

©Copyright 2023

Ethan K. Gordon

Tractably Adaptable Food Manipulation for Robot-Assisted Feeding

Ethan K. Gordon

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

Doctor of Philosophy

University of Washington

2023

Reading Committee:
Siddhartha S. Srinivasa, Chair
Maya Cakmak
Kevin Jamieson

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Tractably Adaptable Food Manipulation for Robot-Assisted Feeding

Ethan K. Gordon

Chair of the Supervisory Committee:
Siddhartha S. Srinivasa
Computer Science and Engineering

Assistive robots can empower those with mobility impairments, engendering feelings of independence. However, to reach the point of actual in-home use, they must manage the trade-off between adaptability, tractability, and comfort. On one hand is the non-stationary distribution of the environment and user preferences; a robot may need to explore to find the best action to take. On the other hand, excessive exploration can make for an uncomfortable experience of failures and unpredictable motion. This is particularly salient for intimate tasks like feeding. Here we focus on the particular problem of food acquisition, with metrics, system design, and assumptions informed by studies with people with upper spinal cord injuries. The problem can be mapped into the well-studied contextual bandit framework to enable online adaptation with theoretical guarantees on performance, though these scale with the size of both the context space and the action space. Both can be large: set by the variety of food users want. We show how we can leverage haptic information and human expertise to shrink both of these spaces, making this online adaptation tractable. Finally, we describe a complete, portable system that can be used for an extended in-home deployment.

TABLE OF CONTENTS

	Page
List of Figures	iii
List of Tables	iv
List of Algorithms	v
Chapter 1: Introduction	1
1.1 Related Work: Food Manipulation	2
1.2 Contributions	3
Chapter 2: User-Defined Problem Formulation	4
2.1 Related Work: Robot-Assisted Feeding Systems	4
2.2 Study and System Design	6
2.3 Experiment: Is More Autonomy Always Better?	9
2.4 Findings and Defining the Bite Acquisition Problem	13
Chapter 3: Adding Adaptability: Multi-Modal Online Learning	16
3.1 Related Work: Online Learning with Bandit Algorithms	17
3.2 Preliminary: Contextual Bandits	18
3.3 Bandit Formulation for Food Acquisition	21
3.4 Experiment: On-Robot Bandit	25
3.5 Integrating Haptic Post Hoc Context	29
3.6 Validation on Synthetic Data	32
3.7 Experiment: On-Robot Post Hoc Augmented Bandit	36
Chapter 4: Tractable Action Space: Leveraging Human Data	39
4.1 Related Work: Learning Grasps from Human Demonstration	40
4.2 SPANet: Analysis of a 2D Action Space	41

4.3	Acquisition Action Schema	43
4.4	Capturing Human Bite Acquisition Strategies	46
4.5	Experiment: Action Space Evaluation	49
4.6	Experiment: Online Action Selection	52
Chapter 5:	Conclusion	55
5.1	Future Work	55
5.2	Closing Thoughts	58
Bibliography	60
Appendix A:	Portable Deployment-Ready System	75

LIST OF FIGURES

Figure Number	Page
2.1 User study participants with various interfaces	7
2.2 User study system flowchart	8
2.3 Results: Effect of autonomy and its perceived error risks.	11
2.4 Results: Level of autonomy preference vs. level of mobility	13
3.1 Contextual bandit food acquisition framework	21
3.2 Parameterized dynamic motion primitives	22
3.3 Hyper-parameter tuning in simulation	24
3.4 Results: Contextual bandit on a single unseen food type	26
3.5 Results: Contextual bandit on multiple unseen food types	27
3.6 Evolution of UCB success probability estimates over time	28
3.7 Post hoc augmented contextual bandit framework	29
3.8 Post hoc context evaluation on synthetic data	33
3.9 Post hoc context results on synthetic data	34
3.10 Results: Post hoc augmented bandit on unseen food items	37
4.1 Partial qualitative taxonomy of human food acquisition [17]	41
4.2 SPANet acquisition classification framework	42
4.3 Results: SPANet performance and generalization	43
4.4 Acquisition action schema and discrete action space pipeline	44
4.5 Human and robot food acquisition data collection setups	47
4.6 Results: Coverage of the discrete action space	51
4.7 Results: Online learning with the discrete action space	53

LIST OF TABLES

Table Number	Page
2.1 User study participant demographic information	10

LIST OF ALGORITHMS

Algorithm Number	Page
1 General Contextual Bandit	19
2 Importance-Weighted Regression	19
3 ϵ -greedy	20
4 LinUCB	20

ACKNOWLEDGMENTS

I am incredibly grateful for all of the support I have received over this almost 6 year PhD journey.

First I have to thank my advisor, Sidd Srinivasa. Robotics and especially assistive robotics represented a significant pivot for me from my undergraduate work, and but for Sidd's decision to take me in I may not be a roboticist today. His early support for conference travel and demonstrations allowed me to build my network and gain an appreciation for the breadth of the field. And his high standards for clarity and rigor not only improved my own output, but helped me develop an eye for quickly understanding the literature in general.

I also have to thank Tapo Bhattacharjee for his early leadership of the feeding project and compassionate mentorship. I will always appreciate his candor and transparency, and it has been a privilege to have been his colleague and his friend.

I appreciate the advice of the rest of my supervisory committee: Maya Cakmak, Kevin Jamieson, Kat Steele, and Brenna Argall. Their passion for HRI and depth of knowledge made this work possible.

I am grateful for all of the people who have helped on the feeding project. Amal Nanavati and Taylor Kessler Faulkner have been wonderful research partners over the past couple of years. With their passion and skill, I am confident that they can lead the project to even greater heights going forward. The work presented here would also not have been possible without all of my colleagues and co-authors: Haya, Bernie, Ramya, Sam, Xiang, Sumegh, Jaclyn, Nansong, Gilwoo, Youngsun, Ryan, Rosario, and Schmittle. I also want to thank Tyler Schrenk, our colleague, co-author, co-designer, and friend. May our work honor his memory for years to come.

I am grateful for all the friendship and support I have found in the Personal Robotics Lab. Even with our disparate research paths, we have always been united in our passion for building robots. Thank you to Taylor, Tapo, Sanjiban, Christoforos, Barnes, Kay, Schmittle, Amal, Bernie, Helen, Sid, Yunchu, Willie, Brian, Sherdil, Patrick, AVK, Gilwoo, Colin, Rockett, and Johan. Thank you to Selest Nashef, Lisa Merlin, and Anna Wehowsky for keeping the lab running through thick and thin, and of course another thank you to Sidd for bringing us together.

I am thankful for all the support I received at UW CSE. The whole graduate advising staff, and especially Elise, had an infinite well of patience for all of my questions and concerns. Thank you Les, Joe, Kristin, Ed, Stefan, and all of the other people who ran phenomenal mentorship programs, fun social events, and enabled all of the outreach efforts that put this project on the map.

Finally, I want to thank all of the family and friends who kept me sane in a new city while living through a once a century pandemic. Thank you to my housemates Cassius, Kato, and Shannon for making life at home an absolute joy. Thank you to all of my uncles, aunts, cousins, and family-friends for getting me to where I am today. Thank you to my brother Gabe and my Uncle Alan for the endless support and putting up with all the phone calls I failed to return on time. Thank you to my partner, Ari, who has shown nothing but love through all my stress and self-doubt.

And I am sure that I am forgetting many, many more people in my absent-mindedness. Behind this document is a small army of love, support, friendship, mentorship, and advising. Thank you all so much from the bottom of my heart.

DEDICATION

To my parents, Lauren and David Gordon, may their memory be a blessing

Chapter 1

INTRODUCTION

Nearly 190 million people in the world live with some form of motor impairment, according to the World Health Organization. Motor limitations have been associated with required assistance performing activities of daily living (ADLs), such as eating, bathing, and dressing [57]. Specifically, in the United States alone, approximately 1.0 million people cannot eat without assistance, according to data from 2010 [25]. The need for constant specialized care creates a large financial burden, while perceived loss of independence introduces mental health challenges for individuals with impairments [84].

On a more personal level, people who need a caregiver to help with eating express feeling self-conscious, like they are a burden, as well as potential frustration with communicating their preferences. This was captured in a recent study done by Nanavati, et al, in 2023 [100]. “Sometimes I’m not eating or I’m barely eating because I’m **self-conscious**.” “I’d have to tell [my caregiver] how to do thing... **it would just take up all the conversation**.” A robot-assisted feeding system has the potential to grant users an increased sense of independence and agency: “If I can have a robot do it, it would be **me feeding me**, and that would be a huge deal”.

