on location and force as 0.72 mm^2 and 0.54 N^2 respectively. The relatively larger force errors can be attributed to an overestimation bias resulting from signals appearing stronger due to the reduced thickness of the flexible board.

2.7 ReSkin in Action

We have demonstrated that ReSkin is a sensor capable of high resolution contact localization and force prediction. The performance does not deteriorate significantly with wear and tear. But most importantly, learned models generalize to new skins with a simple self-supervised adaptation scheme. We now highlight how ReSkin’s compact design allows it to be used in diverse applications with different form factors. In this section, we use ReSkin in different settings to emphasize its effectiveness in a range of application scenarios. These demos are for proof-of-concept only. For the following, we fabricated both flexible and rigid circuit boards of the exact same design: flexible boards are more comfortable and thinner, while rigid boards can withstand larger applied forces.

Force Sensitivity: Water in shot glass To visually illustrate the force sensitivity of our sensor, we do a pouring demo where we place a shot glass on top of a ReSkin sensor. As the water fills up, we see monotonically increasing sensor measurements indicating the sensor’s ability to distinguish forces as small as the weight of less than 20 mL ($< 0.2\text{N}$) of water (Supplementary Video on <https://reskin.dev>).

Robot Gripper. Next, we show that ReSkin can be a useful tactile sensor for robotics applications such as grasping delicate objects such as blueberries and grapes (See Figure 2.1B). Two ReSkin sensors with flexible circuit boards are placed on either side of a parallel jaw gripper (Robotiq Hand-E Gripper on Sawyer Arm). Grasping soft and squishy objects requires force feedback – applying too much force will squish blueberry and the grape. We show that the built-in force sensing (30N minimum) is insufficient for the task, whereas ReSkin does an excellent job of using force feedback to control grasping. Furthermore, we demonstrate that the grasping continues to work well, with no tuning required, when we replace one of the skins with a new skin (Supplementary Video on <https://reskin.dev>).

Location Sensitivity: Poking To visually illustrate the location sensitivity of our sensor, we do a simple poking task on a new sensor and show the resolution of real-time contact location estimation (See Figure 2.1D and Supplementary Video on <https://reskin.dev>).

Dog Shoe. Our next application demonstrates how ReSkin’s compact design makes it non-obtrusive and useful for measuring tactile forces in the wild. One magnetic skin and flexible circuit board is placed inside the sole of a dog shoe (size: 1.75in). A 1/16in layer of urethane foam is added on top for comfort. The data is collected on-board and logged to an SD card at 250 Hz. The sensorized shoe is worn on the front right leg of a small dog (17 lb). The sensor tracks magnitude and direction of applied force while resting, walking, and running (See Figure 2.1C and Supplementary Video on <https://reskin.dev>).

Glove. We also demonstrate how ReSkin can be used to measure forces during natural human-object interactions. A magnetic skin and rigid circuit board is placed on the right-hand index finger.

A nitrile glove was placed over to hold the board in place, and keep the objects clean. The data was collected on-board and logged to an SD card at 250 Hz. We demonstrate sensor output during the sealing of dough (See Figure 2.1F and Supplementary Video on <https://reskin.dev>). **Arm Sleeve.** Finally, we want to demonstrate that ReSkin is a surface sensor and can be used for wide coverage tactile sensing. Specifically, we connected 8 flexible boards in two rows of four and fabricated a larger, continuous skin (2in x 4in). All 8 boards are connected to a microcontroller (QT Py; Adafruit) that samples all 40 magnetometers at 133 Hz. We show how ReSkin can be scaled up for contact localization across larger surface areas (See Figure 2.1E Supplementary Video on <https://reskin.dev>).

2.8 Conclusion

We present ReSkin: a low-cost, compact and long-lasting surface tactile sensor with high localization accuracy and force sensitivity. ReSkin combines soft sensing with recent advances in machine learning to develop models that generalize across time and individual skins. More specifically, we use multi-sensor learning combined with self-supervised triplet loss for slow feature changes. We also present an SSL adaptation procedure to further refine the models for new skins. Therefore, ReSkin sensors have easily replaceable skin (as easy as peeling and putting new band-aid) that can be used right away. We demonstrate that the compact form of ReSkin makes it an ideal candidate for diverse applications: from grasping delicate objects to measuring forces exerted by dog feet; from building wide-coverage contiguous skin to measuring contact forces in the wild.

Limitations and Future Work: While we have shown promising results on contact localization and force prediction, there is enormous untapped potential for ReSkin at this stage. Experiments in this paper are based on single point contact, and we aim to further investigate multi-point contact. An interesting direction for future work is to analyze the effect of external magnetic fields and metallic objects on ReSkin’s sensing ability. ReSkin can stream data up to 400 Hz and we aim to leverage this capability to train better models using dynamic time-series data. We believe that ReSkin (and its desirable properties) will make tactile perception far more accessible for real-world use.

Acknowledgments

The authors would like to thank Sudeep Dasari for his help with setting up the Sawyer robot and Yunsik Ohm and Zach Patterson for their help with fabrication in the initial stages of the project. The authors would also like to thank everyone at AGI Labs as well as the Soft Machines Lab for their constant help and support. CM was supported in part by NSF-NRI #1830362.

Chapter 3

D'Manus: A dexterous hand with large-area sensing

Bhirangi, R., DeFranco, A., Adkins, J., Majidi, C., Gupta, A., Hellebrekers, T., & Kumar, V. (2023). All the Feels: A dexterous hand with large-area tactile sensing. *IEEE Robotics and Automation Letters*.

AD and JA were responsible for running experiments. RB was responsible for building on VK's design of the D'Manus to integrate tactile sensing and improve dexterity. RB also designed and analyzed the experiments shown in the paper.

Abstract

High cost and lack of reliability have precluded the widespread adoption of dexterous hands in robotics. Furthermore, the lack of a viable tactile sensor capable of sensing over the entire area of the hand impedes the rich, low-level feedback that would improve the learning of dexterous manipulation skills. This paper introduces an inexpensive, modular, and robust platform - the *D'Manus* - aimed at resolving these challenges while satisfying the large-scale data collection demands of deep robot learning paradigms. Studies on human manipulation point to the criticality of low-level tactile feedback in performing everyday dexterous tasks. The *D'Manus* comes with ReSkin sensing on the entire surface of the palm as well as the fingertips. We also demonstrate the generalizability of tactile models trained with the fully integrated system in a tactile-aware task - bin-picking and sorting. Code, documentation, design files, detailed assembly instructions, trained models, task videos, and all supplementary materials required to recreate the setup can be found on <https://sites.google.com/view/dmanus>.



Figure 3.1: The *D'Manus* – a low-cost, 10 DoF, reliable prehensile hand with ReSkin [13] sensing.

3.1 Introduction

Humans routinely operate in unstructured, cluttered environments through surprisingly imprecise, improvised motions. Think about finding the keys hiding at the bottom of your bag, pulling a box from the back of the fridge, or finding the steel ladle among the wooden spatulas. While you rely on vision to plan motion at a high level, executing low-level actions involves using a wealth of tactile signals to spatially understand and characterize the environment. The tactile information, combined with natural compliance and underlying motion, enables the effortless dexterity of the human hand. In moving towards robots with human-like sensorimotor abilities, there is a clear need for systems that integrate rich tactile sensing capabilities with dexterous motion.

However, the high dimensionality of dexterous systems also makes them difficult to control. Data-driven methods have emerged as promising approaches to high-dimensional control [15, 114], but success with dexterous manipulators has been limited [5, 64], and often restricted to simulation [25, 119]. The contact-rich nature of tasks like in-hand manipulation and tool use makes it difficult for policies learned in simulation to generalize to the real world. Collecting data from real-world interactions, on the other hand, is difficult due to the absence of an affordable, reliable hand that can handle the demands of large-scale data collection. Efforts aimed at developing such hardware have been few and far between [2, 151], largely due to the manufacturing cost and the lack of reliable sensing and actuation technologies.

In this work, we leverage recent advancements in rapid prototyping, modular actuation and large-area sensing to present a hand that can make real-world dexterous learning accessible to a wider community of researchers and roboticists. Concretely, our contributions are as follows:

- We present the *D'Manus* – an inexpensive, robust prehensile hand geared towards real-world robot learning, complete with a detailed Mujoco-based simulation model for ease of development and prototyping. We rigorously test the hand to withstand long(>400) hours

of operation with no breakages;

- We equip the *D’Manus* with customized, integrated ReSkin [13, 69] sensors that provide large-area tactile sensing over the entire surface of the palm and the fingertips, while maintaining sharp fingertips/nails critical for dexterous manipulation;
- We demonstrate the caliber of the *D’Manus* along sensory effectiveness, dexterity, and robustness axes by learning tactile perceptive models for softness and texture identification, and validate their generalizability to unseen objects in a tactile-aware bin sorting task.

3.2 Related Work

3.2.1 Dexterous Hands and data-driven learning

The versatility of the human hand has long inspired a number of efforts aimed at creating similarly capable robotic hands dating back to the early days of robotics [10, 103]. Concurrent work in prosthetics and assistive robotics [89] has often overlapped with and contributed to research in creating general-purpose robotic hands. More recently, advances in material science and rapid prototyping as well as control algorithms have further pushed the envelope of capable dexterous hands [113]. Since these efforts have primarily been directed towards demonstrating added functionality on human control, they tend to fall short on the scalability, reliability, affordability, and other capabilities required for the prolonged operation demands of robot learning. Despite the recent advancements in data-driven robotics [114], robust dexterous platforms capable of meeting the data needs of real-world learning have been few and far in between [2, 151]. This has restricted recent investigations with dexterous hands to simulation [25, 119] or the few researchers who can afford the hardware expense [5]. The *D’Manus* is an open-sourced hand that fills a crucial void in the robot hand landscape – integrated large-area sensing and a palm unlike [2, 151], critical adduction and abduction capabilities unlike the Allegro, 30× less expensive than alternatives like [45, 82], and tested to be robust and easy to fix.

Additionally, most recent works aimed at solving dexterous manipulation [5, 110, 163] conspicuously use a single exteroceptive sensory modality – vision. Vision provides rich sensory information about the scene and the visual properties of objects, and has been successfully integrated with robot learning frameworks [15, 114]. However, dexterous tasks are generally contact-rich and require reasoning about contact information that cannot be captured entirely using vision. We posit that the lack of rich tactile information limits a manipulator’s ability to effectively perform real-world dexterous manipulation tasks involving force control, flexible objects, and deformable media, particularly with smaller objects that receive degraded visual signals due to occlusions. The *D’Manus* comes with integrated large-area sensing that offers a rich tactile sensory modality and extensive spatial coverage suitable for learning such contact-rich manipulation skills.

3.2.2 Tactile sensing

The modality of touch has a long history in robotic grasping and manipulation [125]. Several different modalities like capacitive [20, 73], resistive [150], piezoelectric [35], magnetic [69, 137],

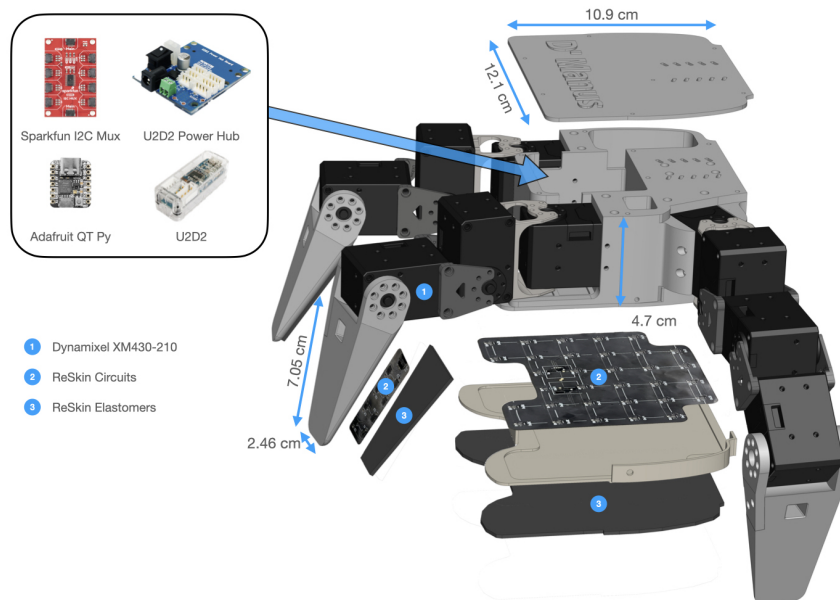


Figure 3.2: **Anatomy of the *D'Manus* hand:** The *D'Manus* is actuated at joint level using Dynamixel XM430-210 smart actuators. ReSkin sensors are integrated with the fingertips and the palm. Each fingertip sensor is comprised of 8 magnetometers while the palm sensor consists of 32 magnetometers for a total of 56 magnetometers. Sensor and motor interfacing components are housed in the core of the hand.

audio [47] and MEMS [112]-based sensors have been explored as tactile sensing alternatives for robotics. With the recent success of deep learning, especially in computer vision, optical tactile sensors [90, 160] have emerged as the popular choice of tactile sensor, due to their high resolution as well as their compatibility with popular neural architectures (CNNs) for processing signals. Most of these solutions, however, have limitations that significantly impede their ability to serve as effective sensors for capable hands, which have strict requirements in terms of sensing, space, cost and robustness. Some of these sensors are bulky [90, 160] or need direct electrical connections between the circuitry and the interface [35, 150], resulting in design constraints that compromise on the manipulation abilities of the hand. Some others are either expensive [150] or difficult to fabricate [20, 73] and cannot be easily replaced, making them less suitable for large-scale data collection given the inevitable wear-and-tear that comes from frequent contact with a wide variety of objects. Yet other alternatives that are affordable and have suitable form factors for dexterous hands tend to lack the resolution, shear sensing [108, 112] required for fine-grained control. Manufacturing, cost, and reliability challenges only escalate with larger area sensing systems, such as the MIT Glove [133], hex-o-skin [108], and uSkin [45, 46] among others [20, 73, 112].

The class of works that come closest to our proposition are [45, 46] that use *uSkin* sensors to sensorize a dexterous hand and demonstrate application in object classification and manipulation tasks. However, *uSkin* uses macro-scale magnets embedded in elastomer as the sensing interface, which involves complex design to avoid crosstalk between magnetometers [137], is bulky, and creates an external magnetic field which can interfere with the environment. To counteract all

Component	Cost
ReSkin Circuits	
Boards	\$ 38.50
Assembly	\$ 149.60
Parts	\$ 56.00
Magnetic Microparticles	\$ 5.50
Smooth-On DragonSkin-10 NV	\$ 5.50
3D printed components	\$ 50.00
Machined components	\$ 200.00
ReSkin Interfacing	\$ 20.45
Dynamixel Interfacing	\$ 51.10
Dynamixel XM430-210 motors	\$ 2899.00
Total	\$ 3475.65

Table 3.1: Cost breakdown for the D’Manus

of these problems, we turn to ReSkin [13, 69], which differs critically in the use of magnetic microparticles instead of macro-sized magnets. ReSkin offers the *D’Manus* a number of key advantages as a dexterous hand for robot learning, namely, (a) favorable form factor: ReSkin can be much thinner ($\sim 2\text{mm}$) than its closest alternatives ($>5\text{mm}$) enabling sharp fingernails critical to dexterous manipulation, (b) cost and replaceability: ReSkin is easily replaceable [13] and costs 50x lesser per sensor ($\sim \$20$) than alternatives like uSkin ($\sim \$1000$), and (c) wear resistance: the absence of a hard-soft interface within the elastomer significantly improves the durability of ReSkin [13] and, as a result, the *D’Manus*.

3.3 Platform and System Details

The *D’Manus* - a combination of Dynamixel and Manus, the Latin word for hand - is a low-cost, reliable prehensile robotic hand with immersive tactile sensing over its larger contact surfaces, i.e. the palm and fingertips as anatomized in Fig. 3.2. To benefit the community and facilitate adoption, *D’Manus* is released as an open-sourced manipulation platform – CAD models, bill of materials, circuit designs, assembly and setup instructions can be found on <https://sites.google.com/view/dmanus>. In this section, we detail the features and properties of the system.

3.3.1 The Hand: Construction and Interfacing

The *D’Manus* hand is a three-fingered, 10-DoF hand – each finger has three degrees of freedom, with a fourth DoF for the thumb. We select 3D printed parts and commercial actuators which allows the *D’Manus* to be easily customized and assembled while maintaining a low price point(\$3500), as detailed in Table 3.1. The hand can be made compatible to be mounted on any robot arm or wrist attachment of choice using a simple 3D printed adaptor. In converging on the design of the *D’Manus*, we focus on three critical components: a palm, an opposable thumb and modular fingers, while ensuring that the fingertips can still perform a precision grasp. While we experimented with versions of the platform with up to 16 DoFs, we converged on the 10 DoF *D’Manus* as it strikes a balance between dexterity, cost, robustness, weight, and size.

3.3.2 Large-area Exteroceptive Sensing: ReSkin

We use ReSkin [13, 69] to endow the hand with large-area exteroceptive tactile sensing. A ReSkin sensor is comprised of a magnetometer circuit in conjunction with a magnetic elastomer. Contact results in deformation of the elastomer which in turn results in a change in magnetic field that is picked up by the magnetometers. Drawing from [13], we scale the sensor circuits and the skins to the size of the palm and the fingertips while maintaining a thickness of 2mm for the skins. Each fingertip sensor is comprised of 8 magnetometers, while the palm sensor consists of 32 magnetometers. The signal from each magnetometer is the 3-axis magnetic flux density. Where our approach deviates significantly from [13] is an improved fabrication procedure for the magnetic elastomer skins used in this work. The skins are cured at room temperature without interfering magnetic fields, and then magnetized using a pulse magnetizer with a 4 Tesla (40 kOe) impulse. This change results in two improvements: (a) stronger signal strength: experiments with the circuits presented in [13] showed at least 5-6x stronger signal along each axis for the same deformation, and (b) ease and scalability of fabrication: magnetic grids scale poorly as the size of the skin increases, making them difficult to assemble in grids as well as to pull apart post-curing. Data from the sensors is streamed to the control computer via USB through a microcontroller + I2C mux. Fig. 3.2 illustrates the construction of the hand and how it integrates with the tactile sensors. A highlight of this design is also the large sensorized area of the palm ($\sim 11 \text{ cm} \times 12 \text{ cm}$) which facilitates stable power grasps and provides a base with force feedback for objects during in-hand manipulation tasks.