This dissertation focuses on the tricky prerequisite process of actually picking food up. Food is as diverse as the human experience. Therefore, it is our assertion that any robot-assisted feeding system should have some degree of online adaptability. We should expect, over a long enough time horizon, food to show up in the user’s home that the robot has never encountered before. At the same time, collecting data on food is hard. Being geometrically deformable with non-constant topology, it is currently hard to accurately simulate. Food is fragile and can fall apart with excessive forces or stick to both the robot and the environment,

making reset procedures difficult and excessive online exploration messy and uncomfortable. Therefore, we have the need for low-data, online *tractable adaptability*.

Our key idea is that we can utilize the application-driven assumptions, problem structure, and select expert human data to reduce this broad problem to an easier, well-studied online learning framework. Combined with the engineering of a portable, wheelchair-mounted system (see Appendix A), this dissertation lays the groundwork for a minimum viable product that can be deployed in a user’s home.

1.1 Related Work: Food Manipulation

General food manipulation has been studied in various environments, such as the packaging industry [33, 45, 98, 26, 147, 21]. These tend to focus on the design of application-specific grippers for robust sorting and pick-and-place. Other work shows the need for visual sensing for quality control [28, 41, 38] and haptic sensing for grasping deformable food items without damaging them [33, 45, 98, 26, 147, 21]. Research labs have also explored meal preparation [87, 127], baking cookies [23], making pancakes [12], separating Oreos [108], and preparing meals [56] with robots. Most of these studies either interacted with a specific food item with a fixed manipulation strategy [23, 12] or with an unchanging set of food items and manipulation strategies [54, 111, 62]. Some of these studies have looked at using multi-modal data [56] but not in combination with online learning.

Food acquisition in particular is a fundamental part of robotic feeding devices [33]. Some prior work creates specialized tools for food manipulation [109, 144, 44, 145]. We focus instead on food acquisition with a fork, as it is a common household utensil with which users are familiar. Other work added variety with utensil swapping [110], which adds hardware complexity, but kept the set of food items and action trajectories small.

Further work in food acquisition uses vision or haptic data to improve the choice of acquisition action. Vision can be used to classify visually different food items, and haptic data can assist in identifying foods that look different but require similar actions (e.g. grapes and cherry tomatoes). Some prior works require additional “probing” actions for every food

item [117, 56, 17] or specialized sensors beyond force torque sensors [150]. Other work uses haptic data to improve food acquisition during the feeding process, but the expert-designed action space does not cover the variety of food necessary for in-home applications [128].

This dissertation expands upon the wealth of work done in the food manipulation space to focus on picking up a wide enough variety of food (including previously-unseen food) with a fork for a successful in-home deployment.

1.2 Contributions

This dissertation makes the following contributions in the field of food manipulation for robot-assisted feeding:

- An exploration of the need for autonomy in assistive feeding and a determination of success metrics informed by a user study with people with upper body mobility limitations (Chapter 2, also published in Bhattacharjee, Gordon, et al. [16])
- A contextual bandit framework for the food acquisition problem (Chapter 3, also published in Gordon et al. [58])
- A method for augmenting a contextual bandit algorithm with (in this case, haptic) *post hoc context* for faster learning (Chapter 3, also published in Gordon, et al. [59])
- A schema for acquisition actions and a discrete set of actions that can cover a wide variety of food items (Chapter 4, also published in Gordon, Nanavati, et al. [60])
- Design of a portable robot system ready for in-home deployment (Appendix A)

Chapter 2

USER-DEFINED PROBLEM FORMULATION

Before diving into the technical problem of autonomous food acquisition, we first need to demonstrate the need for autonomy and define the success metrics relevant to the user population. To that end, this chapter covers a user study where people with various degrees of upper-body mobility impairment were invited to try a simplified version of the robot-assisted feeding system covered in more detail in Appendix A. We tested the hypothesis that varying levels of autonomy would create a trade-off between user comfort and effort. On one hand, a minimally-autonomous system requires more effort from the user, but allows greater control and customization. On the other, a fully autonomous system requires less effort, but is prone to errors that could lead to a reduced sense of control. This trade-off could be affected by the user’s level of mobility impairment as well as the success rate of the autonomous system.

This study also investigated variations in user interface and robot speed, but that is outside of the scope of this work. We direct interested readers to our full publication [16]. The key findings from this study that are relevant to food acquisition are discussed in Section 2.4.

2.1 Related Work: Robot-Assisted Feeding Systems

While specialized feeding devices for people with disabilities have been introduced on the market [105, 99, 95, 94, 148, 14, 96, 103, 11], they lack widespread acceptance. Numerous researchers have also developed robot-assisted feeding systems, either as table-mounted robotic arms [30, 126, 119, 107, 114, 113, 63, 70, 86, 132, 140, 71, 136, 137, 153, 152, 130], general purpose mobile manipulators [17, 116, 111], or wheelchair-mounted robotic arms [19, 5, 43, 64, 88, 48, 54]. A comprehensive review of meal assistance robots is given in [102, 135, 77, 27].

Some assistive systems focus more on support or adaptation to the user. For example, several emphasize tremor cancellation during the feeding task [106, 83, 112], while others provide arm support for users who can move their arm but do not have sufficient muscular strength for continued large movements [138, 69, 104]. Among some of the developed assistive feeding systems based on autonomous robots, Park et al. [111] used a general purpose manipulator to scoop yogurt with a spoon communicating with the user via a web interface. Ettehadi and Behal [47] used learning from demonstration to learn scooping trajectories for robotic feeding. Other examples of robotic systems which can skewer solid food items with a fork are Herlant [62], who also explored modeling an intelligent bite-timing model in social dining scenarios, and Gallenberger et al. [54], who developed a system that can detect a solid food item specified by a user and pick it up autonomously. Both these studies did not develop an explicit user-interface and did not test the system with people with mobility impairments. Also notable is the work by Canal et al. [29], who developed an intelligent personalization framework for robot-assisted feeding systems that can adapt to user preferences.

Full autonomy, as attempted by the studies above, comes with various challenges. As studied in [10], using a 10-point taxonomy for categorizing robot autonomy levels, there are various levels of robot autonomy which may be appropriate in different scenarios. Kim et al. [73] discussed how autonomy impacts performance and satisfaction for users with Spinal Cord injuries. They looked at full teleoperation mode to full autonomy mode without any errors in pushing and reaching tasks. Javdani et al. [71] looked at shared autonomy when compared to direct teleoperation and full autonomy for the bite acquisition of marshmallows. However, there is hardly any work on analyzing the effect of autonomy with perceived error risks and variables such as robot speeds and interfaces on robot-assisted feeding systems for user acceptance. The work in this chapter aims towards closing this gap by exploring multiple variables and analyzing the effect of perceived error risks on robot autonomy.

2.2 Study and System Design

2.2.1 Modes of Feeding: Autonomy and Speed

Our robotic system can switch between different modes of operation. For speed, our robot arm can move in either the slow setting (0.2 rad/s for each joint during most motions, 0.1 rad/s for each joint during approach to face) or in fast setting (0.8 rad/s for each joint during most motions, 0.2 rad/s for each joint during approach to face). For autonomy, once a food item is selected by the user, our robot can function in any of the following modes:

Full Autonomy - FA : At the start of a fully autonomous feeding trial, the user can select a specific food item on a plate or have the robot choose one randomly using an interface. The robot arm moves to a pre-determined configuration above the plate facing down to see the entire plate using the wrist-mounted camera. Depending on what food item the user selects, it perceives the food items on the plate using perception algorithms [53] and decides what is the best strategy to pick up the one selected. Once the algorithms determines the strategy, the robot arm servoed to it using feedback from the visual modality, acquires the bite using feedback from the haptic modality [17], and then moves up to a pre-determined configuration to perceive the user's face. The robot detects the user's intent to eat when it perceives the user's mouth open using the wrist-mounted camera, which is facing the user sitting on the wheelchair. This is based on our initial discussions when the caregivers mentioned that they look for the mouth-open cue to determine when to approach to feed the care-recipients [15]. Once the robot detects the 3D position of the face and sees the mouth open, it approaches. The robot-arm then determines the best strategy [53] to feed the food item on the fork so that it is easy for the person to take a bite. Once the robot-arm has reached its final position, it communicates to the user that the food is ready to eat and the user takes a bite. The robot waits for a fixed duration of time. The feeding trial ends once the robot moves away from the mouth and goes back to its home position.

Low Autonomy - LA : Similarly to FA, the feeding trial starts with a user selecting a food item. The robot then moves to a pre-determined configuration above the plate to see the plate and perceives the food item on the plate using perception algorithms [53]. However, instead of selecting the best pick-up strategy by itself, it asks the user to choose an action to acquire the food item. Since this study only deals with solid fruit pieces, there are two options available to the user: *vertical* skewering (VS) with the fork handle orthogonal to the table, and *tilted* (TV) skewering with the tines of the fork orthogonal to the table. Once the user selects one option, the robot performs the action and picks up the food. Then, it moves to the pre-determined configuration to focus on the user's face. The robot then asks the user to select the best strategy to transfer the food item to their mouth and when to feed them. There are two options for choosing a strategy for transferring a bite: *horizontal* and *tilted* transfer. Once a user selects a transfer strategy and asks the robot to feed, the robot approaches the person's mouth. Once the robot has reached its final position, it communicates to the user that the food is ready to eat and the user takes a bite. The feeding trial ends with the robot moving away from the mouth and returning to its home position for next trial.



Figure 2.1: Participants with mobility impairments with options for the voice interface using Alexa and web-interface displayed in their/our tablets mounted in different positions according to their preference or constraints of system integration. When needed a switch was installed to control the web interface.

Partial Autonomy - PA: : In partial autonomy mode, we maintain any one of the three phases of feeding autonomous while the rest are non-autonomous. In the *autonomous acquisition - AAC* mode, the bite acquisition phase is autonomous and the robot decides the

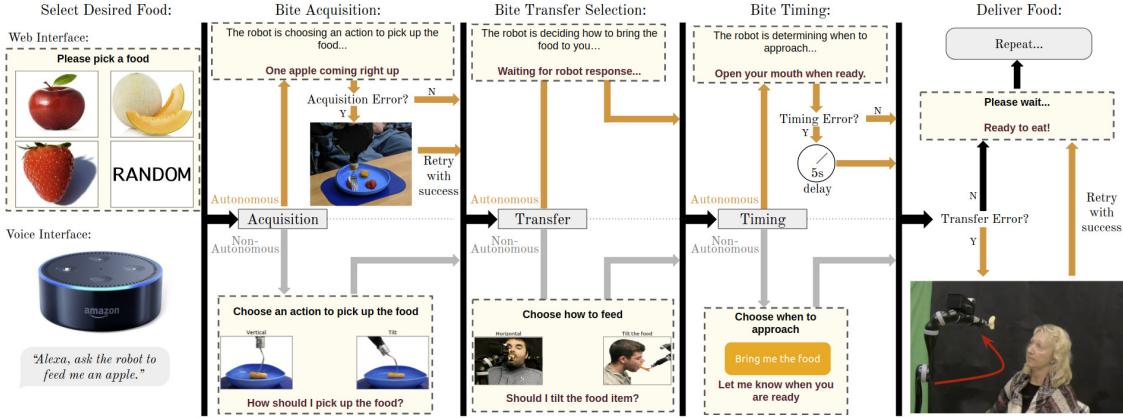


Figure 2.2: Each phase of interaction and error recovery is depicted as a column in the system flow chart. Orange arrow branches indicate flow under the *autonomous* condition for each phase while gray arrow branches indicate *non-autonomous*. Each dashed box shows what the user sees on the web GUI, depending on the phase condition. The dark red text in each yellow box indicates the robot’s current status. In the ‘voice interface’ condition, the robot speaks the red text rather than displays it.

best strategy to pick up the food item that the user selected using its algorithms [53]. In the *autonomous timing - ATI* mode, the bite timing phase is autonomous and the robot decides when to approach the user’s face based on whether the mouth is open or not. Detecting the face and whether the mouth is open or not is not trivial choice for automation given the different angles in which the faces are oriented for this target population as well as possible occlusion from the food on the fork itself. In the *autonomous transfer - ATr* mode, the bite transfer phase is autonomous and the robot decides the best strategy to transfer the food item that the user selected using its algorithms [53] during bite transfer.

2.2.2 Interface and Safety

We provide two interface variants to the users: bi-directional voice and bi-directional web GUI (see Figure 2.1). Both of these interfaces follow the same procedural flow shown in Figure 2.2. For the web GUI specifically, we gave participants the ability to access it via their personal tablet device, allowing interaction through their preferred adaptive button or

touch input. If the user does not have their own personal device, we provide a stand-mounted tablet adjacent to the plate of food. The user interacts with this tablet via a specialized button [134] they can press with minimal movement.

We implemented three levels of safety in our system. First, a tight collision box is positioned around the user’s face to ensure collision avoidance. We have also implemented a low force threshold on the instrumented fork and the robot behaviors are designed such that it will stop the moment it touches anything on its way to the face. Finally, we have an operator sitting close to the robot throughout the experiment who can stop the robot if unwanted behavior is detected.

2.3 Experiment: Is More Autonomy Always Better?

It is inevitable that an autonomous robotic system will have errors when used in real-life for long term. We designed the study to see the participants’ autonomy preferences with perceived error risks by deliberately introducing errors in some of the autonomous trials. We designed the study such that in trials with errors, the robotic system recovered from the error autonomously after one try.

2.3.1 Participants

For this study, we recruited ten participants, eight male and two female, between the ages of 28 and 57. We provide some details on their type of mobility impairment in Table 2.1 and the modifications made to the experimental setup to facilitate their participation.

Given the range of mobility limitations in the participant pool, we grouped participants based on it: six of the participants showed higher mobility limitations and the remaining four showed lower mobility limitations. Mobility level was also considered as a factor while evaluating their responses and ratings to the system.

Two of our higher mobility limitation participants had little to no neck movement. This required an additional calibration of the system to the final phase of food transfer to occur inside their mouth. These participants used devices controlled by sip and puff. Depending

Table 2.1: Participants self-reported mobility limitation description and grouping based on mobility (lower (L) vs higher (H) mobility limitation)

Participant	Age	Self-reported mobility limitation	Group
P1	34	C1 quadriplegia	H
P2	38	C2 quadriplegia	H
P3	57	C5-C6 complete	L
P4	40	Arthrogryposis	L
P5	31	C1 tetraplegia	H
P6	33	C1 tetraplegia	H
P7	37	C1-C2 quadriplegia	H
P8	28	C3-C4 quadriplegia	H
P9	45	C4 incomplete	L
P10	38	C4-C5 complete	L

on their mobility limitation, some participants used a push button with their heads, or used their own arms while grasping a tool to touch the tablet's screen.

Some of our participants were more proficient with technology than others. One of the participants was an expert Tecla user with access to a Kinova arm with no autonomy features. This participant shared his experience of teleoperating the robot (fully non-autonomous) to pick one piece of fruit and bring it to his mouth would take him approximately 45 minutes. Other participants did not use a particular electronic device with assistive features, so they had to familiarize themselves with the one provided in the lab.

Additionally to our in-person participants, we gathered responses from 8 additional participants using a video questionnaire, where an actor was used to showcase the different autonomous modes and the associated errors in multiple feeding trials. Online participants' age range was between 25 - 77 years, 7 male and 1 female, with mobility impairments like spinal cord injuries, loss of strength in joints, or shaky hands that impede them from feeding

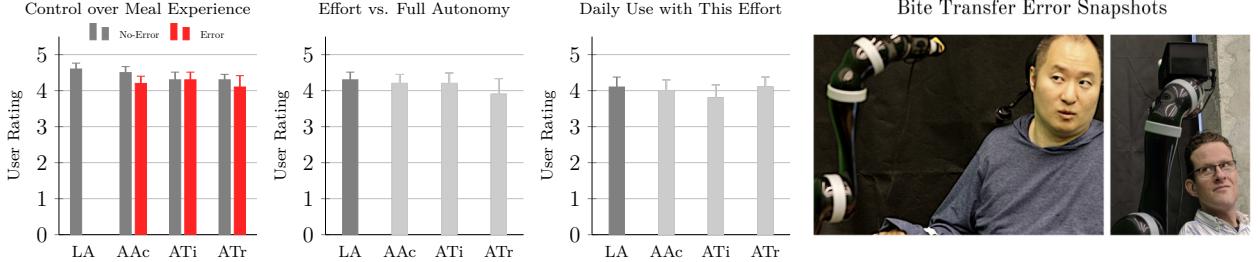


Figure 2.3: Effect of autonomy and its perceived error risks. Participants consider *low-autonomy* trials to require more effort compared to *fully autonomous* trials and partial autonomy does not help reduce the effort. Also, participants do not have a clear preference to use partially autonomous trials when compared with *low-autonomy* trials. Additionally, with autonomy comes potential risk of errors, and participants may have a negative preference for errors that *low-autonomy* trials may have higher perceived control over the meal experience compared to trials with erroneous partial autonomy. Generally, bite transfer errors may be penalized higher than other types of errors.

themselves. Due to the online nature of the questionnaire, the videos were presented in the same order for all online participants.