3.3.3 Control and Proprioceptive Sensing

To enable closed loop manipulation strategies with strong sensory feedback, the *D’Manus* also comes with a range of proprioceptive sensing capabilities at the actuated joints, as listed in Table 3.2. Control strategies for manipulators lie on a spectrum between position/velocity control and force control. When interaction forces are negligible, position control enables more precise control of the end effector, while velocity control allows for smoother movements. On the other hand, constraints in the environment and frequent interaction forces lend themselves better to force control or “compliant” strategies [102]. The use of Dynamixel smart actuators afford the

Property	Options
Control	Position, Velocity, Current, PWM
Proprioceptive Sensing	Position, Velocity, Current, Realtime tick, Trajectory, Input Voltage
Exteroceptive Sensing	ReSkin (30 Hz)
Limits	Position, Velocity, PWM, Current
Baudrate	9600 bps ~ 4.5 Mbps

Table 3.2: Operational Details for the D’Manus

D’Manus a number of control modes as outlined in Table 3.2, allowing operational flexibility for end user applications¹.

3.3.4 Software

The *D’Manus*’s software package includes a python driver that exposes all the operational modalities outlined in Table 3.2, a detailed simulation model of the *D’Manus* based on MuJoCo (Figure 3.3), and a placeholder model of ReSkin sensors intended for prototyping. The software has been structured for ease of simulation testing and transfer to real hardware.

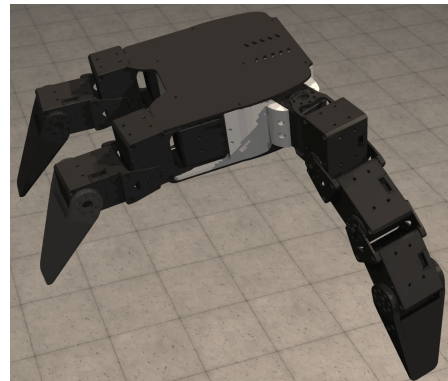


Figure 3.3: Simulated *D’Manus*

3.4 Experiments

The *D’Manus* is designed to sustain and support long hours of contact-rich interactions and data collection with minimal, easily fixable breakages. Such robustness allows the *D’Manus* to be used for long durations in a real-world robot learning setup similar to the systems demonstrated in *ROBEL* [2]. We evaluate the effectiveness of *D’Manus* as a testbed for real world robot learning along various axes –

1. *Dexterity*: In section 3.6.1, we evaluate *D’Manus*’s prehensile ability by subjecting it to a variety of objects and grasping scenarios.
2. *Tactile Perception*: In sections 3.6.2, 3.6.3, we validate the discriminative ability of extensive tactile sensing by using the *D’Manus* to perform material, softness, and texture identification purely based on surface properties.

¹PWM and current mode allow for force and hybrid position-force control

3. *Perceptive Generalization*: We demonstrate generalization of learned tactile models for softness and texture identification to unseen objects in section 3.6.3, and unseen tasks in section 3.6.4, to substantiate the stationarity and richness of ReSkin data.
4. *Integrated system*: We also corroborate the capabilities of the integrated *D’Manus* in section 3.6.4 by exposing it to unseen, real-time interactions in a bin picking setup and demonstrating automated bin sorting purely from tactile information (no visual inputs).
5. *Robustness*: Finally, in section 3.6.5, we outline *D’Manus*’s endurance and resilience towards extended periods of interaction rich operation.

While the *D’Manus* can be mounted on any robot arm, we used Franka Emika Panda robot for all our experiments. Neural network models presented in the following sections are trained on a single GPU (NVIDIA GeForce GTX 1080 Ti), and only take about 15 minutes of training time.

In the following section, we elaborate on the data collection and modeling choices for our learned tactile perception models before presenting experimental results in more detail.

3.5 Tactile Perception: Data and Modeling

We learn two *generalizable* tactile perception models training on ReSkin interaction from a variety of objects. The closest works in this space [45, 46] are restricted to classification problems that only demonstrate effectiveness on the **same** set of objects in the training set. We validate our models through testing on new, **unseen** objects. In this section, we detail the data collection setup and the modeling frameworks used to build these perception models.

3.5.1 Data Collection

To collect tactile interaction data, we fix the *D’Manus* such that the palm is facing upwards as shown in Fig. 3.4. For every object, we collect several trajectories of interaction data by placing it on the palm and executing a noisy, scripted motor babbling policy (at 30 Hz control frequency) for 10 seconds. While it is possible to train a policy optimized for recognition, we selected motor babbling due to its simplicity and ability to collect data with minimal user intervention, and to study the effectiveness of the tactile sensors in isolation from policy learning. Our data collection trajectory was generated by smooth interpolation between randomly sampled waypoints in the joint space. The waypoints were sampled in permissible range respecting the joint limits and self-collision. We found that this naive policy provided sufficient data diversity for training our models, and demonstrating generalization to unseen objects and tasks.

As the interaction policy is executed, ReSkin data from the fingertips and the palm is streamed to the control computer at every time step. Each frame of data consists of 3-axis magnetic flux measurements for each of the 56 magnetometers enumerated in Fig. 3.2. Sample data from an interaction trajectory can be seen in Fig. 3.5. A more dynamic visualization of the raw data can be found in the accompanying video.



Figure 3.4: **Data collection setup:** Tactile data is collected by placing the object on the palm and executing a human-scripted interaction policy for motor babble.

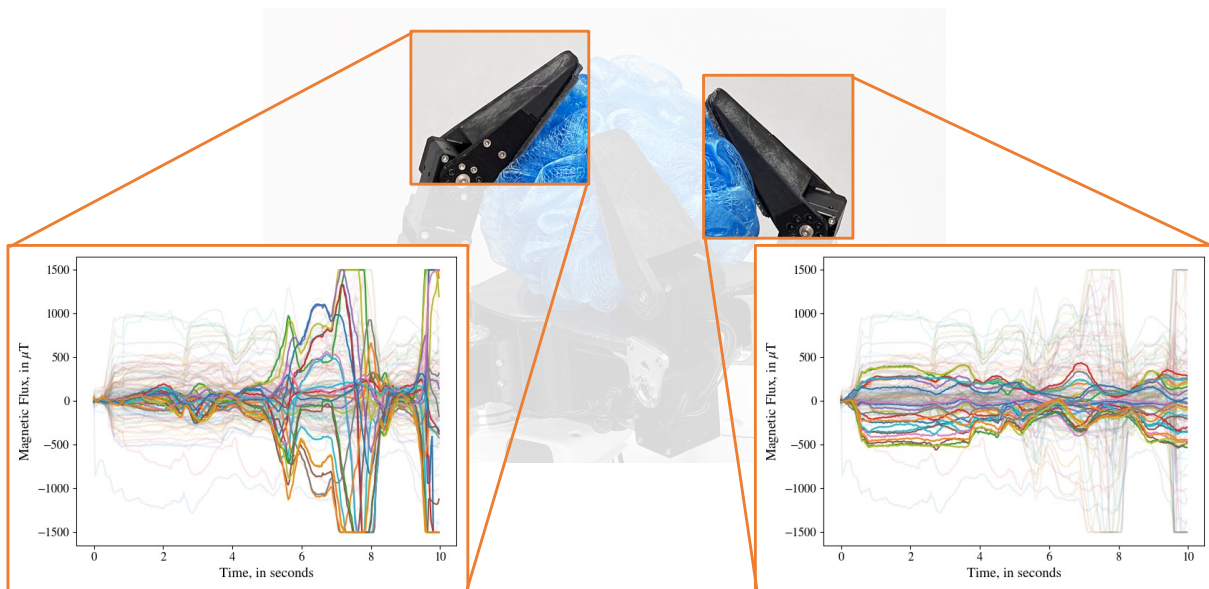


Figure 3.5: **Sample ReSkin data:** Visualization of tactile data from two of the fingers while interacting with the loofah in Fig. 3.4.

3.5.2 Model Learning

Vision-based tactile sensors [90, 160] have naturally leveraged convolutional neural networks (CNNs) as a backbone for processing tactile information, as the signal is fundamentally images. In contrast, the electromagnetic signals of ReSkin have much less redundancy and have relatively lower dimensionality. In order to allow the learning algorithm to pick from a larger class of functions, we choose to use fully connected multilayer perceptrons (MLPs) as building blocks for the neural architecture used to process ReSkin signal. Further, contact information from interaction is naturally sequential, and our model architecture must be capable of leveraging temporal correlations. To ensure this, we use a recurrent neural architecture, an LSTM, at the base of our model. The neural architecture used in this work, as shown in Fig. 3.6, consists of an LSTM with 2 hidden layers followed by 3 fully connected layers. All the models in the experiments take a sequence of magnetic flux vectors $\mathbf{B}_{t \times 168}$ as input and are classification models trained to minimize cross-entropy loss $-\sum_{x \in X} \log P(f_{\theta}(x) = \hat{y})$.

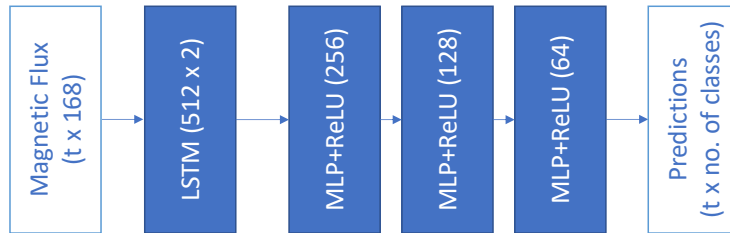


Figure 3.6: Model architecture

3.6 Results

3.6.1 Dexterity of the *D'Manus*

We qualitatively demonstrate the dexterous capabilities of the *D'Manus* (Fig. 3.8) using interactions with everyday objects. We observe that the *D'Manus* is effective at grasping and (in-hand as well as hand-arm) manipulation of day-to-day objects. Its abilities, however, are somewhat restricted for in-hand manipulation of small objects (e.g. counting coins on palm). This is in accordance with the dexterity and robustness trade-off we made and detailed in section 3.3.1.

3.6.2 Tactile Perception: Material Identification

To establish the effectiveness of tactile perceptual capabilities, we task the *D'Manus* to leverage only its tactile signals to classify objects. The idea is to demonstrate that we can build tactile models capable of capturing the differences between the tactile signature of different materials as obtained by the *D'Manus*. We pick a set of six balls of identical shape and size, but with a different outer covering – small bubble wrap, large bubble wrap, corrugated cardboard, silicone sponge, a combination, and no covering material – as shown in Fig. 3.7, and collect interaction data as described in Sec. 3.5.1. We use a 30-5 train-validation split for each ball.

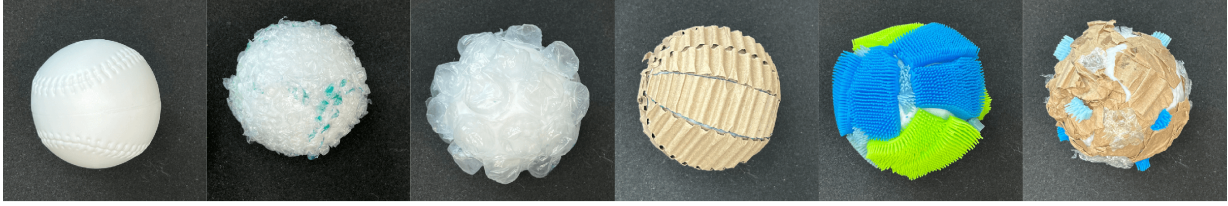


Figure 3.7: **Material coverings for Material Identification Task:** Uncovered, small bubble wrap, large bubble wrap, corrugated cardboard, silicone sponge and combination of materials



Figure 3.8: Illustration of the *D'Manus* grasping different objects with a variety of grasps [43, 156]

We train classification models to learn to predict a probability distribution over the six materials from tactile data, as detailed in section 3.5.2. Our models show a 71.24% accuracy on the 30 held-out trajectories (5 per material) described above, as shown in Table 3.3. This result confirms the discriminability of tactile interaction data obtained by the *D'Manus*.

Task	Validation accuracy
Material Identification	71.24%
Softness Classification	76.17%
Texture Classification	59.03%

Table 3.3: The *D'Manus* can distinguish between different materials purely using tactile feedback (Sec. 3.6.2). Further, models trained for softness and texture classification generalize to interactions with unseen objects (Sec. 3.6.3).

3.6.3 Perceptive Generalization: Softness and Texture

Having verified the distinguishability of sensory signals of *D'Manus*, we shift our focus to the consistency of the sensory signal across different objects and scenarios. We develop tactile perception models for softness and texture identification and demonstrate its generalization to unseen objects. For generalization, it is imperative that the (a) *D'Manus*'s sensors capture overall surface characteristics from interaction, (b) tactile perception models are robust to sensory drift over time, and (c) models are effective outside the training environment. To make our models robust to drift, training data is collected over an extended period of time (a few days). Test data is

collected in the subsequent days to validate robustness to drift. Further, the bin sorting experiment in Sec. 3.6.4 is performed about two weeks after training data is collected.

To learn tactile identification models, we would need quantifiable descriptions of surface characteristics. We create a three-point scale to quantify softness – *Hard, Medium, Soft* – as well as texture – *Smooth, Medium, Rough*. We manually assign softness and texture labels to over 50 objects by consensus among the authors **before** starting the study. We use a set of 20 training objects and 9 validation objects for these tasks. The full set of objects and their split can be found in Fig. 3.9. The corresponding datasets are created by collecting 15 trajectories of tactile interaction data for each of the training objects and 5 trajectories for each of the validation objects. We use the training data to train softness and texture identification models and examine their generalizability to unseen objects.



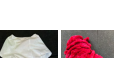










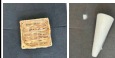









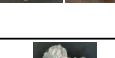






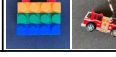


	Training			Validation			Test		
	Hard	Medium	Soft	Hard	Medium	Soft	Hard	Medium	Soft
Smooth									
									
Medium									
									
Rough									

Figure 3.9: Datasets used for Softness and Texture Identification Models

Softness Classification

We train a classification model as described in section 3.5.2 to predict softness categories, and present the results in Table 3.3. Our models successfully learn to classify objects on the softness scale defined above.

Texture Classification

Training a model for texture classification analogous to softness classification has low generalization on validation set. This can be attributed to the difficulty of characterizing texture independently of its softness properties. A hard, highly textured ball (Fig. 3.9, bottom left) has better defined irregularities on the surface when compared to the ball of yarn (Fig. 3.9, bottom right). Without softness labels, neural networks struggle to find correlations between textured objects across softness categories.

To get around this problem, we train separate *Softness-Conditioned* texture identification models for each softness category trained on the corresponding subset of training data. To make a prediction, a sequence of tactile measurements is first input to the softness model which outputs a softness label. The texture model corresponding to this softness label is then used to predict

a texture label from the same input sequence. The accuracy presented in Table 3.3 is the mean accuracy over the three softness categories.

A highlight of *D’Manus* design, as outlined in section 3.3.2, is the large sensorized area of the palm. To corroborate this design choice, we train standalone softness and texture identification models corresponding to each individual fingertip and the palm. We evaluate the performance of these standalone models and present a comparison in Table 3.4. The performance on the individual finger models is significantly lower than the palm as well as the full prediction model. While some of this discrepancy can be attributed to the fingers losing contact with the object during parts of the interaction trajectory, strong performance of other models emphasizes the benefits of large area sensing available on *D’Manus*.

Component	Softness Accuracy	Texture Accuracy
Finger 1	56.27%	49.77%
Finger 2	48.48%	46.99%
Finger 3	59.59%	50.50%
Palm	74.31%	54.20%
All	76.17%	59.03%

Table 3.4: Comparison of models trained using data from different components of the hand.

To further substantiate the strength of tactile perceptual capabilities of the *D’Manus*, and demonstrate the integrated system in an unseen environment, we deploy the learned softness and texture identification models in a tactile-aware bin sorting task in the following section.

3.6.4 Tactile Bin Sorting

As our final experiment, we assess the ability of our trained models to generalize to realistic environments and tasks. For this evaluation, we pick a cluttered bin sorting experiment. We attempt to pick objects from a cluttered bin and sort them according to softness and texture from tactile signals. We start with a cluttered bin as shown in Fig. 3.1. The robot samples a random (x,y) location and reaches down into the bin until the ReSkin signal exceeds a specified threshold, indicating the presence of an object. Then, a predefined grasp is executed and the hand is raised. If the grasp is unsuccessful, the robot returns to random location selection. If the grasp is successful, we predict softness and texture labels and sort the object into corresponding bins. We then replace it by adding a new object to the bin and the process is continued. Over 20 successful grasps of different objects, our models achieve a prediction accuracy of 65% on both softness and texture prediction, confirming the ability of our models to extend to unseen tasks and environments in the real world. Further, it is worth noting that the tactile models trained with the hand upright are able to generalize to this setting where the hand is primarily operated in a downward-facing configuration. Through this experiment, we validate the ability of the integrated *D’Manus* system as well the generalizability of our models in performing tactile-rich tasks in unseen, real-world environments.

3.6.5 Robustness and Reliability

Amongst various versions of the platform, we have logged over 10,000 hours of operational time over the course of 12 months in 3 different locations with a total of 5 breakages. These breakages consisted of three motor failures, one 3D printed part failure, and operational deterioration of wires – all of which were repaired in-house within 30 minutes by computer scientists unfamiliar with the working details of the *D’Manus*. We attribute the robustness largely to the motor selection - the high-quality Dynamixel XM430-210 uses metallic gears and a high safety factor for our required force range. The version of the platform being released has significantly benefited from aggressive real world testing of prior versions. The specific copy of the *D’Manus* used for experimental results has been running for over 400 hours over the last 8 months with no breakages, corroborating our claims about the robustness and reliability of this system for real-world learning in contact-rich robotic tasks.

3.7 Conclusions and Limitations

We present the *D’Manus* – a low-cost, 3D printable, prehensile robotic hand geared towards robot learning. The hand comes with multiple actuation modes, proprioceptive sensing abilities as well as ReSkin-based large-area tactile sensing. We demonstrate the dexterity of this platform in grasping a variety of objects. To exemplify the utility of the large-area sensing, we validate the discriminability of the tactile signal by learning models for material identification as well as category-level softness and texture identification. Further, we illustrate the transferability of learned tactile models to unstructured, real-world environments through a touch-based bin picking and sorting task. The design, assembly and setup instructions have all been open-sourced to facilitate adoption by the community.

Limitations: While we validate the tactile capabilities of the hand, it falls short of validating the tactile sensing in conjunction with dexterity. For future work, we would like to build a dexterous policy with tactile sensing to explore this further. The global semiconductor shortage limited the number of magnetometer chips we were able to integrate into this system. As a result, the system lacks sensing on the phalanges, ie. the surface of the motors, as well as the sides and backs of the fingertips. The design principles outlined in section 3.3.2 make this a simple extension of the present version and will be addressed in subsequent work. We also believe that unlocking the full potential of all-over tactile sensing requires integration of other sensory modalities like vision and audio, allowing the system richer sensory inputs to solve complex dexterous tasks. Finally, this work lacks quantitative comparisons to other existing platforms due to the high cost involved in such a pursuit. We hope that open-sourcing *D’Manus* and its low cost will help with comparative evaluations with our system.