2.3.2 Methods and Procedures

The study was conducted with the approval of our university's Institutional Review Board (IRB). Before scheduling each participant, we provided an initial recruitment questionnaire that confirmed if the participants identified as people with mobility impairments and if they currently used a power wheelchair. We collected details relevant to the design of our experiment: confirming they could use both selected interfaces (*voice* and *web*), any food allergies to determine what items to include, and how to acquire consent. We selected three fruits that the robot would feed the participants during the experiment: 1) apples, 2) strawberries, and 3) cantaloupes. We selected a limited number of items to reduce uncertainty in perception and manipulation, as those factors are outside the scope of this particular study.

Due to challenges related to transferring the participants from their wheelchairs to our robot-on-wheelchair system, a modification to the system was introduced. The participants'

comfort in their customized chairs was of paramount importance, given that they would spend a long period of time in our laboratory. Mounting the robotic arm on each participant’s chair was also challenging given different mounting requirements for different wheelchairs. Hence, the robotic arm was mounted on a tripod at the same height of where it would be mounted on the wheelchair. In a real home environment for long-term use, the robot arm would be mounted in their wheelchairs with all the required customizations.

Once the participants came to the laboratory, we went over the consent form together with the caregiver. We then asked the participants to position their wheelchair facing the table. We positioned the tripod near the wheelchair and based on this, we re-calibrated the experimental setup. There was an initial setup time in our experiment where the participants familiarized with the use of both interfaces. All participants required different accommodations in terms of using their or our own devices to display the web interface, as well as setting up a *Tecla* device and a scanning/switch button on their wheelchair to navigate and select in the web interface. The initial setup finished going through the study protocol with both the participant and their caregivers, as well as the safety measures.

We designed this study to have 7 trials total presented to the participants in four blocks with different modes of autonomy (See Section 2.2.1). The *low-autonomy* trial was considered a block by itself. One block had the *autonomous acquisition* trial followed by *autonomous acquisition with error*. Another block had the *autonomous transfer* trial followed by *autonomous transfer with error*. And the fourth block had the *autonomous timing* trial followed by *autonomous timing with error*. The four blocks were presented at a random order across the in person participants.

The speed of the robot was fixed to the *fast* setting (i.e., robot joint speeds were limited to 0.8rad/s), the dining scenario was *individual* (i.e., only the participant was at the table) and the *web* interface was used for all trials. After each trial, the participants were asked questions about their preferences of autonomy and effort, their preference given errors and general technology acceptance questions based on the TAM model [36].

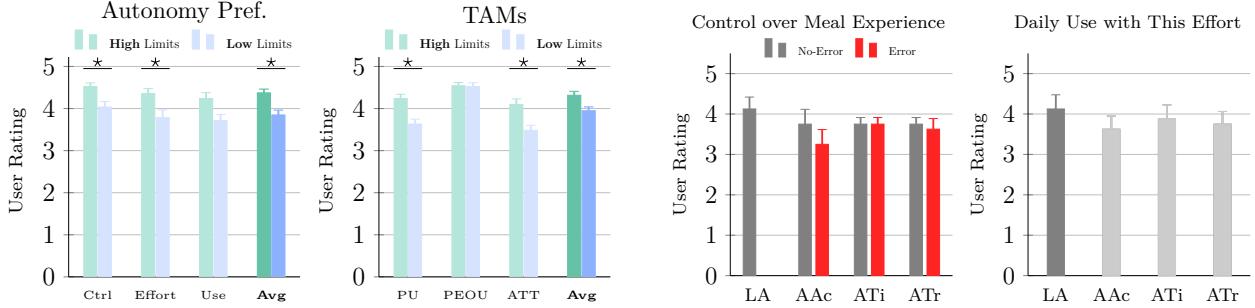


Figure 2.4: (*Left*) Findings on autonomy preference and TAM responses based on mobility grouping. Statistical significance was found for control, effort and the average over autonomy related questions; additionally, all TAM sub-groups were found significantly different except perceived ease of use (PEOU). (*Right*) Results from online participants show preference to versions without error compared to with error. Also, participants agree that the *low-autonomy* version provides more control over the meal experience than the partial autonomy versions with error.

2.4 Findings and Defining the Bite Acquisition Problem

A General Linear Model was fitted for each dependent variable using *Mobility* and *Autonomy Level* as main factors. Interactions between factors was found non-significant and the following analysis focuses on the main effects.

More autonomy is not always better: Effect of errors Figure 2.3 shows the results. Even though the current experiment did not find any statistically significant differences between autonomy levels, there were trends. Participants usually preferred trials with no error compared to those with error when sufficient control over their meal experience or potential daily usage was considered. However, they *agree* that the *low-autonomy* trial requires more effort than the *fully autonomous* trials experienced earlier in the study (see [16]). Notably, partial autonomy does not help in reducing the effort. In terms of the *mobility* factor, as shown in Figure 2.4(Left), participants with higher mobility impairments gave significantly higher scores on items related to autonomy preference than participants with lower mobility impairments ($p < 0.03$). Interestingly, the participants still *agree* that

they would use the robot with the increased effort with preference for the *low-autonomy* trial when compared to trials with partial autonomy. When asked about the percentage of errors during a meal that they would tolerate with increased autonomy, on an average they mentioned that they would be fine with bite-acquisition errors happening around 30% of the time but they penalized the bite-transfer errors slightly higher at a rate of 25%. Notably, none of the participants perceived the error in bite-timing in which the approach was delayed by 5s after detecting the mouth-opening. When responding to the post-task questionnaire, P1 mentioned that "I didn't like having it do step by step. I wanna say give me food and have it fly down there and give me food". P1 also made a remark regarding the lack of continuity of the feeding process from the robot side. P3 supported that statement with a similar one "Picking things up and selecting was slow (...) there was a lot of thinking going on"

Participants agree to the technology's perceived acceptance and usage Mobility was found as a significant factor for TAM responses. As shown in Figure 2.4(Left), participants with higher mobility limitations gave significantly higher scores on perceived usefulness (PU), attitude (ATT), and on the average of all 10 TAM questions (Avg). The participants' preference for our robotic system in general is supported by the following statements gathered during the post task questionnaires: "There seem to be some limitations (...) it still wouldn't stop me from using it, let me be clear. Totally beats going hungry" (P2). "Overall great work, it was fun to use. None of the errors bothered me, especially when it didn't pick something up (...) I liked that it went ahead and it tried something new every time so that's great" (P4).

Online findings Figure 2.4(Right) shows the results for the online responses. The current experiment did not find any statistically significant differences but the ratings show an overall decline when errors occurred. Also, similar to the in-person study, online participants usually preferred low-autonomy when compared to partial autonomy with errors especially

for autonomous bite acquisition errors. However, note that the online participants experienced the trials from a 3rd-person viewpoint. Interestingly, among partial autonomy trials with no errors, *ATi* received the highest ratings. This could be because detecting the mouth opening is a clear sign of automation that is easily perceived even through video.

Acquisition Metrics In general, despite the inherent limitations of a single-utensil, fork-only end-effector, all participants gave the feeding system fairly high marks in terms of Technology Acceptance Metrics (TAMs). This implies that, at least for a minimum viable product, the fork-only acquisition is sufficient. Based on this study, this system can be evaluated in terms of time and success:

- Time Metric: Our actions took on the order of 90s to complete in the *fast* setting. This was sufficient for all participants regardless of mobility limitation. All acquisition attempts should take on the order of this length of time or shorter.
- Success Metric: Averaging across in-person users, an acquisition success rate of $\sim 70\%$ was sufficient for daily use, with users with higher mobility limitations tolerating a lower success rate and users with lower mobility limitations preferring a higher success rate. Averaging with online participants, we will use a success rate of 80% as our target.

As a note, the bite transfer errors, where the robot moved the most erratically, required a higher success rate. This implies that the robot motion should generally be predictable for maximum comfort, even if safety is guaranteed. The next two chapters cover how the robot can adapt and acquire previously unseen food items by trying to hit these metrics as soon as possible.

Chapter 3

ADDING ADAPTABILITY: MULTI-MODAL ONLINE LEARNING

The next two chapters focus on the specific sub-problem of food acquisition. As shown in the previous chapter, there is some allowance for robot failures in all stages of food manipulation. However, the threshold of 80% success rate for acquisition is difficult to hit consistently given the wide variety of food types that the robot may encounter in the home.

Different food items require different manipulation strategies [17]. While recent work has achieved some successes in developing strategies that can acquire a variety of food items [49, 53], it is unclear which strategy works best for previously unseen food. Even food items that look similar, such as ripe and un-ripe banana slices, can have very different consistencies, leading to different bite acquisition strategies. One option is to collect a wide variety of food items and train an end-to-end acquisition model in the lab in advance. We argue that this approach is likely untenable due to the shear variety of food a person could want to eat, even in the limited case of bites that can be acquired with a fork. As an example, a conventional grocery store alone typically stock in excess of 40,000 products [89].

Another approach could be to collect the data autonomously through extensive simulation. While promising in the long term, current food simulation is still a developing field. Existing simulation technologies that can handle the topology-breaking present in skewering and cutting motions generally focus on specific applications like organ modeling for surgery [34] or are limited to specific motions such as vertical cutting [61].

Given all of these limitations for pretraining models in a laboratory environment, we argue that online learning is a necessity for any feeding system deployed long-term in a user’s home. Such a system needs to have a constrained space of possible acquisition actions, to make sure

that the robot motion is generally predictable. How to design such a space is discussed in the next chapter. This chapter assumes that such a space of actions already exists and focuses on the learning problem. We model food acquisition as a linear contextual bandit, leveraging both visual context and post hoc haptic context to allow the robot to learn how best to pick up food it may have never seen before in a reasonable amount of time.

The work in this chapter was also in two publications [58, 59].

3.1 Related Work: Online Learning with Bandit Algorithms

Bandit algorithms have seen widespread success in online advertising [133, 24], health interventions [75, 65], clinical trials [123], adaptive routing [8], education [90], music recommendations [143], financial portfolio design [121], and any application requiring a more optimized version of A/B testing. Adoption in robotics has been more limited, e.g., to selecting trajectories for object rearrangement planning [76], kicking strategies in robotic soccer [97], and, perhaps most closely related, selecting among deformable object models for acquisition tasks [93]. Unlike previous work, we argue that it is untenable to construct deformable object models for every food item, as conventional grocery stores typically stock in excess of 40,000 products [89]. Instead, this thesis explores a model-free approach that operates directly on the image context space.

No-regret algorithms for solving bandit problems include UCB [6] and EXP3 [7] for stochastic and adversarial reward distributions, respectively. They were also extended to the bandits-with-expert-advice setting (a generalization of the contextual bandit problem for small policy classes) with EXP4 [7]. Baseline methods for the contextual bandit problem include epoch-greedy [78] and greedy [9], both of which are simple to implement and perform well in practice, although they do not achieve optimal regret guarantees. More recent advances include LinUCB [82, 146], RegCB [51] and Online Cover [2], a computationally efficient approximation to an algorithm that achieves optimal regret. For a recent and thorough overview, we refer the interested reader to [79, 20].

The post hoc augmented contextual bandit presented later in this chapter is distinct

from bandits with delayed feedback [139] in that the post hoc context is not delayed by any time steps, but just observable after action selection. This work could potentially be compared with the bandits-with-expert-advice setting and associated algorithms like EXP4 [7], in the sense that the context model and the post hoc context model could be thought of as competing action recommendations, though the experts in this setting generally make their predictions exclusively prior to action selection.

3.2 Preliminary: Contextual Bandits

General online supervised learning has an agent learn a map $f : \mathbb{R}^{d_c} \rightarrow \mathbb{R}^K$ between a d_c -dimension context vector c and a K -dimension loss vector l given a sample (c_t, l_t) at each time step t . In a discrete interactive learning setting, the agent will first observe the context c_t , choose an action $a_t \in [K]$, and then observe the full loss vector l_t while incurring loss $l_t[a_t]$. The agent's goal in this setting is to minimize *cumulative regret*

$$R_T := \max_{a'} \sum_t^T (l_t[a_t] - l_t[a']) \quad (3.1)$$

the difference between the loss incurred by the agent and the lowest loss it was possible to incur.

In a contextual bandit setting, the agent is restricted to *bandit feedback*: observing only the loss incurred ($l_t[a_t]$) rather than the full loss vector l_t . This creates a trade-off between exploring actions we are unsure about and exploiting actions likely to incur little loss. In general, a contextual bandit algorithm consists of two parts: (1) an *exploration strategy* that determines which action to take at each time step given c_t and some policy $\pi : c_t \rightarrow a_t$, and (2) a *learner* that incorporates the bandit feedback received into the π . Algorithm 1 presents this structure with the addition of a feature extraction function.

3.2.1 Learning: Importance-Weighted Linear Regression

Assume that the true map f^* exists in some function class \mathcal{F} . One method for solving the contextual bandit setting is to reduce the problem to regular online supervised learning and

create an estimate of this function \hat{f} with least squares regression. Importance weighting [20] using inverse propensity scores can eliminate the bias that comes from only using partial feedback from random exploration.

Under the assumption that f^* is linear and all observed noise is Gaussian, such that $l_t[a] = \theta_a^\top c_t + \epsilon$ with weights $\theta_a \in \mathbb{R}^{d_c}$ and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I}\sigma^2)$, least squares regression can be applied. This produces familiar weight estimates

$$\widehat{\theta}_a = \arg \min_{\theta_a} \sum_{t=0}^{T_a} \frac{1}{\mathbb{P}_t(a_t)} (\theta_{a_t}^\top \phi(x_t) - l_t(a_t))^2 = (\mathbf{C}_a^\top \mathbf{C}_a)^{-1} \mathbf{C}_a^\top L_a \quad (3.2)$$

where $\mathbf{C}_a \in \mathbb{R}^{T_a \times d_c}$ is the matrix contexts observed during the T_a time steps where the agent selected action a , $L_a \in \mathbb{R}^{T_a}$ is the vector of (inverse propensity weighted) scalar losses observed on those same time steps. A policy checkpoint π_t can be stored by caching the data matrix $\mathbf{C}_a^\top \mathbf{C}_a \in \mathbb{R}^{d_c \times d_c}$ and the action-loss vector $\mathbf{C}_a^\top L_a \in \mathbb{R}^{d_c}$, and these can be updated each round as described in Algorithm 2. For notation purposes in this work, \mathbf{C} and L without an action subscript are used to identify data matrices and loss vectors calculated from all data, not just data collected for a single action.

Algorithm 1:
General Contextual Bandit

Input: (optional) Featurizer ϕ ,

Environment E

Initialize Context $x \in \mathcal{X} \sim E$

for $t = 1, \dots, T$ **do**

Find features $\phi(x)$

$\mathbb{P}_t \leftarrow \text{explore}(\phi(x))$

Sample action $a_t \sim \mathbb{P}_t$

Receive loss $l_t \sim E|a_t$

learn($\phi(x), a_t, l_t, \mathbb{P}_t$)

Re-sample context $x \sim E$

Algorithm 2:
Importance-Weighted Regression

Input: L2 Regularization λ ,

of Features d

Initialize $\pi_0: \forall a \in \mathcal{A}$:

$\mathbf{C}_a^\top \mathbf{C}_a \leftarrow \lambda \mathbf{I}_{d_c \times d_c}; \mathbf{C}_a^\top L_a \leftarrow \vec{0}$

Function **learn**($\phi(x), a_t, l_t, \mathbb{P}_t$):

$\mathbf{C}_a^\top \mathbf{C}_a \leftarrow \mathbf{C}_a^\top \mathbf{C}_a + \frac{1}{\mathbb{P}_t(a_t)} \phi \phi^\top$

$\mathbf{C}_a^\top L_{a_t} \leftarrow \mathbf{C}_a^\top L_{a_t} + \frac{l_t}{\mathbb{P}_t(a_t)} \phi$

$\pi_{t+1} \leftarrow (\mathbf{C}^\top \mathbf{C}, \mathbf{C}^\top L)$

3.2.2 Exploration: ϵ -greedy

One of the simplest approaches to exploration is the ϵ -greedy algorithm, shown in Algorithm 3. This algorithm opts for the optimal action based on previous observations with probability $(1 - \epsilon)$ and explores all actions uniformly with probability ϵ . We consider both purely greedy ($\epsilon = 0$) and exploratory ($\epsilon > 0$) variants.

With arbitrary contexts, the ϵ -greedy algorithm (with optimized ϵ) has a cumulative expected regret bound $\mathbb{E}[R_T] \leq \tilde{O}(T^{2/3})$, though it can perform well empirically [20]. Repeated contexts on failure also enables a better regret bound since taking multiple actions can provide effectively better-than-bandit feedback for a given context.