Chapter 4

AnySkin: Tailoring tactile skins for robot learning

Bhirangi, R., Pattabiraman, V., Cao, Y., Haldar, S., Majidi, C., Gupta, A., Hellebrekers, T., & Pinto, L. *In Preparation*.

VP and YC were responsible for data collection. SH and RB were responsible for policy learning architecture search. VP, SH and RB were responsible for experiment design and analysis. RB was responsible for designing the fabrication procedure for AnySkin.

Abstract

While tactile sensing is widely accepted as an important and useful sensing modality, its use pales in comparison to other sensory modalities like vision and proprioception. AnySkin addresses the critical challenges of versatility, replaceability, and data reusability, which have so far impeded the development of an effective solution. By decoupling the sensing electronics from the sensing interface, AnySkin simplifies integration as well as replacement, making it as straightforward as putting on a phone case and connecting a charger. This work makes three key contributions: first, we introduce a streamlined fabrication process and a design tool for creating an adhesive-free, durable and easily replaceable magnetic tactile sensor; second, we characterize and policy learning with a AnySkin sensor; and finally, we demonstrate the generalizability of models trained on one instance of AnySkin to new instances.

4.1 Introduction

Touch sensing is widely recognized as a crucial modality for biological movement and control [77, 78, 79]. Unlike vision, sound, or proprioception, touch provides sensing at the point of contact, allowing agents to perceive and reason about forces and pressure. However, a closer examination of robotics literature reveals a different narrative. Prominent works and current state-of-the-art in robot learning primarily utilize vision sensing in conjunction with proprioception to train manipulation skills [16, 161], often ignoring touch. If touch is indeed vital from a biological perspective, why does it remain a second-class citizen in sensorimotor control?

To address this question, let’s examine what made cameras ubiquitous in robotics. Three key factors are at play: cost, convenience, and consistency. Cameras are relatively inexpensive (under \$20), easy to integrate on a wide variety of robot platforms (e.g. multi-view, depth, ego-centric), and allow for models trained on them (e.g. object detection, segmentation) to easily transfer to images captured with new cameras. In contrast, touch sensors are often costly due to expensive fabrication processes [132] or the need for high-end components (e.g., Gelsight). They are inconvenient to use on different robot platforms, being custom-built for specific robot end-effectors and constrained form factors requiring extensive adaptation for different shapes [40, 146]. Finally, touch sensors are inconsistent. Due to boutique fabrication, sensor profiles can vary significantly even when produced through the same process. This inconsistency poses a challenge when transferring tactile-based models across different instances of the same sensor. This transfer is particularly critical for touch sensors, which must conform to their environment to ensure stable grasps when sensing contact information. The requirement for a soft sensing interface to achieve conformal contact accelerates wear, leading to more frequent replacements.

In this work we present AnySkin, a new touch sensor that is cheap, convenient to use and consistent across different sensor instances. AnySkin builds on ReSkin [13], a magnetic-field based touch sensor, by improving its fabrication, separating the sensing mechanism from the interaction surface, and developing a new attachment mechanism. This allows AnySkin to (a) have stronger magnetic fields, which significantly improves its sensor response, (b) be easy to fabricate for arbitrary surface shapes, which allows easy use on different end-effectors, (c) be easy to replace the sensor without adverse affecting the data collection process or the efficacy of tactile-model trained on previous sensors.

We run a suite of experiments to understand the efficacy of AnySkin as a touch sensor for policy learning. Our main findings can be summarized below:

1. AnySkin can readily be used on a variety of robots including xArm, Franka, and the four-fingered LEAP hand [124] as shown in Figure 4.1 (See fabrication details in Section 4.4).
2. AnySkin is compatible with ML techniques for visuo-tactile policy learning for precise tasks such as inserting plugs (See learning details in Section 4.5).
3. Models trained on one AnySkin directly transfers to a different AnySkin with only a 6% reduction in performance on the plug insertion task compared to the 66% drop in performance with ReSkin sensors (See results in Section 4.5).

AnySkin is fully open-sourced and videos of fabrication, attachment, and robot policies are best viewed on our project website.

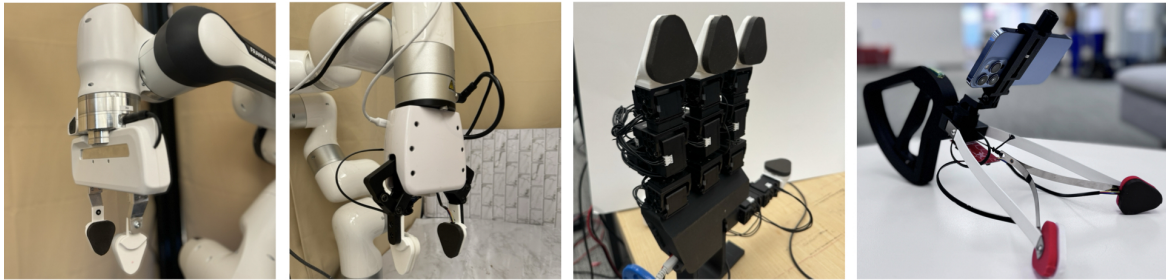


Figure 4.1: AnySkin is easy to integrate with a range of end effectors

4.2 Related Work

4.2.1 Tactile sensing

Existing literature on tactile sensing explores a wide range of modalities, each with their own set of advantages and limitations. Capacitive sensors [128] sense contact through changes in capacitance, offering high sensitivity, but struggling to model shear force and being prone to breakage due to direct electrical connections between the circuitry and elastomer. Resistive sensors [132] are simple and durable, but tend to provide more spatially discrete sensing with lower spatial resolution. Optical sensors [18, 19] capture contact information using cameras to track the deformation of an elastomer and provide high spatial resolution, but often pose hard, stringent limits on the sensor form factor, due to physical constraints on the camera field of view. This complicates integration for a wide range of applications and significantly increases the effort required to sensorize surfaces of different shapes and sizes.

Magnetic tactile sensors [69, 70] largely overcome these limitations due to three salient advantages: (a) separating the sensing electronics from the sensing interface to improve robustness (b) compatibility with different form factors, and (c) an ability to capture shear forces as demonstrated in [13]. Two prominent classes of magnetic sensors in robotics right now – ReSkin [13] and uSkin (by Xela Robotics) use elastomeric sensing interfaces with magnetic microparticles and macro-sized magnets respectively. In this work, we build on ReSkin sensors due to their lower cost and ease of fabrication.

4.2.2 Replaceability for Tactile Sensors

Recent developments in rapid prototyping and elastomer technology have spurred a substantial rise in the number of robotic tactile sensors. Discussions on replaceability of these sensors continue to be few and far between. There are two main factors to consider when evaluating replaceability: (a) the physical ease of replacing the sensor, and (b) signal consistency when replacing the sensor with a new instance of the same sensor. Relatively speaking, the former is

more frequently discussed [13, 90, 150] and resolved by simply separating the sensing interface – generally the elastomer that is more susceptible to wear – from the sensing electronics which tend to last much longer. The latter, however, tends to be much less discussed, and anecdotal evidence suggests most researchers circumvent the problem by sticking to a single instance of the sensor throughout their experiments. Signal consistency is imperative to making tactile sensing a ubiquitous presence in robot learning, since it facilitates better generalizability of trained models when sensing interfaces are inevitably replaced. In this paper, we evaluate the consistency of AnySkin signal through a policy learning experiment in Section 4.5.

4.2.3 Visuotactile Policy Learning

As tactile sensors have increased in number and popularity, so have learning frameworks attempting to use tactile data in conjunction with other modalities to learn policies for robot manipulation. A number of these works however, are restricted to simulation [65] with limited transfer to the real world [92]. Works that demonstrate impressive sim2real transfer are often restricted to imprecise continuous motion tasks, and the corresponding algorithms do not admit themselves to precise manipulation in the real world [118, 157]. Often, learned multimodal policies are evaluated with very limited variability prompting questions of their improvement over open-loop rollouts of recorded demonstrations [92, 94], or do not sufficiently disentangle the effect of different modalities, thereby bringing into question the multimodal nature of the policy [28]. In this work, we learn visuotactile policies capable of solving a highly precise manipulation task: plug insertion, while varying the location of the socket each time, and achieving successful insertion for unseen socket locations in the real world, while providing a comprehensive analysis of the disentangled effect of different modalities.

4.3 Background

4.3.1 ReSkin: replaceable magnetic tactile skins

ReSkin [13] is a set of magnetic tactile skins that uses a magnetic elastomer in conjunction with a magnetometer circuit to measure deformation. In this work, we build on the ReSkin sensor to create the AnySkin sensor that enables us to use the same circuitry as [13], but an upgraded, self-adhering skin. The fabrication of these skins is described in Section 4.4, and the advantages in terms of replaceability are elucidated in Section 4.5.

4.3.2 BAKU: transformer architecture for multimodal learning

In order to learn visuotactile policies in this work, we build on the BAKU architecture [62]. A schematic representation of the architecture used in our Behavior Cloning experiments is presented in Figure 4.2. Each modality – four cameras and a tactile sensor – is first tokenized using a corresponding encoder. An action token is appended to the tokenized encoders and the resulting token sequence is passed through a transformer encoder. The output of this encoder corresponding to the action token is then passed through an action head (in our case, an MLP) and

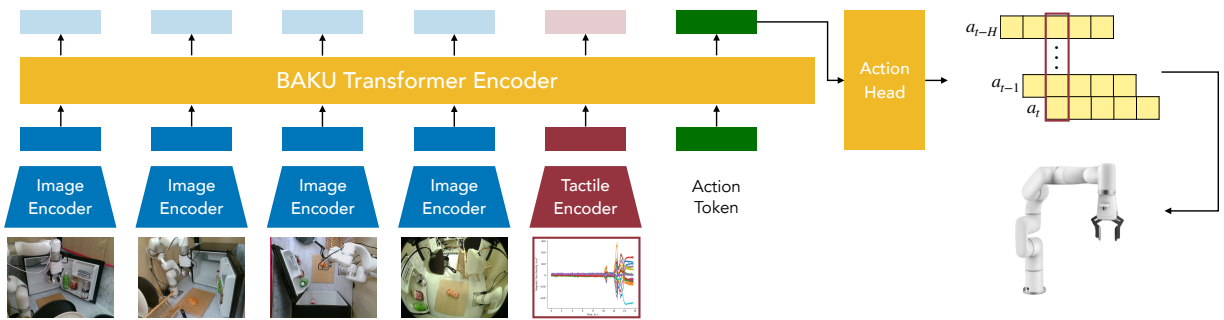


Figure 4.2: BAKU architecture used in our experiments

is used to predict the next H actions [162]. At evaluation time, drawing from action chunking with transformers [162], we use a smoothed average of the current action predicted over the last H timesteps.

4.4 Fabrication

4.4.1 Mold design

While ReSkin significantly reduces the difficulty of replacing the sensing skin by separating the sensing interface from the electronics, robust integration is still challenging due to the difficulty of securing a soft skin to a hard surface. Using screws [13] results in a loose fit between the electronics and skin and tends to tear the skin at the fastening point. While using adhesives can be an attractive option on face value, adhesion between silicone and non-silicone materials is an active topic of research and the process of properly securing the skin can be cumbersome and time-consuming. Furthermore, adhesives tend to wear out over time and replacing the adhesive can result in significant variability in sensor response as noted in Chapter 3. To circumvent these problems, we present a self-adhering mold design that enables us to design skins that naturally cling to the surface of the end effector similar to a phone case.

To create self-adhering skins, we create two part mold as shown in Fig. 4.3. We use a triangular shape to allow the fingertip to reach into cramped spaces and choose a thickness of 2 mm following [13]. While the results in this paper are presented for the triangular shape shown in Fig. 4.3, we also release a CAD tool that can convert a 2D drawing of a fingertip surface and design a corresponding 2-part mold for creating custom-shaped skins. Unlike a number of tactile sensors proposed in recent years that require significant effort and systems research for the smallest change in form factor [90], AnySkin seeks to simplify the process of diversifying your tactile sensor.

4.4.2 Magnetic elastomer fabrication

For the fabrication of the magnetic elastomer, we build upon the procedure detailed in Chapter 3, incorporating a pulse magnetizer to magnetize the skins. To ensure a more uniform distribution of magnetic particles throughout the skin, we utilize finer NdFeB MQFP particles. Additionally,

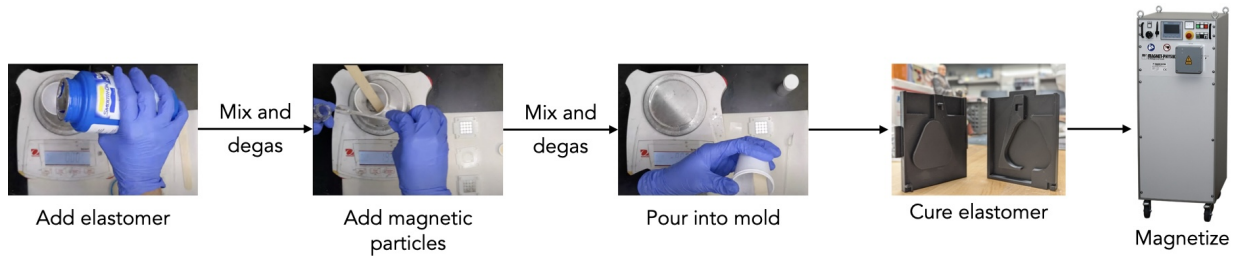


Figure 4.3: Fabrication procedure for AnySkin

we employ custom-designed two-part molds to create self-adhering, adhesive-free skins. These molds are 3D-printed using PLA plastic and are precisely aligned using integrated alignment guides. While the results presented in this paper use a triangular tip, as depicted in Figure 4.3, we also provide an open-source CAD tool that enables users to generate new mold designs by simply sketching the desired 2D surface of the tip. The two-part mold is secured with simple plastic clamps, and the elastomer compound is mixed and degassed before the magnetic microparticles are manually stirred into the mixture, followed by another round of degassing. After degassing, the magnetic elastomer mixture is poured through the inlet point shown in Figure 4.3, with the process paused as needed to allow the mixture to flow through the mold and emerge at the outlet point. The filled mold is then placed in a vacuum chamber, where a pressure of 29 mm Hg is applied. The process is paused intermittently to prevent overflow as the bubbles dissipate. This pressure is maintained until the mixture stops bubbling before the vacuum is released. The molds are left to rest for 16 hours before being carefully opened to reveal the fully cured AnySkin. After removing any excess material, the cured skin is placed in the pulse magnetizer, where a 4 Tesla (40 kOe) impulse is applied perpendicular to the largest surface of the skin, completing the magnetization process.

4.5 Experiments

4.5.1 Experimental Setup

Our experimental setup consists of a 7-DOF X-Arm robot as shown in Figure 4.4. There are three fixed cameras attached to the frame of the robot cage, in addition to one wrist camera attached to the robot. The AnySkin sensor is integrated with one of the tips attached to the X-Arm gripper, while the other tip is identically shaped and covered with non-sensorized elastomer. Manipulation policies are trained using behavior cloning; the Meta Quest 3 is used with the OpenTeach [74] library to collect 100 teleoperated demonstrations for the task at hand. For the analysis presented here, we evaluate performance on the plug insertion task shown in Figure 4.5. The location of the socket strip is changed for each demonstration in training data. For evaluation, we set aside socket strip locations never seen in the training data and use the same set of locations for all policy evaluations presented here.

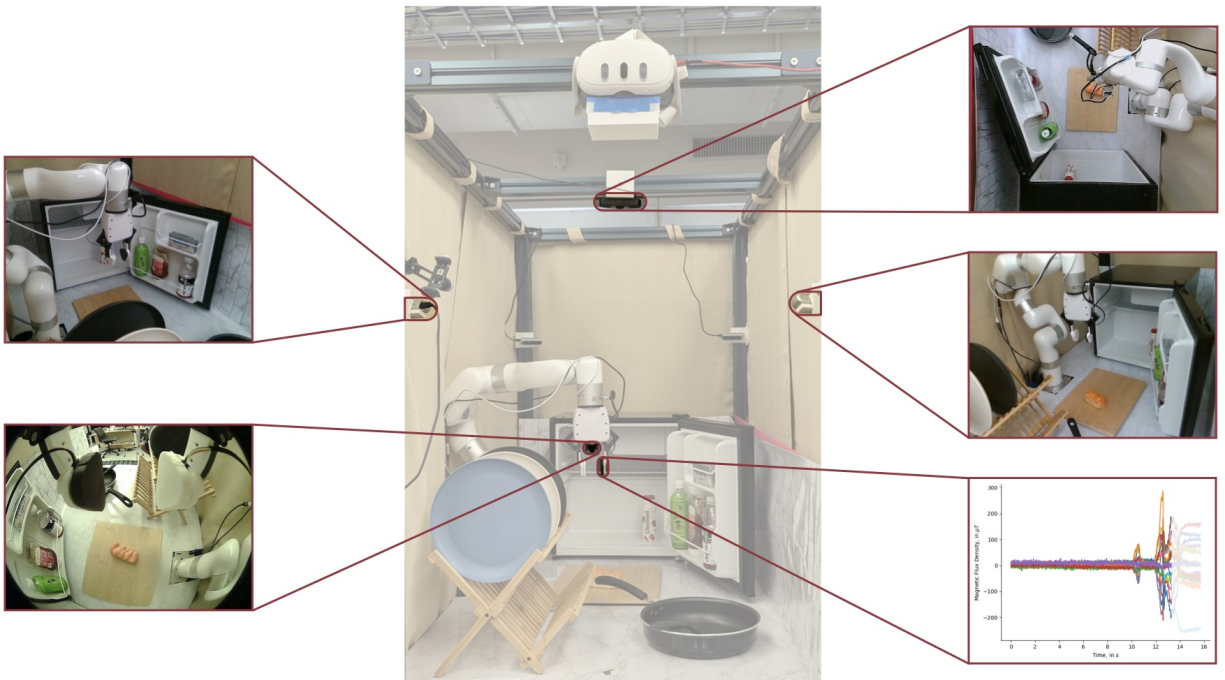


Figure 4.4: Experimental setup for visuotactile policy learning consisting of three fixed cameras, one wrist camera, and an AnySkin sensor on one of the gripper tips

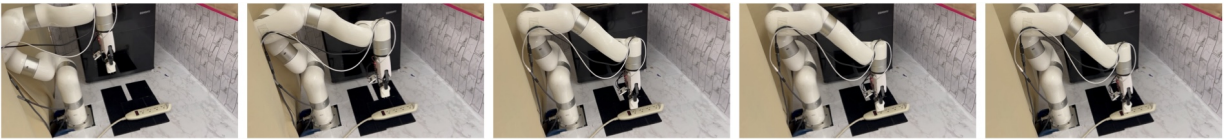


Figure 4.5: Progression of the plug insertion task used in policy learning experiments

Modalities used	Success rate
Fixed cameras	0/15
Fixed cameras, wrist camera	3/15
Fixed cameras, AnySkin	5/15
Fixed cameras, wrist camera, AnySkin	10/15
Fixed cameras, wrist camera, swapped AnySkin	9/15

Table 4.1: Policy performance on the plug insertion task with different input sets.

4.5.2 Results

In our policy learning experiments for the plug insertion task, we aim to address two primary questions:

- What impact do the addition of a wrist camera and the AnySkin sensor have on policy performance?
- How does substituting AnySkin influence the performance of the learned policy?

To explore the first question and isolate the effects of various inputs on policy performance, we trained four distinct policies, each utilizing a different combination of inputs, as detailed in Table 4.1. We began by establishing a baseline policy using only fixed cameras, which failed entirely to perform the task. This outcome was expected, given that the insertion task demands highly precise control of the robot, which is challenging to achieve using the distant perspective provided by the fixed cameras alone. Next, we assessed policies that incorporated the fixed cameras along with either the wrist camera or the AnySkin sensor, but not both. While these policies were able to complete the task, their performance remained suboptimal. Finally, we evaluated a policy that integrated both the wrist camera and the AnySkin sensor in addition to the fixed cameras. This combination led to a twofold improvement in performance compared to the next best policy. This result underscores the value of using wrist cameras in tandem with tactile sensing, such as the AnySkin sensor, for executing precise manipulation tasks effectively.

To address the second question, we applied the same policies using fixed cameras, the wrist camera, and the AnySkin sensor, but with a new tactile skin. This approach allows us to measure the effect of the skin replacement on performance, providing insight into how changing tactile skins influences the learned policies. Our results show a minor performance drop of about 6% with the new skin as shown in Table 4.1. In contrast, using ReSkin sensors for an analogous experiment had similar success rates with the training data skin, but swapping skins led to a significant performance drop of approximately 60%, rendering the policy unable to complete the task. This demonstrates that the AnySkin sensor offers better signal consistency across different instances, making it more replaceable than the ReSkin sensor.

4.6 Conclusion

To conclude, we have developed an enhanced version of the ReSkin sensor, which eliminates the need for adhesive and offers improved signal consistency. This advancement is achieved through a refined fabrication process that incorporates a two-part mold and a pulse magnetizer. We have also integrated the AnySkin sensor into policy learning for a precision manipulation task, where it significantly outperforms the combination of fixed cameras and a wrist camera. Furthermore, the sensor exhibits superior replaceability compared to ReSkin.

Given these results, the AnySkin sensor stands out as a promising alternative for robot learning, thanks to its seamless integration with various robotic systems. Its ease of replacement is particularly advantageous for gathering extensive tactile data across different platforms, potentially facilitating large-scale representation learning with touch information. Even though our experiments are focused on a single task, further preliminary results suggest that our findings on replaceability and policy learning gains are applicable to a range of precision manipulation tasks, including card swiping and USB insertion. While some limitations of ReSkin, such as interference from magnetic objects, persist in AnySkin, we believe that AnySkin holds significant promise as an effective and scalable tactile solution for robotic manipulation, paving the way for future advancements in this field.

Chapter 5

Hierarchical state space models for continuous sequence-to-sequence modeling

Bhirangi, R., Wang, C., Pattabiraman, V., Majidi, C., Gupta, A., Hellebrekers, T., & Pinto, L. (2024). Hierarchical State Space Models for Continuous Sequence-to-Sequence Modeling. In Forty-first International Conference on Machine Learning.

CW and VP were responsible for collecting data and setting up experiments for the "collected datasets" presented as part of CSP-Bench. RB was responsible for designing the model architecture, training and evaluating models, as well as the analysis presented in the paper.

Abstract

In delving into representation learning for ReSkin, we found that reasoning from sequences of raw sensory data is a ubiquitous problem across fields ranging from medical devices to robotics. These problems often involve using long sequences of raw sensor data (e.g. magnetometers, piezoresistors) to predict sequences of desirable physical quantities (e.g. force, inertial measurements). While classical approaches are powerful for locally-linear prediction problems, they often fall short when using real-world sensors. Sensors are typically non-linear, are affected by extraneous variables (e.g. vibration), and exhibit data-dependent drift. For many problems, the prediction task is exacerbated by small labeled datasets since obtaining ground-truth labels requires expensive equipment. In this work, we present Hierarchical State-Space Models (HiSS), a conceptually simple, new technique for continuous sequential prediction. HiSS stacks structured state-space models on top of each other to create a temporal hierarchy. Across six real-world sensor datasets, from tactile-based state prediction to accelerometer-based inertial measurement, HiSS outperforms state-of-the-art sequence models such as causal Transformers, LSTMs, S4, and Mamba by at least 23% on MSE. Our experiments further indicate that HiSS demonstrates efficient scaling to smaller datasets and is compatible with existing data-filtering techniques. Code, datasets and videos can be found on <https://hiss-csp.github.io>.

5.1 Introduction

Sensors are ubiquitous. From air conditioners to smartphones, automated systems analyze sensory data sequences to control various parameters. This class of problems - continuous sequence-to-sequence prediction from streaming sensory data - is central to real-time decision-making and control [122, 130]. Yet, it has received limited attention compared to discrete sequence problems in domains like language [39] and computer vision [38].

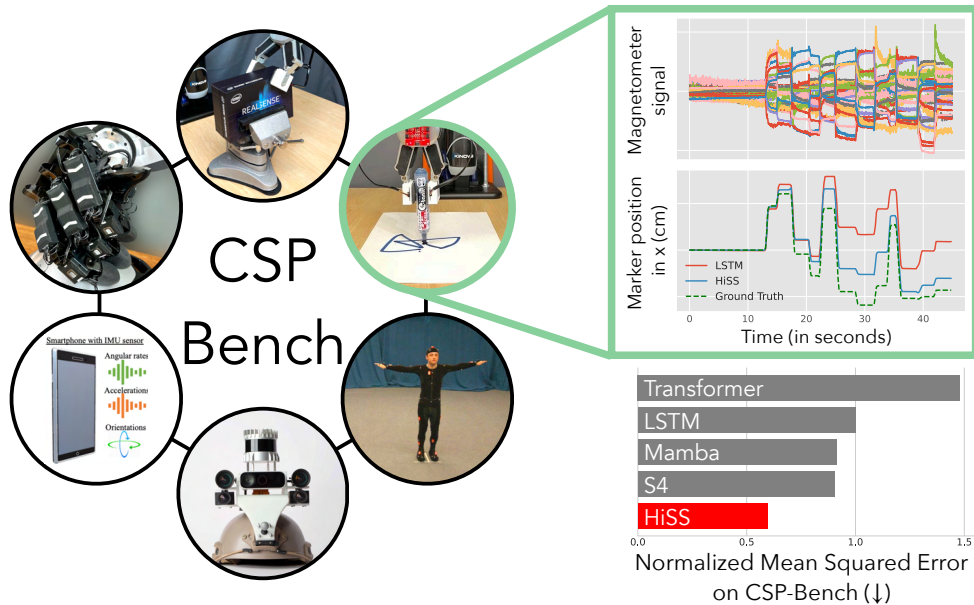


Figure 5.1: CSP-Bench is a publicly accessible benchmark for continuous sequence prediction on real-world sensory data. We show that Hierarchical State Space Models (HiSS) improve over conventional sequence models on sequential sensory prediction tasks.

Existing approaches for prediction from sensory data have largely relied on model-based solutions [37, 149]. However, these approaches require domain expertise and accurate modeling of complex system dynamics, which is often intractable in real-world applications. Moreover, sensory data contains noise and sensor-specific drift that must be accounted for to achieve high predictive performance [99]. In this work, we investigate deep sequence-to-sequence models that can address these challenges by learning directly from raw sensory streams.

However, to make progress on continuous sequence prediction (CSP), we first need a representative benchmark to measure performance. Most prior works in CSP focus on a single class of sensors [71, 99], making it difficult to develop general-purpose algorithms. To address this, we created CSP-Bench, a benchmark consisting of six real-world labeled datasets. This collection consists of three datasets created in-house and three curated from prior work – a cumulative 40 hours of real-world data.

Given data from CSP-Bench, an obvious modeling choice is to use state-of-the-art sequence models like LSTMs or Transformers. However, sensory data is high-frequency, leading to long sequences of highly correlated data. For such data, Transformers quickly run out of memory, as they scale quadratically in complexity with sequence length [142], while LSTMs require

significantly larger hidden states [87]. Deep State Space Models (SSMs) [56, 57] are a promising new class of sequence models. These models have been shown to effectively handle long context lengths while scaling linearly with sequence length in time and memory complexity, with strong results on audio [54] and language modeling. On CSP-Bench, we find that SSMs consistently outperform LSTMs and Transformers with an average of 10% improvement on MSE metrics (see Section 5.6). But can we do better?

A key insight into continuous sensor data is that it has a significant amount of temporal structure and redundancies. While SSMs are powerful for modeling this type of data, they are still temporally flat in nature, i.e. every sample in the sequence is reasoned with every other sample. Therefore, inspired by work in hierarchical modeling [135, 158], we propose Hierarchical State-Space Models (HiSS). HiSS stacks two SSMs with different temporal resolutions on top of each other. The lower-level SSM temporally chunks the larger full-sequence data into smaller sequences and outputs local features, while the higher-level SSM operates on the smaller sequence of local features to output global sequence prediction. This leads to further improved performance on CSP-Bench, outperforming the best flat SSMs by 23% median MSE performance across tasks. We summarize the contributions of this paper as follows:

1. We release CSP-Bench, the largest publicly accessible benchmark for continuous sequence-to-sequence prediction for multiple sensor datasets. (Section 5.4)
2. We show that SSMs outperform prior SOTA models like LSTMs and Transformers on CSP-Bench. (Section 5.6.1)
3. We propose HiSS, a hierarchical sequence modeling architecture that *further* improves upon SSMs across tasks in CSP-Bench. (Section 5.5)
4. We show that HiSS increases sample efficiency with smaller datasets, and is compatible with standard sensor pre-processing techniques such as low-pass filtering. (Sections 5.6.5, 5.6.6)

5.2 Related Work

5.2.1 Sequence-to-sequence prediction for sensory data

Most real world control systems, such as wind turbine condition monitoring [130], MRI recognition [84] and inertial odometry [4, 98], often process noisy sensory data to deduce environmental states. Traditionally, these problems were solved as estimation and control problems using filtering techniques, like the Kalman Filter [104, 126], that still require complex sensor models. Deep learning has shown promise in domains without analytical models, yet many solutions continue to be sensor-specific [71, 153].

Deep State Space Models (SSMs) [56, 57, 116, 127] are an emerging class of models that improve over conventional sequence models in modeling long-range dependencies – an important consideration for high-frequency sensory data. However, to the best of the authors’ knowledge, none of these models have been evaluated on continuous sensing data beyond audio [54]. In this work, we benchmark deep SSMs on six sensory sequence-to-sequence prediction tasks on sensors such as ReSkin, XELA, accelerometers, and gyroscopes.

5.2.2 Hierarchical Modeling

Incorporating temporal hierarchies into sequence modeling architectures has been shown to improve performance across a number of tasks like recommender systems [158], human activity recognition [135] and reinforcement learning [49, 88, 134]. HiSS is inspired by this line of work and extends it to SSMs for continuous seq-to-seq tasks.

5.2.3 Data for Continuous Sequence Prediction

A primary challenge with developing general models for continuous sequence prediction is the lack of a concrete evaluation benchmark. Odometry/SLAM datasets [50, 101] are viable candidates [21, 131] for CSP datasets. But most data across sensory modalities like audio [51, 148], ECG [109, 144], IMU [22, 24, 106] and tactile sensing [14, 45, 115] is labeled sparsely only at the sequence level.

The recent proliferation of sensors in smartphones and other smart devices has resulted in renewed interest in creating labeled datasets for CSP [23, 71]. A common setting is to use a motion capture system to obtain dense, sequential labels for sensory data from inexpensive IMU sensors [48, 138]. In this work, we curate three such datasets as part of CSP-Bench: a continuous sequence prediction benchmark.

Another category of sensors of significant interest for CSP are touch sensors. Touch sensors capture the dynamics of contact between the robot and its surroundings. Deep learning and rapid prototyping have driven a rapid surge across a range of tactile modalities from optical [90, 160] to capacitive [128] and magnetic sensing [13, 137]. Most work on continuously reasoning over tactile data is directed towards policy learning [19, 59, 60], where small datasets and confounding factors make it difficult to evaluate the efficacy of architectures for CSP. In this work, we set up supervised learning problems to investigate sequence-to-sequence models for two magnetic tactile sensors: ReSkin [13] and XELA [137].

5.3 Background

5.3.1 Sequence-to-sequence Prediction

Consider a data-generating process described by the Hidden Markov Model in Figure 5.2. The observable processes – sensor, S , and output, Y , represent two measurement devices that capture the evolution of the unobserved latent process, X . Generally, S is a noisy, low-cost device like an accelerometer, and Y is a precise, expensive labeling system like Motion Capture. The goal is to learn a model that allows us to estimate Y using data sequences from S .

The CSP problem involves estimating the probability of the t -th output observation, y_t , given the history of input observations, $s_{1:t}$. For the experiments listed in this paper, we approximate this probability by a Gaussian with constant standard deviation, ie. $p(y_t | s_1, \dots, s_t) = \mathcal{N}(\mu_\theta(s_{1:t}), \sigma^2 I)$, where σ is a constant, and parameterize μ_θ by a deep sequence model. Our goal is to find the maximum likelihood estimator for this distribution – $\arg \min_\theta \sum_t \|y_t - \mu_\theta(s_{1:t})\|^2$. Therefore, our models are trained to minimize MSE loss over the length of the output sequence.

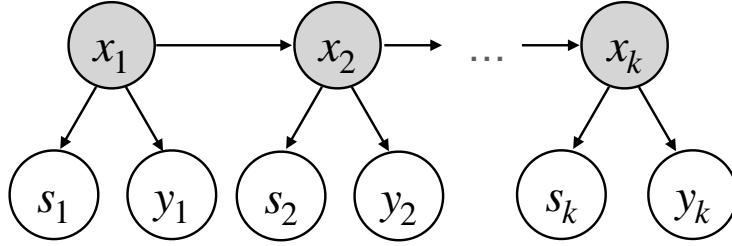


Figure 5.2: Hidden Markov Model for a two-sensor system. X is a data-generating process. Sensor, S , and output, Y , are two observable processes.

5.3.2 Deep State Space Models

Deep State Space Models (SSMs) build on simple state space models for sequence-to-sequence modeling. In its general form, a linear state space model may be written as,

$$\begin{aligned} x'(t) &= \mathbf{A}(t)x(t) + \mathbf{B}(t)u(t) \\ y(t) &= \mathbf{C}(t)x(t) + \mathbf{D}(t)u(t), \end{aligned}$$

mapping a 1-D input sequence $u(t) \in \mathbb{R}$ to a 1-D output sequence $y(t) \in \mathbb{R}$ through an implicit N-D latent state sequence $x(t) \in \mathbb{R}^n$. Concretely, deep SSMs seek to use stacks of this simple model in a neural sequence modeling architecture, where the parameters, \mathbf{A} , \mathbf{B} , \mathbf{C} and \mathbf{D} for each layer can be learned via gradient descent.

SSMs have been proven to handle long-range dependencies theoretically and empirically [58] with linear scaling in sequence length, but were computationally prohibitive until Structured State Space Sequence Models (S4) [57]. S4 and related architectures by [44, 116, 127] are based on a new parameterization that relies on time-invariance of the SSM parameters to enable efficient computation. Recently, Mamba [56] improved on S4-based architectures by relaxing the time-invariance constraint on SSM parameters, while maintaining computational efficiency. This allows Mamba to achieve high performance on a range of benchmarks from audio and genomics to language modeling, while maintaining linear scaling in sequence length. In this paper, we benchmark the performance of SSMs like S4 and Mamba on sensory CSP tasks, and show that they consistently outperform LSTMs and Transformers.

5.4 CSP-Bench: A Continuous Sequence Prediction Benchmark

We address the scarcity of datasets with dense, continuous labels for sequence-to-sequence prediction by collecting three touch datasets with 1000 trajectories each and combining them with three IMU datasets from literature to create CSP-Bench. For each dataset, we design tasks to predict labeled sequences from *single* sensor data to avoid confounding factors. We also include data from varied sources like cameras and robot movements to facilitate future research in multi-sensor integration and multimodal learning. The detailed characteristics of these datasets are summarized in Table 5.1, aiming to support diverse sensory data analysis.

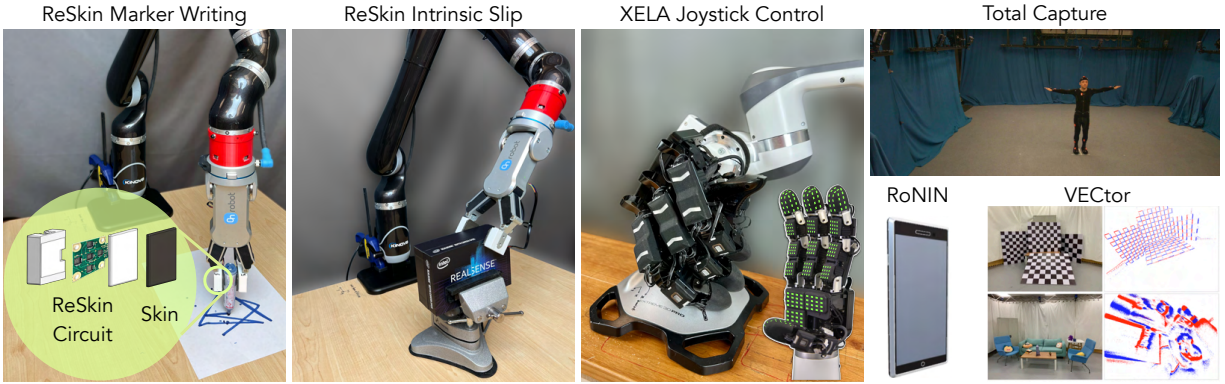


Figure 5.3: **CSP-Bench** is comprised of six datasets. Three datasets – ReSkin Marker Writing, ReSkin Intrinsic Slip and XELA Joystick Control are tactile datasets collected in-house on two different robot setups as demonstrated above. Three other datasets – RoNIN [71], VECTor [48] and TotalCapture [138] are curated open-source datasets.

5.4.1 Touch Datasets

Our touch datasets are collected on two magnetic tactile sensor designs: ReSkin [13] and Xela [137]. The ReSkin setup consists of a 6-DOF Kinova JACO Gen1 robot with a 1-DOF RG2 OnRobot gripper as shown in Figure 5.3. Both gripper surfaces are sensorized with a $32\text{ mm} \times 30\text{ mm} \times 2\text{ mm}$ ReSkin sensor. Each sensor has five 3-axis magnetometers which measure changes in magnetic flux resulting from the deformation of the skin on the gripper surface. Appendix A.1 contains more details on the fabrication and integration of ReSkin into the gripper.