Algorithm 3: ϵ -greedy	Algorithm 4: LinUCB
Input:	Input: Width parameter α
Exploration parameter	Function <code>explore</code> ($\phi(x)$):
$\epsilon \in [0, 1]$	for $a \in \mathcal{A}$ do $lcb_a \leftarrow \hat{\theta}_a^\top \phi(x) -$ $\alpha \sqrt{\phi(x)^\top (\mathbf{C}_a^\top \mathbf{C}_a)^{-1} \phi(x)}$ $\mathbb{P}_t(a) \leftarrow \mathbf{1}\{a = \arg \min lcb_a\}$
Function <code>explore</code> ($\phi(x)$): $\mathbb{P}_t(a) \leftarrow$ $\frac{\epsilon}{K} + (1 - \epsilon)\mathbf{1}\{\pi_t(\phi(x))\}$	

3.2.3 Exploration: LinUCB

The other algorithm we use is Linear Upper Confidence bound (LinUCB) [146], presented in Algorithm 4. LinUCB implicitly balances exploration and exploitation by using the estimated linear model to construct a confidence interval around l_t for a given context c_t and optimistically playing action a_t with the lowest lower confidence bound on its expected loss (or, identically, the highest upper confidence bound on its expected reward). In this way, the algorithm prefers relatively unknown actions with larger intervals (encouraging exploration) and actions with low loss (encouraging exploitation). UCB-style algorithms like this are known to achieve cumulative regret bounded by $\mathbb{E}[R_T] \leq \tilde{O}(d_c K \sqrt{T})$ [1].

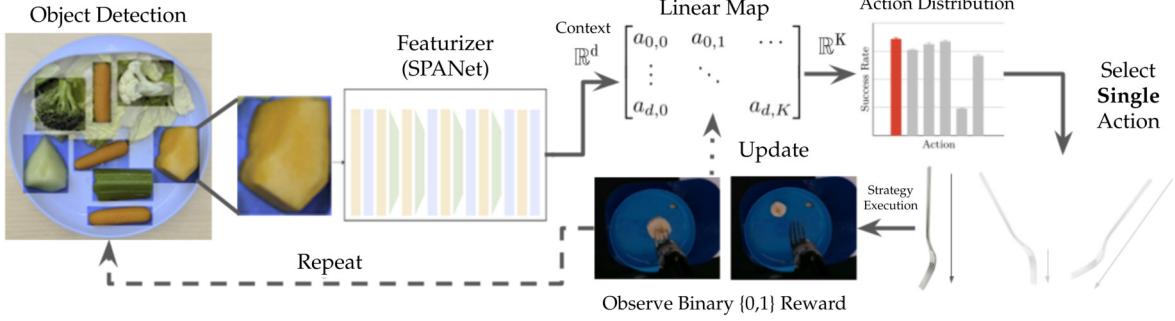


Figure 3.1: Contextual Bandit Food Acquisition Framework. SPANet [49] is trained on previously seen food items, and then all but the last layer is frozen as a featurizer. The final linear layer becomes the “linear map” that is updated after each subsequent attempt. The result is the estimated success rate of each action on the given food item. A single action is attempted before updating the linear map.

For a given confidence level, this lower bound can be calculated as [146]

$$LCB(a) = \hat{\theta}_a^\top c_t - \alpha \sqrt{c_t^\top \Sigma_a c_t} \quad (3.3)$$

for some constant $\alpha > 0$ and $\Sigma_a := (\mathbf{C}_a^\top \mathbf{C}_a)^{-1}$ the covariance of the estimator $\hat{\theta}_a$ and the inverse of the data matrix.

3.3 Bandit Formulation for Food Acquisition

Work highlighted in Section 4.2 demonstrates that a model could accurately recommend an optimal dynamic motion primitive for a given food item in a full supervised learning setting. This feed-forward neural network, SPANet [49], takes in a cropped image of a given food item and, for each primitive, outputs its expected success rate. The last layer of this network was linear. Therefore, we are motivated to freeze all but the last layer of a pretrained model, using the activation of the penultimate layer as features $\phi(x)$. The last layer can be updated in an online fashion for new food items as described in Section 3.2.1.

In a meal setting, we operate under the assumption that the user is provided with a plate of food already prepared such that individual bites can be acquired with a single fork. This can include discrete solid food items (e.g. pieces of fruit) or a continuous food item that will not slip through the fork tines (e.g. mashed potatoes). As described in Fig. 3.1, at each round $t = 1, \dots, T$, the interaction protocol consists of:

1. *Context observation.* The user selects a food item to acquire (in this chapter, an instance of RetinaNet [85] is used to detect objects and segment into bounding boxes). We observe an RGBD image containing the single food item. We pass this through SPANet and use the penultimate layer as the context features $\phi(x_t) \in \mathbb{R}^{d_c}$. The RGBD image is also used to localize the object for execution of the action.
2. *Action selection.* The agent selects one motion primitive $a_t \in \mathcal{A} = \{1, 2, \dots, K\}$. In this initial implementation, $K = 6$, with 3 pitch angles and 2 roll angles (parallel and perpendicular to the food), as shown in Figure 3.2. The robot always skewers the center of the food item.
3. *Partial loss observation.* The environment provides a binary loss $l_t(a_t, \phi(x_t)) \in \{0, 1\}$, where $l_t = 0$ corresponds to the robot successfully acquiring the single desired food item.

Each manipulation strategy is executed by a simple impedance controller with 3 parameters: (1) the angle of the fork handle relative to vertical, (2) the roll angle of the fork relative to the major axis of the food item, and (3) the maximum force to impart on the food

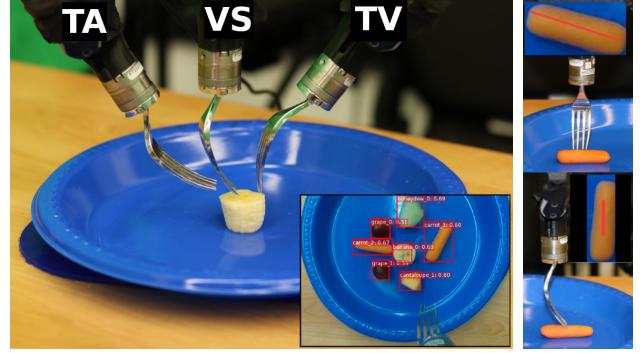


Figure 3.2: Dynamic motion primitives parameterized by fork pitch (*left*) and roll angle (*right*), creating an action space of size $K = 6$.

during skewering. Specifically, the 6 discrete actions in this setting, motivated by previous qualitative work [17], are parameterized as follows

1. *Vertical Skewer 0 (VS0)*: Pitch 0, Roll 0, 25N
2. *Vertical Skewer 90 (VS90)*: Pitch 0, Roll $\frac{\pi}{2}$, 25N
3. *Tines Vertical 0 (TV0)*: Pitch -0.5, Roll 0, 20N
4. *Tines Vertical 90 (TV90)*: Pitch -0.5, Roll $\frac{\pi}{2}$, 20N
5. *Tilted Angled 0 (TA0)*: Pitch 0.4, Roll 0, 10N
6. *Tilted Angled 90 (TA90)*: Pitch 0.4, Roll $\frac{\pi}{2}$, 10N

3.3.1 Hyperparameter Tuning with a Doubly-Robust Estimator

In addition to the normal hyper-parameters associated with linear regression (context dimension d_c and L2 regularization parameter λ), each algorithm has its own exploration hyper-parameter. We tune these by constructing a simulated training environment using the same data collected for validation of SPANet. Specifically, three food items with very different success rate distributions over primitives are excluded from SPANet’s training data. Banana slices are very sensitive to fork pitch, with TA performing the best by a wide margin because it prevents the slice from slipping off the fork. Grapes are, in general, very difficult to pick up, with the best strategy still dependent on biases in perception and planning. Apple slices are, in general, very easy to acquire, with some sensitivity to fork roll angle due to their length. Based on [49], VS, perpendicular roll angle, is likely the best strategy by a slight margin.