The Xela setup consists of a 7-DOF Franka Emika robot fitted with a 16-DOF Allegro hand by Wonik Robotics. Each finger on the hand is sensorized with three 4×4 uSkin tactile sensors and one curved uSkin tactile sensor from XELA Robotics as shown in Figure 5.3. Sensor integration was provided by XELA robotics, which was designed specifically for the Allegro Hand. While the underlying sensory mode is the same for both ReSkin and Xela, they differ in spatial and temporal resolution, physical layout, and magnetic source.

ReSkin: Marker Writing Dataset

We collect 1000 Kinova robot trajectories of randomized linear strokes across a paper. Initially, the marker is arbitrarily placed between the gripper tips, and data collection begins when the marker touches the paper. The robot then moves linearly between 8-12 random points uniformly sampled within a $10\text{ cm} \times 10\text{ cm}$ workspace, pausing for a randomly sampled delay of 1-4 seconds after each motion. Images of sample trajectories can be found in Appendix A.3.

The goal of this sequential prediction problem is to use tactile signal from the gripper to predict the velocity of the end-effector in the plane of the table. Velocity labels are easily obtained from robot kinematics, and serve as a proxy for the velocity of the marker strokes against the paper. What makes this problem challenging is that the sensor picks up contact information from both, the relative motion between the marker and the gripper, and the motion of the marker against the paper. The model must learn to disentangle these two motions to make accurate predictions.

Table 5.1: Summary of all the modalities present in CSP-Bench. Modalities used for training are *italicized*. In addition to the data used for training models, we also release synchronized video and robot kinematics data to facilitate further research in CSP problems.

Dataset	Modalities	Model Inputs (dim)	Model Outputs (dim)	Size (min)
Marker Writing	<i>ReSkin</i> (100 Hz), 2 Cameras (30 Hz), <i>Robot</i> (45 Hz)	ReSkin (30)	End-effector velocity (2)	420
Intrinsic Slip	<i>ReSkin</i> (100 Hz), 3 Cameras (30 Hz), <i>Robot</i> (45 Hz)	ReSkin (30)	End-effector velocity (3)	640
Joystick Control	<i>Xela</i> (100 Hz), 2 Cameras (30 Hz), <i>Robot</i> (50 Hz), <i>Hand</i> (300 Hz), <i>Joystick</i> (20 Hz)	<i>Xela</i> (552)	Joystick State (3)	580
VECtor [48]	<i>IMU</i> (200 Hz), 2 Cameras (30 Hz), <i>RGBD</i> (30 Hz), <i>Lidar</i> (10 Hz), <i>MoCap</i> (120 Hz)	<i>IMU</i> (7)	User velocity (3)	22
TotalCapture [138]	<i>IMU</i> (60 Hz), 8 Cameras (60 Hz), <i>MoCap</i> (60 Hz)	<i>IMU</i> (39)	Joint velocities (60)	45
RoNIN [71]	<i>IMU</i> (200 Hz), <i>3D Tracking</i> <i>Phone</i> (200 Hz)	<i>IMU</i> (7)	User velocity (2)	600

ReSkin: Intrinsic Slip Dataset

We again use the Kinova setup to collect 1000 trajectories of intrinsic slip – the gripper grasping and slipping along different boxes clamped to a table. At the start of every episode, we close the gripper at a random location and orientation on the box and start recording data. We sample 8-12 random locations and orientations within the workspace of the robot along the length of the box, and then command the robot to move along the box while slipping against it. We use 10 boxes of different sizes to collect this dataset to improve data diversity in terms of contact dynamics. Example images and dimensions are available in Appendix A.3.1.

The goal of the sequential prediction problem is to use the sequence of tactile signals from the gripper tips to predict the translational and rotational velocity of the end-effector (again obtained from robot kinematics) in the plane of the robot’s motion. In addition, the abrasive nature of the task causes the skin to wear out over time. To account for this wear, we change the gripper tips and skins after 25 trajectories on every box, improving data diversity as a result.

XELA: Joystick Control Dataset

For our final dataset, we record 1000 trajectories of data from the Allegro hand interacting with the joystick as shown in Figure 5.3. The hand/robot setup is teleoperated using a VR-based system derived from HoloDex [6]. Joystick interactions are logged synchronously with robot data, tactile sensing data, and the camera feed. Specifically, this includes the full robot kinematics (7 DOF Arm at 50 Hz + 16 DOF Hand at 300 Hz), XELA tactile output (552 dim at 100 Hz), and 2 Realsense D435 cameras (1080p at 30 Hz). Each trajectory consists of 25-40 seconds of interaction with the joystick.

The goal of the sequential prediction problem is to use tactile signal from the Xela-sensorized robot hand to predict the state of the joystick, which is recorded synchronously with all the other modalities. The extra challenge with this dataset, in addition to the significantly higher dimensionality of the observation space, is the noisier trajectories resulting from human demos instead of a scripted policy.

5.4.2 Curated Public Datasets

In addition to the tactile datasets we release with this paper, we also test our findings on data from other datasets, particularly ones using IMU sensor data (illustrated in Figure 5.3) – the RoNIN dataset [71] which contains smartphone IMU data from 100 human subjects with ground-truth 3D trajectories under natural human motions, the VECtor dataset [48] – a SLAM dataset collected across three different platforms, and the TotalCapture dataset – a 3D human pose estimation dataset.

5.5 Hierarchical State-Space Models (HiSS)

In this work, we focus on continuous sequence-to-sequence prediction problems for sensors i.e. problems that involve mapping a *sequence* of sensory data to a *sequence* of outputs. In

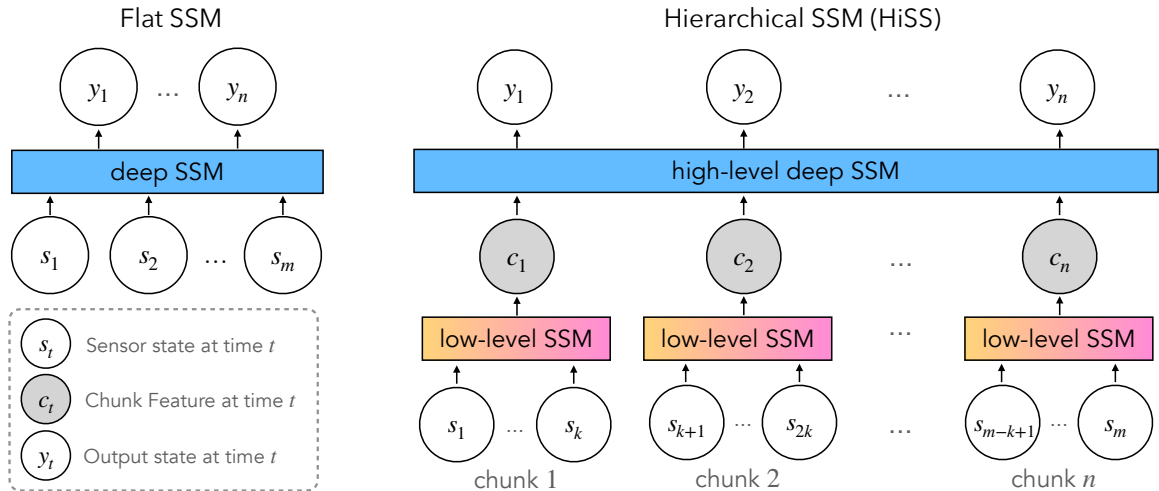


Figure 5.4: (Left) Flat SSM directly maps a sensor sequence to an output sequence. (Right) HiSS divides an input sequence into chunks which are processed into *chunk features* by a low-level SSM. A high-level SSM maps the resulting sequence to an output sequence.

the following sections, we describe our preprocessing pipeline and HiSS – our approach to sequence-to-sequence reasoning at different temporal scales.

5.5.1 Data Preparation and Sampling

Every sensor in the real world operates at a different frequency, and data from different sensors is therefore collected at different nominal frequencies. Generally, our sensor sequences come from an inexpensive, noisy sensor operating at a higher frequency than an expensive, high precision device which gives us output sequences. To emulate this scenario and standardize our experiments, all sensor sequences are resampled at a frequency of 50Hz, and output sequences are resampled at 5Hz for all the datasets under consideration, unless specified otherwise. The specific choice of these frequencies is dictated by the sampling frequencies of sensors in the available data.

All the sensors considered in CSP-Bench are prone to drift; therefore, in line with previous work [13, 60, 71], we estimate a resting signal at the start of every sensor trajectory and deviations from this resting signal are passed to the model. Since sensor drift can be causally data-dependent, the entire sensory trajectory is passed to the model as input. Sensor and output sequences are normalized based on data statistics for their corresponding datasets, and details are listed in Appendix A.2. Additionally, we find that appending one-step differences to every element in the sensor sequence helps improve performance, in line with numerous prior works [26, 72].

5.5.2 Model Architecture

Here we describe Hierarchical State Space Models (HiSS) – a simple hierarchical architecture that uses SSMs to explicitly reason over sequential data at two temporal resolutions, as shown in Figure 5.4. The sensor sequence is first divided into a set of equally-sized chunks of size k . Each chunk is passed through a shared SSM, say S4, which we refer to as the *low-level SSM*.

The outputs of the low-level SSM corresponding to the k -th element of each chunk are then concatenated to form a rarified *chunk feature* sequence. Finally, this sequence is passed through a *high-level* sequence model to generate the output sequence.

Why should HiSS work? Sequential sensory data is subject to phenomena that occur at different natural frequencies. For instance, an IMU device mounted on a quadrotor is subject to high-frequency vibration noise and low-frequency drift characteristic of MEMS devices [83]. With HiSS, our goal is to create a neural architecture with explicit structure to operate at different temporal scales. This will allow the low-level model to learn effective, temporally local representations, while enabling the high-level model to focus on global predictions over a shorter sequence.

5.5.3 Training details

We focus on sequence-to-sequence prediction tasks. All our models are trained end-to-end to minimize MSE loss as explained in Section 5.3.1. For all tactile datasets and VECtor, we use an 80-20 train-validation split. For the RoNIN dataset, we use the first four minutes of every trajectory for our analysis, and use a validation set consisting of trajectories from unseen subjects. For TotalCapture, we use the train-validation split proposed by Trumble et al. [138]. Hyperparameter sweep ranges for each of our models and baselines are listed in Appendix A.2. We maintain similar ranges of parameter counts across models for the same task.

5.6 Experiments and Results

In this section, we evaluate the performance of HiSS models on CSP tasks and understand their strengths and limitations. Unless otherwise specified, we use non-overlapping chunks of size 10, and aim to answer the following questions:

- How do SSMs compare with LSTMs and Transformers on CSP-Bench?
- Can HiSS provide benefits over temporally flat models?
- How does chunk size affect the performance of HiSS?
- Is HiSS compatible with existing preprocessing techniques like filtering?
- How does HiSS perform in low-data regimes?

Baselines: We use two categories of baselines: Flat and Hierarchical. Flat models consist of LSTMs, Causal Transformers, S4 and Mamba, in addition to MEGA [100]. Hierarchical baselines include variations of HiSS models where the high-level and/or low-level SSMs are replaced by causal transformers and LSTMs, and MEGA-chunk [100], which is loosely classified as a high-level transformer with a low-level MEGA model. Table 5.2 presents a performance comparison on CSP-Bench for each of these baselines and proposed HiSS models.

Table 5.2: Comparison of MSE prediction losses for flat and HiSS models on CSP-Bench. Reported numbers are averaged over 5 seeds for the best performing models. MW: Marker Writing, IS: Intrinsic Slip, R: RoNIN, V: VECtor, JC: Joystick Control, TC: TotalCapture

Model type	Model Architecture		MW	IS	JC	R	V	TC	
			(cm/s)			(m/s)	(m/s)	(m/s)	
Flat		Transformer	2.375	0.460	1.020	-	0.043	-	
		LSTM	1.168	0.310	1.074	0.044	0.035	0.177	
		S4	1.319	0.262	0.980	0.038	0.034	0.348	
		Mamba	0.883	0.176	1.064	0.040	0.032	0.364	
		MEGA	0.8960	0.2105	0.9806	0.0370	0.0330	0.1944	
Hierarchical		High-level							
			Low-level						
			Transformer	0.668	0.219	0.911	0.062	0.037	0.305
			LSTM	0.996	0.253	0.935	0.042	0.038	0.320
		Transformer	S4	<u>0.620</u>	0.157	0.898	0.036	0.037	0.358
			Mamba	1.027	0.202	0.906	0.047	0.037	0.456
		(MEGA-chunk)	MEGA	1.1270	0.2090	1.0450	0.0512	0.0403	<u>0.1940</u>
			Transformer	0.762	0.937	1.609	0.388	0.030	0.294
		LSTM	LSTM	0.866	0.284	1.076	0.044	0.029	0.252
			S4	0.637	<u>0.153</u>	0.908	0.048	0.032	0.350
			Mamba	0.792	0.192	1.061	0.044	0.029	0.364
			Transformer	0.757	0.290	0.925	0.044	0.030	0.245
		S4	LSTM	0.859	0.180	0.952	0.032	0.029	0.245
			S4	0.626	0.155	0.906	0.026	0.030	0.344
			Mamba	0.826	0.182	0.920	0.032	0.029	0.408
			Transformer	0.702	0.301	0.955	0.037	0.030	0.206
	Mamba	LSTM	0.759	0.175	0.964	0.035	<u>0.027</u>	<u>0.243</u>	
		S4	0.566	0.132	<u>0.901</u>	<u>0.030</u>	0.030	0.253	
		Mamba	0.725	0.168	0.905	0.032	0.025	0.376	
	HiSS	boost over best Flat	+36%	+25%	+8%	+31%	+21%	-37%	

5.6.1 Performance of Flat models on CSP-Bench

At the outset, we see that SSMs – Mamba and S4, consistently outperform the best-performing Transformer and LSTM models by 10% and 14% median MSE respectively across CSP-Bench tasks. The only anomaly is the TotalCapture dataset where the LSTM outperforms all other models. We analyze this later in Section 5.6.7.

5.6.2 Improving CSP Performance with HiSS

HiSS models perform better than the best-performing flat models, SSM or otherwise, with a *further* improvement of $\sim 23\%$ median MSE across tasks. Among hierarchical models, HiSS models continue to do as well as or better than the others with a relative improvement of $\sim 9.8\%$ median MSE. Further, we make two key observations within models that use a specific high-level architecture: (1) these models consistently outperform corresponding flat models, indicating that temporal hierarchies are effective at distilling information from continuous sensory data; (2) the best models use S4 as the low-level model, indicating that S4 is particularly adept at capturing low-level temporal structure in the data.

These observations raise a natural question: What is happening under the hood? In the next four sections, we attempt to better understand the working of HiSS.

5.6.3 Does HiSS Simply do Better Downsampling?

The first question we seek to answer is whether simply downsampling the sensor sequence to the same frequency as the output would do just as well as HiSS. As we see in Table 5.3, while some flat models with downsampled sensor sequences indeed improve performance over flat models in Table 5.2, they remain far behind HiSS models. This reinforces our hypothesis that HiSS models distill more information from the sensor sequence than naive downsampling.

One advantage of using hierarchical models is memory efficiency. They can significantly reduce computational load for models like transformers which scale quadratically in the length of the sequence. Using an SSM such as S4 or Mamba as the low-level model can significantly reduce the computational load ($O(N^2) \rightarrow O(N^2/k^2)$) for $k \ll N$, where k and n are chunk size and sequence length respectively. Table 5.2 shows that such a model consistently improves performance over a flat causal Transformer across tasks.

5.6.4 Effect of Chunk Size on Performance

Having established the effectiveness of HiSS relative to conventional sequence modeling architectures, we seek to investigate the effect of a key parameter – the chunk size – on the performance of HiSS models. Downsampling the sensor sequences at the output frequency, as presented in Section 5.6.3 essentially corresponds to using a chunk size of 1. The rest of the analysis presented so far uses a chunk size of 10, corresponding to the largest non-overlapping chunks that cover the entire sensory sequence given sensor and output sequence frequencies of 50 Hz and 5 Hz respectively. In this section, we conduct two additional experiments with chunk sizes of 5 and 15 and present the results in Table 5.4. We see that while performance improves drastically as the

Table 5.3: Performance comparison with (a) downsampled inputs, (b) low pass filter on input sequences, and (c) fewer training samples

	MW	IS	JC	R	V	TC
<i>Downsampled inputs</i>						
Trnsfrmr	2.41	0.33	.957	.116	.039	0.34
LSTM	1.92	0.27	.975	.094	.034	0.20
S4	2.22	0.29	.974	.081	.036	0.31
Mamba	1.96	0.26	.980	.077	.033	0.25
HiSS	0.57	0.13	.901	.027	.025	0.26
<i>Low Pass Filtering</i>						
Trnsfrmr	1.79	0.31	1.01	-	.034	0.38
LSTM	1.15	0.26	1.08	.038	.024	0.12
S4	1.19	0.22	0.94	.031	.022	0.25
Mamba	0.78	0.14	0.95	.030	.018	0.17
HiSS	0.55	0.11	0.87	.036	.020	0.13
<i>Smaller Training Dataset</i>						
Fraction	0.3	0.3	0.3	0.3	0.5	0.5
Trnsfrmr	4.30	0.85	1.237	-	.046	0.54
LSTM	1.83	0.54	1.313	.053	.039	0.39
S4	2.31	0.45	1.197	.043	.038	0.43
Mamba	1.74	0.37	1.195	.039	.036	0.48
HiSS	1.26	0.29	1.106	.034	.029	0.37

Table 5.4: Effect of chunk size on performance of HiSS models

Chunk size	MW	IS	JC	R	V	TC
5	1.18	0.20	.933	.046	.033	0.32
10	0.57	0.13	.901	.027	.025	0.25
15	0.54	0.12	.899	.035	.026	0.24

chunk size increases, it plateaus once the chunk size reaches the ratio of the sensor and output frequencies (10 in our case). This behavior can be explained by the fact that chunk sizes smaller than this ratio result in the model never seeing parts of the sensor sequence, while chunk sizes larger than this ratio result in an overlap between chunks.

5.6.5 Effect of Sensory Preprocessing on Performance

A common approach to preprocessing noisy sensor data is to design low-pass filters to process the signal before it’s passed through the model. To analyze the compatibility of HiSS models with such existing preprocessing techniques, we separately apply order 5 Butterworth filters with 3 different cut-off frequencies to the sensor sequence and report model corresponding to the best cut-off frequency in Table 5.3. We make two key observations: (1) with the exception of the HiSS model for RoNIN, low pass filtering improves performance across the board; (2) HiSS models still perform comparably with or better than flat models.

With respect to (1), we see that the best-performing HiSS model from Table 5.2 continues to outperform the best flat model using filtered data, implying that the low-pass filter may have filtered useful information could have been used to improve task performance. This points to an important pitfall of handcrafted preprocessing techniques – they can often filter out information that could have been exploited by a sufficiently potent model. Consequently, the ability of HiSS models to require little to no preprocessing of the input sequence bolsters their credentials to serve as general purpose models for CSP data.

5.6.6 How Does HiSS Perform on Smaller Datasets?

The lack of a comprehensive benchmark for continuous sequence prediction so far speaks to the difficulty of collecting large, labeled datasets of sensory data. Therefore, performance in low-data regimes could be critical to wider applicability of different sequence modeling architectures. To benchmark this performance, we compare the performance of flat as well as HiSS models on subsets of the training data. While TotalCapture and VECtor are substantially smaller than the other datasets (see Table 5.1), we include them in this analysis while using a larger fraction of training data than other datasets. Results are presented in Table 5.3. We only present the best performing HiSS model here for conciseness. The full table can be found in Appendix A.4.

We see that on smaller fractions of the training dataset, HiSS outperforms flat baselines on *every* task in CSP-Bench. This indicates an important property of HiSS models – data efficiency. Low-level models operate identically on all of the chunks in the data, allowing them to learn more effective representations from small datasets than flat models.

5.6.7 Failure on TotalCapture

The most visible failure case for the performance of both flat SSMs as well as HiSS models is on the TotalCapture dataset, where the flat LSTM significantly outperforms all other models. We hypothesize that the high dimensionality of the input and output spaces prevents SSMs from learning sufficiently expressive representations that can filter out high frequency data. This is also evidenced by the higher performance of LSTM low-level models across hierarchical architectures for this dataset, which correlates with the correspondingly higher effectiveness of the flat LSTM over flat SSMs. Further evidence of the inability of SSMs to filter out noise can be found in Section 5.6.5, where the performance of HiSS models nearly matches the LSTM when the input sequence is passed through a lowpass filter. This indicates that the HiSS model struggles to learn the filtering behavior from data here, unlike other datasets where performance remains fairly consistent with and without the lowpass filter.

5.7 Conclusion and Limitations

We present CSP-Bench, the first publicly available benchmark for Continuous Sequence Prediction, and show that SSMs do better than LSTMs and Transformers on CSP tasks. Then, we propose HiSS, a hierarchical sequence modeling architecture that is more performative, data efficient and minimizes preprocessing needs for CSP problems. However, sequence-to-sequence prediction from sensory data continues to be an open, relatively underexplored problem, and our work indicates significant room for improvement. Moreover, while SSMs show significant promise for CSP tasks, they are relatively new architectures whose strengths and weaknesses are far from being well-understood. Section 5.6.7 explains some of the challenges of SSMs, and as a result, HiSS, on high-dimensional prediction problems with small datasets of noisy sensor data. In terms of ease of training, current HiSS models introduce an additional hyperparameter of chunk size. While our models are robust to large chunk sizes, finding optimal ones is an exciting future direction. Finally, while CSP-Bench is large, the number of sensors that can benefit from learned models is larger. Hence, we are committed to supporting CSP-Bench and adding more, larger datasets in the future.

Chapter 6

Conclusion and Future Prospects

This thesis sought to explore the full trajectory of taking a tactile sensor from design and development all the way into the fold of data-driven learning for robotics. With ReSkin and AnySkin, and the associated focus on repeatability and replaceability, we provide sensing alternatives that are perfectly suited for robot learning. While some of the takeaways outlined through this document are specific to magnetic tactile sensing, we believe a number of the questions we address are fairly general to understanding and improving the interplay between tactile sensing and robot learning. In the rest of this chapter, we discuss a few open questions, connect the work in this thesis to those broader questions, and outline directions for future research that would take us closer towards finding answers to them.

How can we collect and effectively use large scale tactile datasets?

If we look at the biggest deep learning success stories in recent years – vision and language – a central component has been the swathes of Internet-scale data that has been used to train performative models with increasingly impressive capabilities. Thinking about large-scale data in the context of tactile sensing requires us to take a step back and acknowledge the number of ways in which tactile sensing is inherently different from these modalities, and the resulting challenges that come with it. A primary challenge is the lack of a standardized capture device as well as representations. Camera images and linguistic vocabularies serve the purpose of effectively capturing visual and linguistic data in a standardized, shareable format. The lack of a standardized tactile sensor and the absence of a framework to translate data between different tactile sensors precludes the creation of datasets at the same scale. Moreover, visual and linguistic information has been collected over the years for the purposes entirely divorced from training deep learning models such as communication, entertainment and recorded history. The lack of such auxiliary applications for tactile sensing also contributes to the limited availability of large collections of tactile data.

This prompts two distinct directions for future research: (i) designing interfaces for collecting tactile information, and (ii) developing algorithms that enable reasoning over tactile information in a shared representation space. There has been some work in the former, such as the MIT Glove [133], but most solutions share the problems inherent to their respective tactile sensors –

they are brittle and are extremely difficult to fabricate. An exciting direction for future work is the integration of robust, reliable sensors like ReSkin and AnySkin with easier to fabricate simpler data collection interfaces like the Stick [123] and UMI [31]. These interfaces can be replicated on the robot and offer the highest chances of success for using tactile data from the data collection interface in training policies for a robot with an identical end-effector.

The latter of the two directions is more ambitious and isn't obviously solvable but offers the highest potential for leveraging large amounts of tactile data. The associated problem is similar to the distribution shift problem when using human videos [8, 9, 11] where the algorithm must account for the difference in appearance between human hands in the datasets and the robot end effectors used during policy deployment. This problem is more pronounced with tactile data. Consider a dataset of a human performing tasks in the kitchen wearing a tactile glove. While the glove may faithfully capture all the contact interactions between the human hand and the environment, it is unclear how this information would transfer to a robot hand with a completely different morphology. Further, it is also difficult to ascertain whether the same strategy used by the human hand is compatible with a morphologically different robot hand, and therefore, if the contact interaction data from the human hand is even useful. The recent surge in humanoid research as well as more anthropomorphic robot hands may hold interesting answers to this question. Another interesting line of research is developing algorithms capable of learning from data collected with different tactile sensors like Gelsight [160] and ReSkin, as a step towards unifying the distributed, sensor-specific strands of contemporary research in tactile sensing.

What is an effective learning paradigm for visuotactile policies?

As of this writing, behavior cloning remains the predominant method for training robot manipulation policies in real-world applications. While it has demonstrated impressive capabilities in robot learning, behavior cloning is not ideal for developing visuotactile policies for robots due to its reliance on supervised learning from demonstration data. This approach can lead to compounding errors, where the robot encounters states not represented in the training data, resulting in poor generalization—an issue particularly acute with tactile data. For instance, in the plug insertion task discussed in Section 4.5, visual data from being close to the socket strip can be nearly indistinguishable from visual data during a collision. However, the tactile data in these scenarios is markedly different. A visual policy trained on expert demonstrations, where collisions are rare, may generalize reasonably well, but a visuotactile policy is much more vulnerable to out-of-distribution errors. Moreover, current teleoperation systems used for collecting demonstrations lack the ability to convey tactile feedback to the operator, further undermining the reliability of behavior cloning.

On the other hand, existing reinforcement learning algorithms, while offering an alternative, are often brittle, require large amounts of data, and depend on environmental resets that are difficult to implement in real-world settings. Therefore, a hybrid approach could offer significant advantages in addressing these challenges. Some recent work has attempted to use reinforcement learning in simulation in conjunction with domain randomization [118, 157] or real-world adaptation. While promising, precise manipulation tasks continue to remain outside the reach of this approach for now. A large part of this shortcoming can be attributed to the lack of reliable, accurate simulators

for tactile sensors. A promising direction for future work is building more performant physics engines capable of simulating the complex phenomena that underlie tactile sensors.

An advantage of ReSkin and AnySkin sensors in the context of policy learning is also the low dimensionality of sensory information while capturing essential characteristics of contact like contact location, and normal and shear force. While these sensors may not be able to reconstruct high resolution contact images akin to optical sensors like Gelsight, the inherent low dimensionality reduces the need for dimensionality reduction in preprocessing [90] and allows the signal to be directly leveraged in the policy pipeline.

In conclusion, this thesis has delved into the potential of tactile sensing to enhance robotic learning and interaction in complex, unstructured environments. Through the design of a sensor optimized for integration with machine learning, we have demonstrated a significant improvement in the ease with which tactile sensing can be employed to enhance robots' abilities to perceive and manipulate objects. These findings make a valuable contribution to the broader field of robotics by underscoring the critical role of tactile feedback and the essential considerations for sensor design to achieve human-like dexterity and adaptability. We hope that the results presented in this thesis will serve as a foundation for further research in robotic learning, bringing tactile sensing closer to becoming an indispensable component of advanced robotic systems.

Appendix A

Appendix for Chapter 5

A.1 ReSkin fabrication details

ReSkin measures the changes in magnetic flux in its X, Y and Z coordinate system, based on the change in relative distance between the embedded magnetic microparticles in an elastomer matrix and a nearby magnetometer. The use of magnetic microparticles enables freedom in regard to the shape and dimensions of the molded skin. In our use case here, we use a skin of thickness 2mm. This section further details the complete fabrication process involved in the sensorized gripper tips we use for our *ReSkin: OnRobot Gripper on a Kinova JACO Arm* setup (See Figure A.1).

A.1.1 Circuitry

Data from the ReSkin sensors is streamed to a computer via USB. The two sensors are connected to an I2C MUX which in turn is connected to an Adafruit QT Py microcontroller as described in Bhirangi et al. [13]. See Figure A.1.

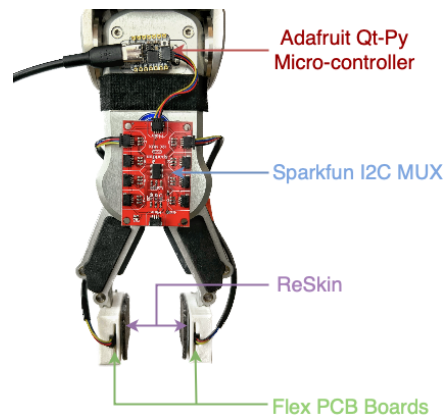


Figure A.1: Circuitry

A.1.2 OnRobot Gripper Tips

The skins are affixed to the 3D-printed gripper tips using silicone adhesive, as shown in Figure A.2. The dimensions of the tips are 32 mm \times 30 mm \times 2 mm. The same tips also house the flex-PCB boards, which measure the change in magnetic flux in all 3 axes.

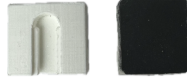


Figure A.2: Gripper Tips with ReSkin

A.2 Model architectures and Training

A.2.1 Flat Architectures

For each of the flat sequence models presented in this work, the input sequence is first embedded into a hidden state sequence by a linear layer. This hidden state is then passed to the respective sequence model. The outputs of the sequence model (the hidden states for LSTM, S4 and Mamba) are then mapped to the desired output space

A.2.2 Hierarchical architectures

The hierarchical models are obtained by simply stacking two flat models together. The input sequence is first divided into equal sized chunks as described in Section 5.5.2. Each chunk is passed through the low-level sequence model and the outputs corresponding to the last timestep of each chunk are concatenated to form the chunk feature sequence. This sequence is passed through a high-level sequence model to obtain the output sequence

A.2.3 Hyperparameters

All models are trained for 600 epochs at a constant learning rate of 1e-3. Learning rate schedulers were not found to improve performance by noticeable amounts. Table A.1 contains the ranges of hyperparameters used for training the flat models presented in the paper. We do not sweep over all of these hyperparameters for each task. A subset of these parameters is chosen for each task depending on the input and output dimensionality of the task and the best-performing models are reported. The exact hyperparameters for each experiment can be found on the Github repository. For any given task, we ensure that sweeps over all model classes consist of models that have the same order of magnitude of learnable parameters.

For the hierarchical models, we use a smaller subset of the parameters listed in Table A.1 to sweep over the high level models. Parameter ranges swept over for low-level models are listed in Table A.2. The exact hyperparameters for each experiment can be found on the Github repository.

LSTM	Transformer	S4	Mamba
Input size 16, 32, 64, 128, 256	Model dim 32, 64, 128, 256, 512	Model dim 32, 64, 128, 256, 512	Model dim 32, 64, 128, 256, 512
LSTM hidden size 256, 512, 1024	No. of heads 2,4		
No. of layers 2	No. of layers 4,6	No. of layers 4, 6	No. of layers 4, 6
Dropout 0.0, 0.1	Dropout 0.0, 0.1	Dropout 0.0, 0.1	

Table A.1: Hyperparameters for flat architectures

LSTM	S4	Mamba
Input size 16, 32, 64	Model dim 16,32,64,128, 256	Model dim 16, 32, 64, 128, 256
LSTM hidden size 16,32,64,128,256		
No. of layers 1	No. of layers 4, 6	No. of layers 3,4

Table A.2: Hyperparameters for low-level models used in hierarchical architectures

A.3 Experimental Setup and Data Collection details

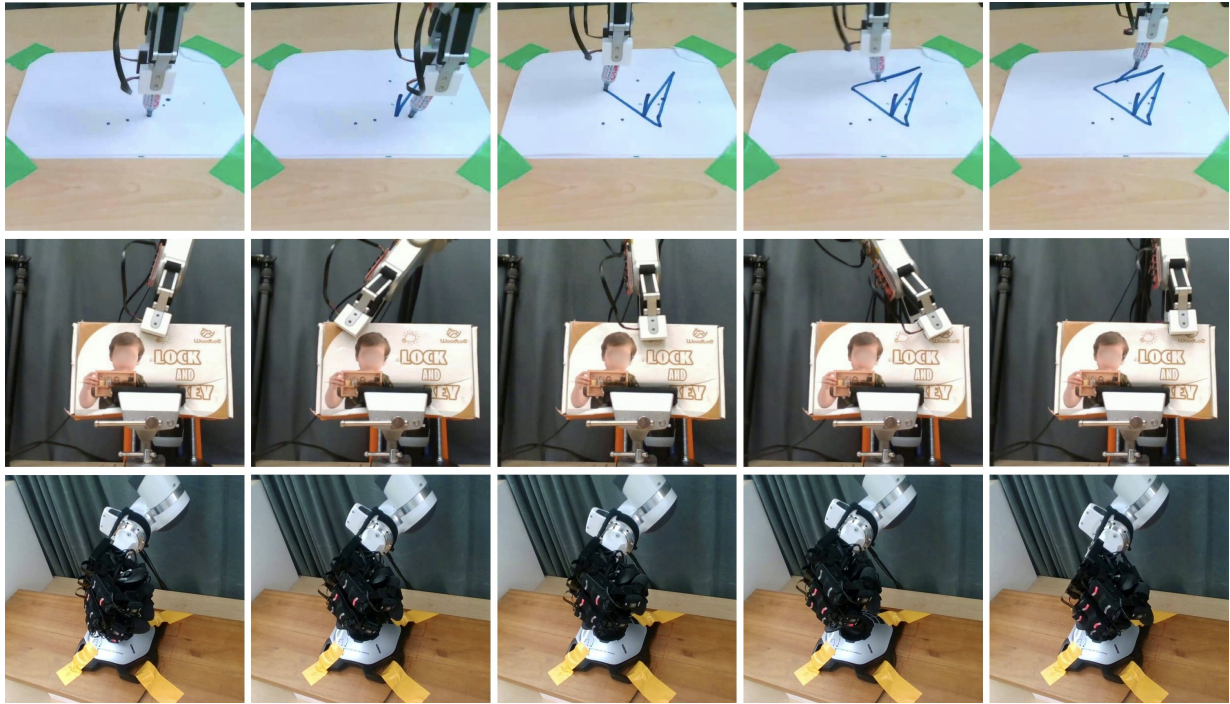


Figure A.3: Marker Writing Frames (Top): The gripper tips hold the marker and bring it in contact with the paper before the sequence starts. The arm maneuvers the marker to execute eight strokes on the paper. Intrinsic Slip Frames (Middle): The gripper tips hold the box to start the sequence, and slip through the robot workspace with different orientations. Joystick Control Frames (Bottom): After the sequence begins, the hand holds the joystick, controlling its movement through various positions.

A.3.1 ReSkin: Onrobot Gripper on a Kinova JACO Arm

Marker Writing

For this experiment, we first grasp the marker with 300 N force in an arbitrary position and bring it in contact with the paper. We then start recording data and command the robot to move sequentially to 8-12 randomly sampled locations within a $10 \times 10 \text{ cm}^2$ plane workspace, making linear strikes on the paper. Figure A.3 illustrates a sample sequence from this dataset. We note that during the strikes, the grasped marker undergoes orientation drifts at times, which adds to the complexity in signal. We record a total of 1000 trajectories of 15-30 seconds each, comprising of 2 different colored markers.

The prediction task here is to predict the strike velocity ($\delta x/\delta t$, $\delta y/\delta t$), given the tactile signals thus reconstructing the overall trajectory.

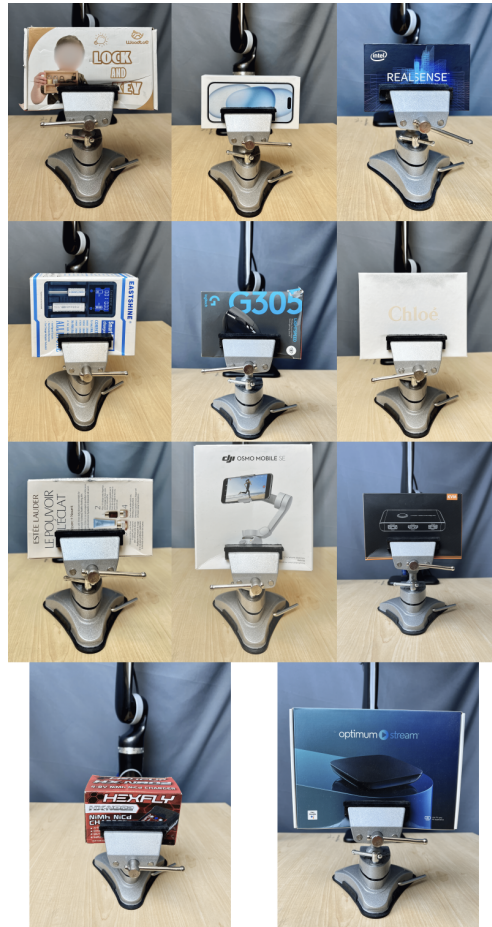


Figure A.4: Boxes in the Dataset

Intrinsic Slip

In Section 5.4.1, we outlined our methodology for collecting data through a total of 1000 trajectories. This involved using 10 distinct boxes and 4 sets of skins for 25 trajectories per combined pair. We first sample a random location and orientation within the task workspace. Next, we close the gripper with a random force sampled in the range of 50-75 N and then start recording data. With the gripper grasping the box, we uniformly sample 8-12 locations sequentially, thus slipping through the robot workspace. Figure A.3 illustrates a sample sequence from this dataset. The workspace is the upper region of the box, which is a space of dimensions $\text{Box Length} \times \text{Tip Size}$ (3cm), shown in Figure A.5. We clamp the wrist rotation limits at $[-\pi/4, \pi/4]$, making the overall local sampling bounds of the gripper tip position (center of tip), $Y:[0, \text{box length}]$, $Z:[0, \text{tip size}]$, $\theta:[-\pi/4, \pi/4]$.