Since this data, by necessity, was collected with off-policy bandit feedback, the counterfactual loss of un-taken actions had to be imputed to create a complete supervised learning dataset. One strategy is to fill in the expected loss based on the average success rate of the

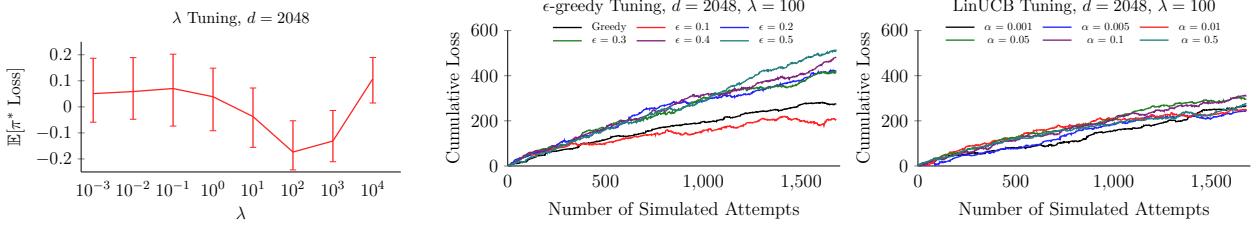


Figure 3.3: Hyper-parameter tuning in simulation, with banana, apple, and grapes excluded from SPANet. (*Left*) full feedback performance as a function of λ with 95% confidence intervals. (*Center/Right*) cumulative loss of the contextual bandit algorithms on the excluded food items as a function of their exploration hyper-parameters.

given action across all foods of the same type, and while this may be an accurate estimate, it can introduce a grouping bias into the simulation relative to the real world. We can eliminate the bias in this initial estimate using a doubly robust [42] estimator

$$\hat{l}_{DR}(x_i, a) = \hat{l}_a + (l_i - \hat{l}_a) \frac{\mathbf{1}(a_i = a)}{\mathbb{P}(a_i|x_i)}, \quad (3.4)$$

where \hat{l}_a is the imputed value from the initial grouping, $\mathbb{P}(a_i|x_i)$ is the probability that a given action was taken a_i during data collection ($\frac{1}{6}$ in this case since data was collected uniformly across all actions), and l_i is the actual binary loss associated with that sample (only available for a_i). This estimator eliminates bias (i.e., $\mathbb{E}[\hat{l}_{DR}] = l$) from our imputed values at the cost of added variance.

First, we tuned the linear regression parameters d_c and λ . This was done by treating the entire dataset as a simple supervised learning problem, directly learning an estimate $\hat{\theta}$ with full feedback. Using the original SPANet feature space with $d_c = 2048$, we found that we needed significant regularization (large λ) to see any results on our limited data set. However, while reducing our feature-space dimension d_c could in theory improve our regret bounds (e.g., LinUCB's $R_T \sim \tilde{O}(d_c)$), it empirically reduced the accuracy of $\hat{\theta}$. For λ , while 100 and 1000 produced similar full-feedback model performance (as shown in Fig. 3.3a), $\lambda = 100$ performed better when tuning exploration parameters.

Figure Fig. 3.3b,c show the results of tuning the exploration parameters ϵ (for ϵ -greedy) and α (for LinUCB). Stochastic ϵ -greedy showed a clear local minimum at $\epsilon = 0.1$. Meanwhile, LinUCB demonstrated exhibited lower sensitivity to α . We selected $\alpha = 0.01$, which reached a slight minimum loss, for the real robot experiments.

3.4 Experiment: On-Robot Bandit

3.4.1 System Description

For these experiments, our system consists of a 6 DoF JACO2 robotic arm [68]. The arm has 2 under-actuated fingers that grab a custom-built, 3D-printed fork holder. For haptic input, we instrumented the fork with a 6-axis ATI Nano25 Force-Torque sensor [120]. For visual input, we mounted a custom built wireless perception unit on the robot’s wrist; the unit includes the Intel RealSense D415 RGBD camera and the Nvidia Jetson Nano for wireless transmission. Food is placed on a plate mounted on an anti-slip mat commonly found in assisted living facilities. A complete system description (including later additions) is covered in Appendix A.

3.4.2 Procedure

For each attempt, we place a single food item in the center of the plate. ADA positions itself vertically above the plate and performs object detection and featurization using a checkpoint of SPANet that was trained with some food items excluded. Importantly, the identity of the food items, while used for object detection, was *never* made available to the contextual bandit algorithm. After performing the requested action, the binary loss is recorded manually, and the learning algorithm is updated. To mimic a realistic feeding setting, we removed and replaced the food item only after a successful acquisition.

We define a bite acquisition attempt as a success ($l_t = 0$) if the target food item, either the whole piece or a cut portion, remains on the fork for 5 seconds after removal from the plate. If the target food item is skewered with at least 2 out of 4 times but the fork fails to

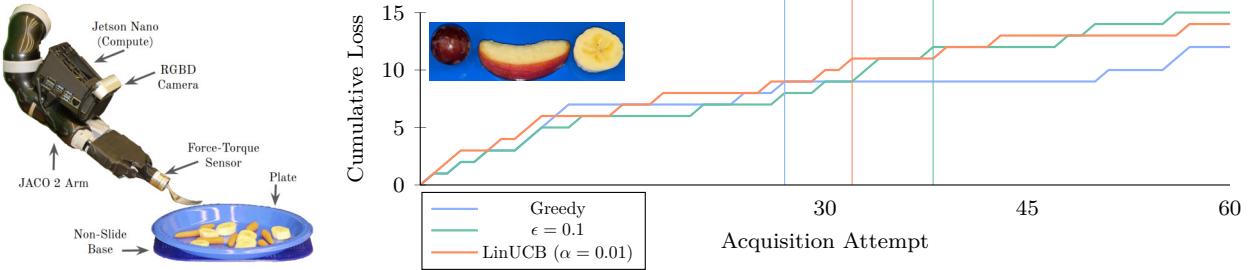


Figure 3.4: Results of Experiment 1 (*Right*) using the Autonomous Dexterous Arm (ADA) (*Left*). SPANet was trained on 12 food types – excluding apples, bananas, and grapes. Initialized to $\theta_0 = \vec{0}$, the robot cycled 20 times through all 3 food types. For each algorithm, the vertical line represents the point after which 100% of the strategies selected were among the best strategies for the food item observed. All algorithms converge within ~ 10 failures per food item.

pick it up or the food falls off soon after lift-off, the attempt is deemed a failure ($l_t = 1$). If less than 2 out of 4 times touch a food item due to system-level errors (e.g., perception or planning), we discard the attempt completely.

3.4.3 Experiment 1

This experiment tests whether the features generated by SPANet trained without previously unseen food items are rich enough for the contextual bandit algorithm to find the best strategy for multiple food items. We cycle through 3 food items (apple, banana, then grape) 20 times, leading to 60 total attempts. We choose these items for the same reason as the simulation: they are representative of the majority of our food data set.

3.4.4 Experiment 2

This experiment tests whether the contextual bandit algorithms can adapt to new food items when given a θ that has already been trained on many previously seen dissimilar food items from the doubly robust simulated environment. Unlike Experiment 1, we test on only one food item at a time, so the set of dissimilar food items is of a non-negligible size. For banana

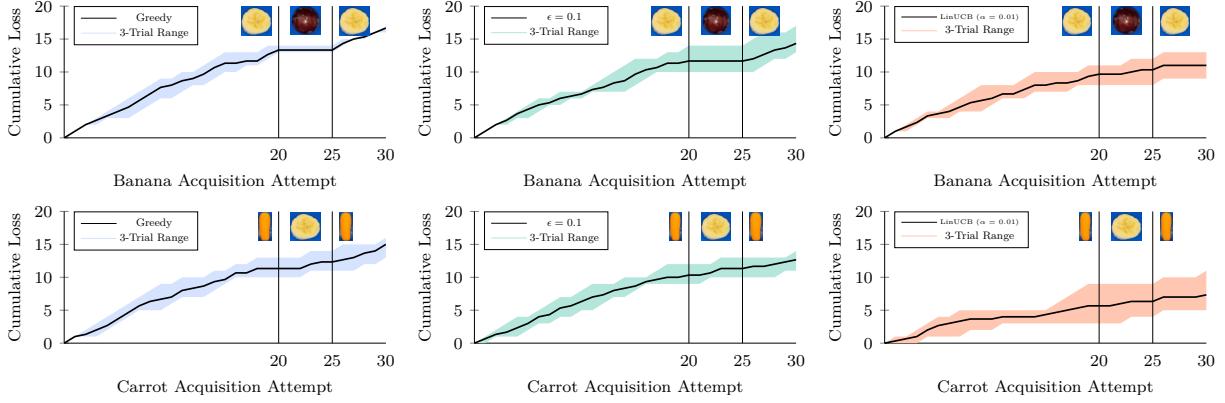


Figure 3.5: Empirical cumulative loss for each contextual bandit algorithm on an unseen food item when θ_0 was trained on all previously seen food items. For each trial, attempts 20-25 were performed on a different, previously seen food item to demonstrate that the policy did not forget its best strategy during the unseen food attempts. LinUCB demonstrates the most stable performance, especially when there are multiple good strategies (as for carrots).

slices, θ was trained on all ~ 8000 attempts on all 15 non-banana food items because it is the only food item sensitive to fork pitch, where TA is the best strategy. For carrots, which are very sensitive to fork roll (i.e., VS and TV, perpendicular roll angle, are the likely best strategies by a wide margin), θ was trained on ~ 3000 attempts, which excluded other food items sensitive to fork roll, such as apples, bell peppers, and celery. For each food item, we conducted 20 attempts, followed by 5 attempts with a previously seen food item (grape and banana, respectively), followed by another 5 attempts of the test food item, to ensure that π did not forget previously seen food items after adapting to a new one.