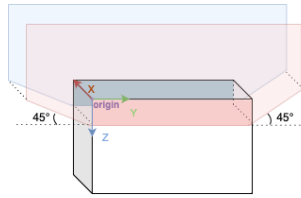


Figure A.5: End-effector Workspace on the Box, & Local Co-ordinate System

Bhirangi et al. [13] characterize the ability of ReSkin sensor models generalize to skins outside the training distribution, but these experiments are limited to single-frame, static data. Here, we collect an analogous dataset for the sequence-to-sequence prediction problem. To avoid confounding effects, the evaluations provided in this paper are based on a random partitioning of this dataset. However, we collect and publish an additional 100 trajectories on an unseen box and an unseen set of skins to test the generalizability of trained models.

The dimensions of all boxes used in this experiment are detailed below. See Table A.3 and Figure A.4.

In this experiment, in addition to predicting the linear velocities of the end-effector, we also predict the angular velocities at the wrist/the end-effector rotation ($\delta x/\delta t$, $\delta y/\delta t$, $\delta \theta/\delta t$).

A.3.2 Xela: Allegro Hand on a Franka Emika Panda Arm

Joystick Control

For the final tactile dataset, we teleoperate an Allegro Hand with Xela sensors mounted on a Franka arm to interact with an Extreme3D Pro Joystick shown in Figure A.6, which streams data comprising of 6 rotation axes (X, Y, Rz, Throttle, Hat0X, Hat0Y) and 12 buttons (Trigger, 2 Thumb Buttons, 2 Top Buttons, 1 Pinkie Button and 6 Base Buttons). Unlike the prior datasets, which originated out of random yet scripted policies, this dataset has an added complexity from the unstructured human interactive control. Figure A.3 illustrates a sample sequence from this dataset. Due to the arm workspace and the finger size constraints, we focus on 3 axes - X, Y and Z-twist for our prediction task. Given the readings from the Xela sensors, we predict the joystick's states of interest.

Box Number	Dimensions (L x H x W cm)
1	20 x 12 x 4
2	16.5 x 8.5 x 3
3	14 x 9 x 5
4	17 x 13 x 4.5
5	15 x 10 x 4.5
6	16.5 x 13 x 6
7	17 x 10 x 5.5
8	18 x 19.5 x 5.5
9	17 x 11 x 3.5
10	12 x 8 x 6.5
11 (unseen)	23 x 16 x 5

Table A.3: Dimensions of Boxes in the Dataset



Figure A.6: Extreme3D Pro Joystick & Co-ordinate System

A.4 Ablations

A.4.1 Data Preprocessing

In this section, we provide more detailed tables for the experiments in Sections 5.6.5. Table A.4 contains results from separately applying order 3 Butterworth filters to the input sequences with cutoff frequencies of 0.75Hz, 2.5Hz and 7.5Hz. For each setting, we pick the set of models corresponding to the cutoff frequency with the best performance, and report average performance over 3 seeds.

Table A.4: Comparison of MSE prediction losses for flat and HiSS models on CSP-Bench when passing the input sequences through a low-pass filter. Reported numbers are averaged over 5 seeds for the best performing models. MW: Marker Writing, IS: Intrinsic Slip, JC: Joystick Control, TC: TotalCapture

Model type	Model Architecture		MW (cm/s)	BS	JC	RoNIN (m/s)	VECtor (m/s)	TC (m/s)
Flat	Transformer		1.7940	0.3096	1.0080	-	0.0346	0.3845
	LSTM		1.1498	0.2596	1.0770	0.0382	0.0242	0.1234
	S4		1.1885	0.2209	0.9449	<u>0.0305</u>	0.0228	0.2467
	Mamba		0.7823	0.1367	0.9459	0.0297	0.0188	0.1661
Hierarchical	High-level	Low-level						
		LSTM	1.0052	0.1883	0.9074	0.0532	0.0284	0.2314
	Transformer	S4	0.6703	0.1249	0.8652	0.0434	0.0260	0.2908
		Mamba	0.8912	0.1251	0.8731	0.0435	0.0243	0.3118
		LSTM	0.8063	0.2434	1.0500	0.0430	0.0272	0.1754
	LSTM	S4	<u>0.6462</u>	0.1477	0.9885	0.0419	0.0288	0.1968
		Mamba	0.7515	0.1657	1.0080	0.0420	0.0234	0.1755
		LSTM	0.8525	0.1390	0.9269	0.0306	0.0272	0.1905
	S4	S4	0.6667	0.1221	0.9296	0.0377	0.0222	0.2284
		Mamba	0.7825	0.1180	0.8898	0.0396	<u>0.0207</u>	0.2527
		LSTM	0.8143	0.1308	0.9660	0.0369	0.0255	0.1594
	Mamba	S4	0.5535	<u>0.1074</u>	<u>0.8665</u>	0.0362	0.0272	<u>0.1301</u>
	Mamba	1.5657	0.1057	0.8765	0.0367	0.0212	0.1466	

Table A.5: Comparison of MSE prediction losses for flat and HiSS models on CSP-Bench when using a fraction of the training dataset. Reported numbers are averaged over 5 seeds for the best performing models. MW: Marker Writing, IS: Intrinsic Slip, JC: Joystick Control, TC: TotalCapture

Model type	Model Architecture		MW	IS	JC	RoNIN	VECTor	TC
			(cm/s)			(m/s)	(m/s)	(m/s)
	(Fraction)		0.3	0.3	0.3	0.3	0.5	0.5
Flat	Transformer		4.2975	0.8509	1.2370	-	0.0460	0.5430
	LSTM		1.8322	0.5376	1.3130	0.0533	0.0390	0.3855
	S4		2.3070	0.4450	1.1970	0.0431	0.0379	0.4338
	Mamba		1.7443	0.3677	1.1950	0.0394	0.0358	0.4838
	High-level	Low-level						
		LSTM	<u>1.5417</u>	0.3428	1.2350	0.0387	0.0331	0.3982
	S4	S4	1.5460	<u>0.2931</u>	1.1260	0.0346	0.0337	0.3992
		Mamba	2.3302	0.3760	1.1060	0.0412	0.0326	0.4913
		LSTM	1.5810	0.3478	1.2410	<u>0.0362</u>	<u>0.0309</u>	0.3530
	Mamba	S4	1.2600	0.2883	1.1370	0.0378	0.0333	<u>0.3675</u>
Hierarchical		Mamba	1.7508	0.3688	<u>1.1140</u>	0.0383	0.0286	0.4320

A.4.2 Smaller Datasets

In this section, we provide more detailed tables for the experiments in Sections 5.6.6. Table A.5 contains results from subsampling the training datasets – 30% of the dataset for MW, IS, JC and RoNIN, and 50% of the dataset for VECTor and TotalCapture. We see that HiSS consistently outperforms flat models across tasks in CSP-Bench when training on fractions of the training dataset, indicating the sample efficiency of HiSS models.

A.5 TotalCapture Preprocessing

This dataset provides readings from 12 IMU sensors and the ground truth poses of 21 joints obtained from the Vicon motion capture system. To standardize the data within a consistent coordinate system, we transformed all IMU sensor readings from their native IMU frames to the Vicon frame. Our task is to predict the velocities of the 21 joints given the IMU acceleration data in the Vicon reference frame.

To convert IMU acceleration data into the Vicon frame, we utilize the calibration results provided in the files named `<subject_id>_<sequence_name>_calib_imu_ref.txt` and `<sequence_name>_Xsens_AuxFields.sensors`. The acceleration of each IMU sensor in the Vicon frame is calculated as follows:

$$a_{\text{vicon}} = R_{\text{inertial}}^{\text{vicon}} R_{\text{imu}}^{\text{inertial}} a_{\text{imu}}, \quad (\text{A.1})$$

where $R_{\text{imu}}^{\text{inertial}}$ is the rotation matrix converted from the IMU local orientation quaternion (w, x, y, z) provided in the `<sequence_name>_Xsens_AuxFields.sensors` files. This quaternion represents the IMU's orientation in the inertial reference frame.

Furthermore, $R_{\text{inertial}}^{\text{vicon}}$ is obtained by converting the quaternion information (`<imu_name> x y z w`) available in the `<subject_id>_<sequence_name>_calib_imu_ref.txt` files, which encapsulates the transformation from the inertial frame to the Vicon global frame.

Bibliography

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1.2
- [2] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pages 1300–1313. PMLR, 2020. 1.1, 3.1, 3.2.1, 3.4
- [3] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik’s cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019. 1.3
- [4] Navid Amini, Majid Sarrafzadeh, Alireza Vahdatpour, and Wenyao Xu. Accelerometer-based on-body sensor localization for health and medical monitoring applications. *Pervasive and mobile computing*, 7(6):746–760, 2011. 1.2, 5.2.1
- [5] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 1.3, 3.1, 3.2.1
- [6] Sridhar Pandian Arunachalam, Irmak Güzey, Soumith Chintala, and Lerrel Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023. 5.4.1
- [7] Ozgur Atalay, Asli Atalay, Joshua Gafford, and Conor Walsh. A highly sensitive capacitive-based soft pressure sensor based on a conductive fabric and a microporous dielectric layer. *Advanced materials technologies*, 3(1):1700237, 2018. 2.1
- [8] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. *arXiv preprint arXiv:2207.09450*, 2022. 6
- [9] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 6
- [10] George A Bekey, Rajko Tomovic, and Ilija Zeljkovic. Control architecture for the belgrade/usc hand. In *Dextrous robot hands*, pages 136–149. Springer, 1990. 1.1, 3.2.1
- [11] Homanga Bharadhwaj, Abhinav Gupta, Vikash Kumar, and Shubham Tulsiani. Towards

- generalizable zero-shot manipulation via translating human interaction plans. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6904–6911. IEEE, 2024. 6
- [12] Tapomayukh Bhattacharjee, Advait Jain, Sarvagya Vaish, Marc D Killpack, and Charles C Kemp. Tactile sensing over articulated joints with stretchable sensors. In *2013 World Haptics Conference (WHC)*, pages 103–108. IEEE, 2013. 1.1, 2.2
- [13] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*, 2021. (document), 3.1, 3.2.2, 3.3.2, 4.1, 4.2.1, 4.2.2, 4.3.1, 4.4.1, 5.2.3, 5.4.1, 5.5.1, A.1.1, A.3.1
- [14] Raunaq Bhirangi, Abigail DeFranco, Jacob Adkins, Carmel Majidi, Abhinav Gupta, Tess Hellebrekers, and Vikash Kumar. All the feels: A dexterous hand with large-area tactile sensing. *IEEE Robotics and Automation Letters*, 2023. 5.2.3
- [15] Cristian Bodnar, Adrian Li, Karol Hausman, Peter Pastor, and Mrinal Kalakrishnan. Quantile qt-opt for risk-aware vision-based robotic grasping. In *Proceedings of Robotics: Science and Systems*, Corvallis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.075. 3.1, 3.2.1
- [16] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 4.1
- [17] Gereon H Büscher, Risto Koiva, Carsten Schürmann, Robert Haschke, and Helge J Ritter. Flexible and stretchable fabric-based tactile sensor. *Robotics and Autonomous Systems*, 63: 244–252, 2015. 2.2
- [18] Roberto Calandra, Andrew Owens, Manu Upadhyaya, Wenzhen Yuan, Justin Lin, Edward H Adelson, and Sergey Levine. The feeling of success: Does touch sensing help predict grasp outcomes? *arXiv preprint arXiv:1710.05512*, 2017. 2.2, 4.2.1
- [19] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307, 2018. 2.2, 4.2.1, 5.2.3
- [20] Giorgio Cannata, Marco Maggiali, Giorgio Metta, and Giulio Sandini. An embedded artificial skin for humanoid robots. In *2008 IEEE International conference on multisensor fusion and integration for intelligent systems*, pages 434–438. IEEE, 2008. 3.2.2
- [21] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8748–8757, 2019. 5.2.3
- [22] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition*

Letters, 34(15):2033–2042, 2013. 5.2.3

- [23] Changhao Chen, Peijun Zhao, Chris Xiaoxuan Lu, Wei Wang, Andrew Markham, and Niki Trigoni. Oxiod: The dataset for deep inertial odometry. *arXiv preprint arXiv:1809.07491*, 2018. 5.2.3
- [24] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021. 5.2.3
- [25] Tao Chen, Jie Xu, and Pulkit Agrawal. A system for general in-hand object re-orientation. In *Conference on Robot Learning*, pages 297–307. PMLR, 2022. 3.1, 3.2.1
- [26] Tien-En Chen, Shih-I Yang, Li-Ting Ho, Kun-Hsi Tsai, Yu-Hsuan Chen, Yun-Fan Chang, Ying-Hui Lai, Syu-Siang Wang, Yu Tsao, and Chau-Chung Wu. S1 and s2 heart sound recognition using deep neural networks. *IEEE Transactions on Biomedical Engineering*, 64(2):372–380, 2016. 5.5.1
- [27] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2.2
- [28] Yizhou Chen, Mark Van der Merwe, Andrea Sipos, and Nima Fazeli. Visuo-tactile transformers for manipulation. In *6th Annual Conference on Robot Learning*, 2022. 1.3, 4.2.3
- [29] Gordon Cheng, Emmanuel Dean-Leon, Florian Bergner, Julio Rogelio Guadarrama Olvera, Quentin Leboutet, and Philipp Mittendorf. A comprehensive realization of robot skin: Sensors, sensing, control, and applications. *Proceedings of the IEEE*, 107(10):2034–2051, 2019. 2.2
- [30] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 1.3
- [31] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024. 6
- [32] Keene Chin, Tess Hellebrekers, and Carmel Majidi. Machine learning for soft robotic sensing and control. *Advanced Intelligent Systems*, 2(6):1900171, 2020. 1.2, 2.2
- [33] Lillian Chin, Jeffrey Lipton, Michelle C Yuen, Rebecca Kramer-Bottiglio, and Daniela Rus. Automated recycling separation enabled by soft robotic material classification. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 102–107. IEEE, 2019. 2.2
- [34] Changhyun Choi, Wilko Schwarting, Joseph DelPreto, and Daniela Rus. Learning object grasping for soft robot hands. *IEEE Robotics and Automation Letters*, 3(3):2370–2377, 2018. 2.2
- [35] Dzung Viet Dao, Susumu Sugiyama, Shinichi Hirai, et al. Analysis of sliding of a soft fingertip embedded with a novel micro force/moment sensor: Simulation, experiment, and

- application. In *2009 IEEE International Conference on Robotics and Automation*, pages 889–894. IEEE, 2009. 3.2.2
- [36] J Dargahi. A piezoelectric tactile sensor with three sensing elements for robotic, endoscopic and prosthetic applications. *Sensors and Actuators A: Physical*, 80(1):23–30, 2000. 1.1
- [37] Fred Daum. Nonlinear filters: beyond the kalman filter. *IEEE Aerospace and Electronic Systems Magazine*, 20(8):57–69, 2005. 5.1
- [38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5.1
- [39] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5.1
- [40] Elliott Donlon, Siyuan Dong, Melody Liu, Jianhua Li, Edward Adelson, and Alberto Rodriguez. Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1927–1934. IEEE, 2018. 1.1, 2.2, 4.1
- [41] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1.2
- [42] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022. 1.3
- [43] Thomas Feix, Javier Romero, Heinz-Bodo Schmiebmayer, Aaron M Dollar, and Danica Kragic. The grasp taxonomy of human grasp types. *IEEE Transactions on human-machine systems*, 46(1):66–77, 2015. (document), 3.8
- [44] Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022. 5.3.2
- [45] Satoshi Funabashi, Gang Yan, Andreas Geier, Alexander Schmitz, Tetsuya Ogata, and Shigeki Sugano. Morphology-specific convolutional neural networks for tactile object recognition with a multi-fingered hand. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 57–63. IEEE, 2019. 3.2.1, 3.2.2, 3.5, 5.2.3
- [46] Satoshi Funabashi, Tomoki Isobe, Fei Hongyi, Atsumu Hiramoto, Alexander Schmitz, Shigeki Sugano, and Tetsuya Ogata. Multi-fingered in-hand manipulation with various object properties using graph convolutional networks and distributed tactile sensors. *IEEE Robotics and Automation Letters*, 7(2):2102–2109, 2022. 3.2.2, 3.5
- [47] Dhiraj Gandhi, Abhinav Gupta, and Lerrel Pinto. Swoosh! rattle! thump!—actions that sound. *arXiv preprint arXiv:2007.01851*, 2020. 3.2.2
- [48] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. Vector: A versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters*, 7(3):8217–8224, 2022. (document), 5.2.3, 5.3, 5.1, 5.4.2

- [49] Natalia Hernandez Gardiol. Hierarchical memory-based reinforcement learning. In *Neural Information Processing Systems (NIPS)*, volume 13, pages 1047–1053. MIT Press, 2000. 5.2.2
- [50] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5.2.3
- [51] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017. 5.2.3
- [52] Pedro Silva Girão, Pedro Miguel Pinto Ramos, Octavian Postolache, and José Miguel Dias Pereira. Tactile sensors for robotic applications. *Measurement*, 46(3):1257–1271, 2013. 1.1
- [53] Oliver Glauser, Daniele Panozzo, Otmar Hilliges, and Olga Sorkine-Hornung. Deformation capture via soft and stretchable sensor arrays. *ACM Transactions on Graphics (TOG)*, 38(2):1–16, 2019. 1.1
- [54] Karan Goel, Albert Gu, Chris Donahue, and Christopher Ré. It’s raw! audio generation with state-space models. In *International Conference on Machine Learning*, pages 7616–7633. PMLR, 2022. 5.1, 5.2.1
- [55] Ross Goroshin, Joan Bruna, Jonathan Tompson, David Eigen, and Yann LeCun. Un-supervised learning of spatiotemporally coherent metrics. In *Proceedings of the IEEE international conference on computer vision*, pages 4086–4093, 2015. 2.2
- [56] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 5.1, 5.2.1, 5.3.2
- [57] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2021. 5.1, 5.2.1, 5.3.2
- [58] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021. 5.3.2
- [59] Irmak Guzey, Yinlong Dai, Ben Evans, Soumith Chintala, and Lerrel Pinto. See to touch: Learning tactile dexterity through visual incentives. *arXiv preprint arXiv:2309.12300*, 2023. 5.2.3
- [60] Irmak Guzey, Ben Evans, Soumith Chintala, and Lerrel Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. *arXiv preprint arXiv:2303.12076*, 2023. 5.2.3, 5.5.1
- [61] Nam Ha, Kai Xu, Guanghui Ren, Arnan Mitchell, and Jian Zhen Ou. Machine learning-enabled smart sensor systems. *Advanced Intelligent Systems*, 2(9):2000063, 2020. 1.2
- [62] Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning. *arXiv preprint arXiv:2406.07539*, 2024. 4.3.2

- [63] Seunghyun Han, Taekyoung Kim, Dooyoung Kim, Yong-Lae Park, and Sungho Jo. Use of deep learning for characterization of microfluidic soft sensors. *IEEE Robotics and Automation Letters*, 3(2):873–880, 2018. 2.2
- [64] Ankur Handa, Arthur Allshire, Viktor Makoviychuk, Aleksei Petrenko, Ritvik Singh, Jingzhou Liu, Denys Makoviichuk, Karl Van Wyk, Alexander Zhurkevich, Balakumar Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv preprint arXiv:2210.13702*, 2022. 3.1
- [65] Johanna Hansen, Francois Hogan, Dmitriy Rivkin, David Meger, Michael Jenkin, and Gregory Dudek. Visuotactile-rl: Learning multimodal manipulation policies with deep reinforcement learning. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8298–8304. IEEE, 2022. 4.2.3
- [66] Evan Harber, Evan Schindewolf, Vickie Webster-Wood, Howie Choset, and Lu Li. A tunable magnet-based tactile sensor framework. In *2020 IEEE Sensors*, pages 1–4. IEEE, 2020. 2.2
- [67] L Harmon. Automated touch sensing: a brief perspective and several new approaches. In *Proceedings. 1984 IEEE International Conference on Robotics and Automation*, volume 1, pages 326–331. IEEE, 1984. 1.1
- [68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1.3
- [69] Tess Hellebrekers, Oliver Kroemer, and Carmel Majidi. Soft magnetic skin for continuous deformation sensing. *Advanced Intelligent Systems*, 1(4):1900025, 2019. 1.1, 2.3, 3.1, 3.2.2, 3.3.2, 4.2.1
- [70] Tess Hellebrekers, Nadine Chang, Keene Chin, Michael J Ford, Oliver Kroemer, and Carmel Majidi. Soft magnetic tactile skin for continuous force and location estimation using neural networks. *IEEE Robotics and Automation Letters*, 5(3):3892–3898, 2020. 1.1, 2.1, 4.2.1
- [71] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3146–3152. IEEE, 2020. (document), 1.2, 5.1, 5.2.1, 5.2.3, 5.3, 5.1, 5.4.2, 5.5.1
- [72] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 5.5.1
- [73] Takayuki Hoshi and Hiroyuki Shinoda. A large area robot skin based on cell-bridge system. In *SENSORS, 2006 IEEE*, pages 827–830. IEEE, 2006. 3.2.2
- [74] Aadithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. *arXiv preprint arXiv:2403.07870*, 2024. 4.5.1
- [75] Arash Jahangiri and Hesham A Rakha. Applying machine learning techniques to transporta-

- tion mode recognition using mobile phone sensor data. *IEEE transactions on intelligent transportation systems*, 16(5):2406–2417, 2015. 1.2
- [76] Dinesh Jayaraman and Kristen Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2016. 2.2
- [77] JR Jenner and JA1197256 Stephens. Cutaneous reflex responses and their central nervous pathways studied in man. *The Journal of physiology*, 333(1):405–419, 1982. 1.1, 4.1
- [78] Roland S Johansson. Sensory control of dexterous manipulation in humans. In *Hand and brain*, pages 381–414. Elsevier, 1996. 1.1, 1.1, 4.1
- [79] RS Johansson. How is grasping modified by somatosensory input. *Motor control: concepts and issues*, 14:331–335, 1991. 1.1, 1.1, 4.1
- [80] DongWook Kim, Junghan Kwon, Byungjun Jeon, and Yong-Lae Park. Adaptive calibration of soft sensors using optimal transportation transfer learning for mass production and long-term usage. *Advanced Intelligent Systems*, 2(6):1900178, 2020. 2.2
- [81] Taekyoung Kim, Sudong Lee, Taehwa Hong, Gyowook Shin, Taehwan Kim, and Yong-Lae Park. Heterogeneous sensing in a multifunctional soft sensor for human-robot interfaces. *Science Robotics*, 5(49), 2020. 2.2
- [82] Anna Kochan. Shadow delivers first hand. *Industrial robot: an international journal*, 2005. 1.1, 3.2.1
- [83] Nasrettin Koksal, Mehdi Jalalmaab, and Baris Fidan. Adaptive linear quadratic attitude tracking control of a quadrotor uav based on imu sensor data fusion. *Sensors*, 19(1):46, 2018. 5.5.2
- [84] Bin Kong, Yiqiang Zhan, Min Shin, Thomas Denny, and Shaoting Zhang. Recognizing end-diastole and end-systole frames via deep temporal regression network. In *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part III 19*, pages 264–272. Springer, 2016. 1.2, 5.2.1
- [85] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1.2
- [86] Ken J Kubota, Jason A Chen, and Max A Little. Machine learning for large-scale wearable sensor data in parkinson’s disease: Concepts, promises, pitfalls, and futures. *Movement disorders*, 31(9):1314–1326, 2016. 1.2
- [87] Oleksii Kuchaiev and Boris Ginsburg. Factorization tricks for lstm networks. *arXiv preprint arXiv:1703.10722*, 2017. 5.1
- [88] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016. 5.2.2
- [89] Peter J Kyberd and Paul H Chappell. The southampton hand: an intelligent myoelectric prosthesis. *Journal of rehabilitation Research and Development*, 31(4):326, 1994. 1.1,

3.2.1

- [90] Mike Lambeta, Po-Wei Chou, Stephen Tian, Brian Yang, Benjamin Maloon, Victoria Rose Most, Dave Stroud, Raymond Santos, Ahmad Byagowi, Gregg Kammerer, et al. Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845, 2020. 1.1, 2.1, 3.2.2, 3.5.2, 4.2.2, 4.4.1, 5.2.3, 6
- [91] Mark H Lee and Howard R Nicholls. Review article tactile sensing for mechatronics—a state of the art survey. *Mechatronics*, 9(1):1–31, 1999. 1.1
- [92] Michelle A Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *2019 International conference on robotics and automation (ICRA)*, pages 8943–8950. IEEE, 2019. 1.3, 4.2.3
- [93] Seungjae Lee, Yibin Wang, Haritheja Etukuru, H Jin Kim, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024. 1.3
- [94] Hao Li, Yizhi Zhang, Junzhe Zhu, Shaoxiong Wang, Michelle A Lee, Huazhe Xu, Edward Adelson, Li Fei-Fei, Ruohan Gao, and Jiajun Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. In *Conference on Robot Learning*, pages 1368–1378. PMLR, 2023. 4.2.3
- [95] Wanlin Li, Jelizaveta Konstantinova, Yohan Noh, Zixiang Ma, Akram Alomainy, and Kaspar Althoefer. An elastomer-based flexible optical force and tactile sensor. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 361–366. IEEE, 2019. 2.1
- [96] Guanhao Liang, Deqing Mei, Yancheng Wang, and Zichen Chen. Modeling and analysis of a flexible capacitive tactile sensor array for normal force measurement. *IEEE Sensors Journal*, 14(11):4095–4103, 2014. 1.2
- [97] Young-Hun Lim, Vasundara V Varadan, and Vijay K Varadan. Finite-element modeling of the transient response of mems sensors. *Smart materials and structures*, 6(1):53, 1997. 1.2
- [98] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios Mourikis, Kostas Daniilidis, Vijay Kumar, Jakob Engel, Abhinav Valada, and Tamim Asfour. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, PP:1–1, 07 2020. doi: 10.1109/LRA.2020.3007421. 1.2, 5.2.1
- [99] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020. 5.1
- [100] Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: moving average equipped gated attention. *arXiv preprint arXiv:2209.10655*, 2022. 5.6
- [101] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

5.2.3

- [102] Matthew T Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(6):418–432, 1981. 3.3.3
- [103] Matthew T Mason and J Kenneth Salisbury Jr. Robot hands and the mechanics of manipulation. 1985. 1.1, 3.2.1
- [104] Johanna L Mathieu, Stephan Koch, and Duncan S Callaway. State estimation and control of electric loads to manage real-time energy imbalance. *IEEE Transactions on power systems*, 28(1):430–440, 2012. 5.2.1
- [105] Tao Mei, Wen J Li, Yu Ge, Yong Chen, Lin Ni, and Ming Ho Chan. An integrated mems three-dimensional tactile sensor with large force range. *Sensors and Actuators A: Physical*, 80(2):155–162, 2000. 1.1
- [106] Daniela Micucci, Marco Mobilio, and Paolo Napolitano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017. 5.2.3
- [107] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1647–1654. IEEE, 2022. 1.3
- [108] Philipp Mittendorfer, Eiichi Yoshida, and Gordon Cheng. Realizing whole-body tactile interactions with a self-organizing, multi-modal artificial skin on a humanoid robot. *Advanced Robotics*, 29(1):51–67, 2015. 3.2.2
- [109] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50, 2001. 5.2.3
- [110] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020. 3.2.1
- [111] William Navaraj and Ravinder Dahiya. Fingerprint-enhanced capacitive-piezoelectric flexible sensing skin to discriminate static and dynamic tactile stimuli. *Advanced Intelligent Systems*, 1(7):1900051, 2019. 2.2
- [112] Lael U Odhner, Leif P Jentoft, Mark R Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R Ma, Martin Buehler, Robert Kohout, Robert D Howe, and Aaron M Dollar. A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752, 2014. 3.2.2
- [113] C Piazza, G Grioli, MG Catalano, and AJAROC Bicchi. A century of robotic hands. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:1–32, 2019. 3.2.1
- [114] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3406–3413. IEEE, 2016. 3.1, 3.2.1
- [115] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. In *Computer*

Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14, pages 3–18. Springer, 2016. 5.2.3

- [116] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. *arXiv preprint arXiv:2302.10866*, 2023. 5.2.1, 5.3.2
- [117] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1.3
- [118] Haozhi Qi, Brent Yi, Sudharshan Suresh, Mike Lambeta, Yi Ma, Roberto Calandra, and Jitendra Malik. General in-hand object rotation with vision and touch. In *Conference on Robot Learning*, pages 2549–2564. PMLR, 2023. 4.2.3, 6
- [119] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017. 3.1, 3.2.1
- [120] Mohammad Nasser Saadatzi, Joshua R Baptist, Zhong Yang, and Dan O Popa. Modeling and fabrication of scalable tactile sensor arrays for flexible robot skins. *IEEE Sensors Journal*, 19(17):7632–7643, 2019. 1.2
- [121] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2.2
- [122] Manfred Schütze, Alberto Campisano, Hubert Colas, Wolfgang Schilling, and Peter A Vanrolleghem. Real time control of urban wastewater systems—where do we stand today? *Journal of hydrology*, 299(3-4):335–348, 2004. 5.1
- [123] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home. *arXiv preprint arXiv:2311.16098*, 2023. 6
- [124] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023. 1.1, 1
- [125] Benjamin Shih, Dylan Shah, Jinxing Li, Thomas G Thuruthel, Yong-Lae Park, Fumiya Iida, Zhenan Bao, Rebecca Kramer-Bottiglio, and Michael T Tolley. Electronic skins and machine learning for intelligent soft robots. 2020. 2.2, 3.2.2
- [126] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006. 5.2.1
- [127] Jimmy TH Smith, Andrew Warrington, and Scott W Linderman. Simplified state space layers for sequence modeling. *arXiv preprint arXiv:2208.04933*, 2022. 5.2.1, 5.3.2
- [128] Harshal Arun Sonar, Michelle Ching-Sum Yuen, Rebecca Kramer-Bottiglio, and Jamie Paik. An any-resolution distributed pressure localization scheme using a capacitive soft sensor skin. In *IEEE RAS International Conference on Soft Robotics (RoboSoft)*, number CONF, 2018. 1.1, 2.2, 4.2.1, 5.2.3

- [129] Stefano Stassi, Valentina Cauda, Giancarlo Canavese, and Candido Fabrizio Pirri. Flexible tactile sensing based on piezoresistive composites: A review. *Sensors*, 14(3):5296–5332, 2014. 1.1
- [130] Adrian Stetco, Fateme Dinmohammadi, Xingyu Zhao, Valentin Robu, David Flynn, Mike Barnes, John Keane, and Goran Nenadic. Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, 133:620–635, 2019. 1.2, 5.1, 5.2.1
- [131] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 5.2.3
- [132] Balakumar Sundaralingam, Alexander Sasha Lambert, Ankur Handa, Byron Boots, Tucker Hermans, Stan Birchfield, Nathan Ratliff, and Dieter Fox. Robust learning of tactile force estimation through robot interaction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9035–9042. IEEE, 2019. 1.1, 4.1, 4.2.1
- [133] Subramanian Sundaram, Petr Kellnhofer, Yunzhu Li, Jun-Yan Zhu, Antonio Torralba, and Wojciech Matusik. Learning the signatures of the human grasp using a scalable tactile glove. *Nature*, 569(7758):698–702, 2019. 1.1, 2.2, 3.2.2, 6
- [134] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999. 5.2.2
- [135] Nguyen Thi Hoai Thu and Dong Seog Han. Hihar: A hierarchical hybrid deep learning architecture for wearable sensor-based human activity recognition. *IEEE Access*, 9:145271–145281, 2021. 5.1, 5.2.2
- [136] Thomas George Thuruthel, Benjamin Shih, Cecilia Laschi, and Michael Thomas Tolley. Soft robot perception using embedded soft sensors and recurrent neural networks. *Science Robotics*, 4(26), 2019. 2.2
- [137] Tito Pradhono Tomo, Massimo Regoli, Alexander Schmitz, Lorenzo Natale, Harris Kristanto, Sophon Somlor, Lorenzo Jamone, Giorgio Metta, and Shigeeki Sugano. A new silicone structure for uskin—a soft, distributed, digital 3-axis skin sensor and its integration on the humanoid robot icub. *IEEE Robotics and Automation Letters*, 3(3):2584–2591, 2018. 3.2.2, 5.2.3, 5.4.1
- [138] Matthew Trumble, Andrew Gilbert, Charles Malleon, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pages 1–13, 2017. (document), 5.2.3, 5.3, 5.1, 5.5.3
- [139] Paul Tuffield and Hugo Elias. The shadow robot mimics human actions. *Industrial Robot: An International Journal*, 2003. 1.1
- [140] Lac Van Duong and Van Anh Ho. Large-scale vision-based tactile sensing for robot links: Design, modeling, and evaluation. *IEEE Transactions on Robotics*, 37(2):390–403, 2021. doi: 10.1109/TRO.2020.3031251. 2.2

- [141] IM Van Meerbeek, CM De Sa, and RF Shepherd. Soft optoelectronic sensory foams with proprioception. *Science Robotics*, 3(24), 2018. 2.2
- [142] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1.3, 5.1
- [143] Joshua Wade, Tapomayukh Bhattacharjee, Ryan D. Williams, and Charles C. Kemp. A force and thermal sensing skin for robots in human environments. *Robotics and Autonomous Systems*, 96:1–14, 2017. ISSN 0921-8890. doi: <https://doi.org/10.1016/j.robot.2017.06.008>. URL <https://www.sciencedirect.com/science/article/pii/S0921889016307837>. 2.2
- [144] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):154, 2020. 5.2.3
- [145] Hongbo Wang, Massimo Totaro, and Lucia Beccai. Toward perceptive soft robots: Progress and challenges. *Advanced Science*, 5(9):1800541, 2018. 2.1
- [146] Shaoxiong Wang, Yu She, Branden Romero, and Edward Adelson. Gelsight wedge: Measuring high-resolution 3d contact geometry with a compact robot finger. *arXiv preprint arXiv:2106.08851*, 2021. 1.1, 2.2, 4.1
- [147] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015. 1.1, 2.2
- [148] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018. 5.2.3
- [149] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995. 5.1
- [150] Nicholas Wettels, Veronica J Santos, Roland S Johansson, and Gerald E Loeb. Biomimetic tactile sensor array. *Advanced Robotics*, 22:829–849, 2008. 3.2.2, 4.2.2
- [151] Manuel Wüthrich, Felix Widmaier, Felix Grimminger, Joel Akpo, Shruti Joshi, Vaibhav Agrawal, Bilal Hammoud, Majid Khadiv, Miroslav Bogdanovic, Vincent Berenz, et al. Trifinger: An open-source robot for learning dexterity. *arXiv preprint arXiv:2008.03596*, 2020. 1.1, 3.1, 3.2.1
- [152] Akihiko Yamaguchi and Christopher G Atkeson. Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1045–1051. IEEE, 2016. 1.1
- [153] Hang Yan, Qi Shan, and Yasutaka Furukawa. Ridi: Robust imu double integration. In *Proceedings of the European conference on computer vision (ECCV)*, pages 621–636, 2018. 1.2, 5.2.1
- [154] Hang Yan, Sachini Herath, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, and new methods. *arXiv preprint arXiv:1905.12853*, 2019. 1.2

- [155] Youcan Yan, Zhe Hu, Zhengbao Yang, Wenzhen Yuan, Chaoyang Song, Jia Pan, and Yajing Shen. Soft magnetic skin for super-resolution tactile sensing with force self-decoupling. *Science Robotics*, 6(51), 2021. 2.2
- [156] Yezhou Yang, Cornelia Fermuller, Yi Li, and Yiannis Aloimonos. Grasp type revisited: A modern perspective on a classical feature for vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 400–408, 2015. (document), 3.8
- [157] Jessica Yin, Haozhi Qi, Jitendra Malik, James Pikul, Mark Yim, and Tess Hellebrekers. Learning in-hand translation using tactile skin with shear and normal force sensing. *arXiv preprint arXiv:2407.07885*, 2024. 4.2.3, 6
- [158] Jiaxuan You, Yichen Wang, Aditya Pal, Pong Eksombatchai, Chuck Rosenberg, and Jure Leskovec. Hierarchical temporal convolutional networks for dynamic recommender systems. In *The world wide web conference*, pages 2236–2246, 2019. 5.1, 5.2.2
- [159] Ping Yu, Weiting Liu, Chunxin Gu, Xiaoying Cheng, and Xin Fu. Flexible piezoelectric tactile sensor array for dynamic three-axis force measurement. *Sensors*, 16(6):819, 2016. 1.1
- [160] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017. 2.2, 3.2.2, 3.5.2, 5.2.3, 6
- [161] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. *The International Journal of Robotics Research*, 41(7):690–705, 2022. 1.3, 4.1
- [162] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 1.3, 4.3.2
- [163] Henry Zhu, Abhishek Gupta, Aravind Rajeswaran, Sergey Levine, and Vikash Kumar. Dexterous manipulation with deep reinforcement learning: Efficient, general, and low-cost. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3651–3657. IEEE, 2019. 3.2.1