3.4.5 Results

Figure 3.4 (right) summarizes the results of Experiment 1. All algorithms suffered a cumulative loss between 10 and 15. The key takeaway is that all algorithms converged to the best strategy set within ~ 10 failures per new food item, after which the best strategy (or a strategy within the best set of strategies) was chosen 100% of the time for each food item.

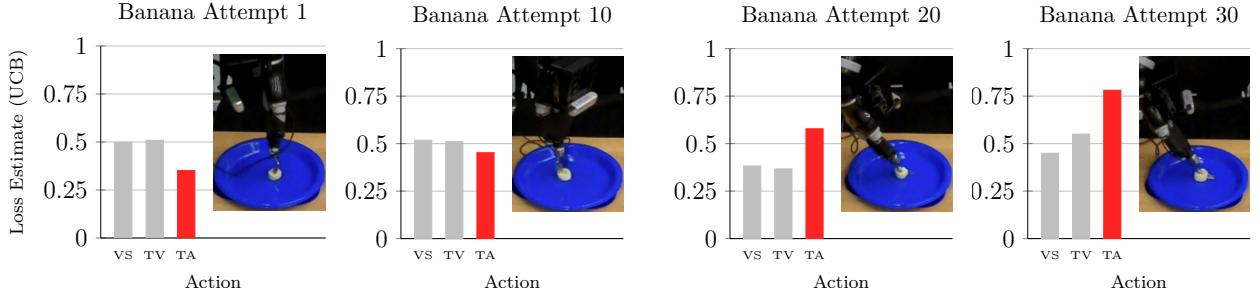


Figure 3.6: Evolution of the internal upper confidence bound (UCB) estimate of the success rate of each action over time for one of the banana experiments. Since banana slices have rotational symmetry, for each fork pitch we present only the maximum UCB of the two roll angles. The optimal fork pitch for acquiring banana, tilted-angled (TA), is highlighted in red and increases over the course of the experiment.

The only subsequent errors were due to uncertainties in perception and planning. Interestingly, greedy had the highest performance using this metric, though, unlike Experiment 2, it was not weighed down by pretraining in θ , and greedy is often empirically competitive in contextual bandit settings [20]. These results suggest that the SPANet features are indeed rich enough for contextual bandit algorithms to learn the best strategy for multiple representative food items simultaneously.

Figure 3.5 summarizes the results of Experiment 2. LinUCB exhibited superior cumulative loss performance for both food items, and greedy exhibited particularly poor performance. ϵ -greedy produced higher-variance results, spanning from the best performance of greedy to the worst performance of LinUCB. The inverse of Experiment 1, this is probably due to the weight of the pretrained θ forcing greedy to try previously good strategies before exploring new ones. LinUCB could capitalize on the uncertainty introduced by seeing a significantly different context. Figure 3.6 shows how LinUCB’s upper confidence bound estimates changed over time as it adapted to bananas. Regardless, its consistent performance on the previously seen food item did demonstrate that the contextual bandit algorithm could adapt to new information without forgetting the best strategies for previously seen food items.

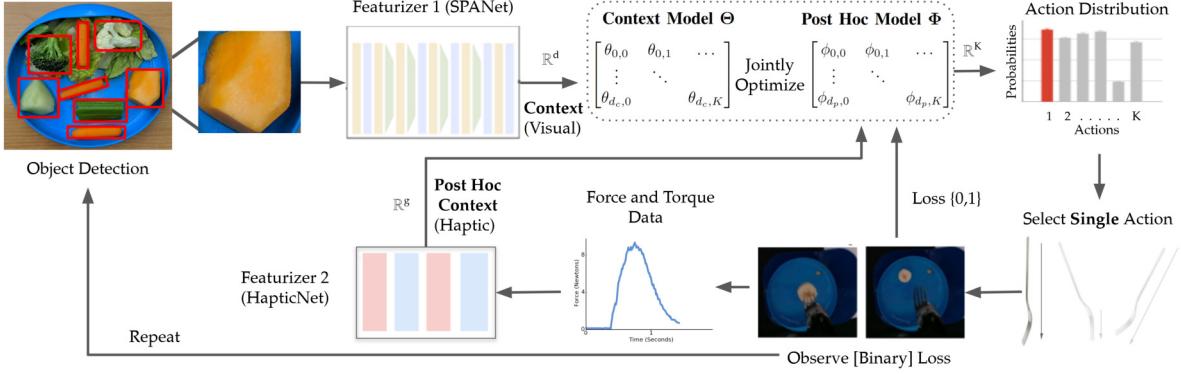


Figure 3.7: Post hoc augmented contextual bandit framework. We only observe the visual context from SPANet prior to action selection, but the post hoc context from HapticNet is used with the observed loss to update the visual model.

One key takeaway from these results is that LinUCB is empirically robust across a range of hyper-parameters and initial conditions. A fluke early failure will not sink a high-expectation action since the increasing variance dampens the decreasing expectation. Robustness is vital for a robotic feeding system: users, especially those with some mobility, may not tolerate too many errors in an autonomous system they use daily [16]. While the number of failures seen here may not be acceptable for a single meal, both experiments suggest that this is a 1-time cost that can be amortized over the life of the feeding system.

3.5 Integrating Haptic Post Hoc Context

While we successfully mapped food acquisition into the contextual bandit framework, relying on visual data alone hampered its effectiveness. Food items that look similar (but not identical), such as ripe and un-ripe banana slices, can have very different consistencies, leading to different optimal manipulation strategies. It is hard to learn this map with only visual data. Haptic feedback from physical interactions can be informative for object classification [118, 4, 17] and inferring object properties such as haptic adjectives [32], rigidity [40], elasticity [52], hardness [131], and compliance [72, 18]. Previous work has also

shown that combining visual and haptic modalities can help towards inferring global haptic mapping [122] and learning multi-modal representations [81] during manipulation. In this section, we show that we can leverage haptic feedback collected after action selection during manipulation to more quickly learn how to map visual information to the optimal strategy for a given type of food.

Specifically, we propose augmenting the conventional contextual bandit setting with an action-independent d_p -dimension post hoc context vector p_t (haptic force-torque feedback in our setting) observable after selecting the action a_t . We can justify this inclusion as follows: At each time step, assume there exists some hidden state z_t (for example, all the information describing an unripe banana slice). In our setting, the context c_t (e.g., the picture of the food) and the post hoc context p_t (e.g., haptic parameters only available after the action is taken) are just alternate representations of this underlying state (e.g., we can name the food item by either looking at it or touching it). Therefore, if there exists some function h that maps the state onto the loss vector, then there should also exist functions f and g that map the context and post hoc context respectively onto that same loss vector. In other words, we assume $\mathbb{E}[l_t] = h(z_t) = f(c_t) = g(p_t)$.

This augmentation maps neatly onto the bite acquisition setting. At each round $t = 1, \dots, T$, the interaction protocol is modified as follows:

4. *Post hoc context observation.* During action execution, time series force and torque data is passed through HapticNet (described below) to create the haptic context $p_t \in \mathbb{R}^4$.

HapticNet is a small multi-layer perceptron (MLP) from previous work [53]. The first 50ms of force and torque data after contact with the food (as determined by force thresholding) are passed through two ReLu layers. The output is the softmax-ed vector p_t classifying the food as “hard”, “medium”, “soft”, or “hard-skin”. Importantly, the hardness of the food is intrinsic, independent of the action. In previous work, the manipulation action used by human participants was directly affected by this categorization, and so we are motivated to use a linear model here as well.

The structure of this environment as it pertains to bite acquisition is shown in Fig. 3.7.

3.5.1 Learning: Joint Model Regression

Recalling our assumption $\mathbb{E}[l_t] = f(c_t) = g(p_t)$, we propose jointly estimating f and g with least squares regression under the constraint that they produce the same outputs, i.e. $f(c_t)[a] = g(p_t)[a] \forall a$. This constraint could be a soft constraint, weighting the square difference between the outputs by hyperparameters. However, since the contextual bandit setting in general does not come with a well-defined training and validation set, it is desirable to reduce the number of hyperparameters requiring tuning. Therefore, in this work, we only consider using a hard constraint. Importantly, this constraint should be valid for all actions, allowing all time steps to factor into the estimate no matter which action was taken.

To demonstrate this, we jump into the linear setting, where $f(c_t)[a] = \theta_a^\top c_t$ and $g(p_t)[a] = \phi_a^\top p_t$.

$$\hat{\theta}_a, \hat{\phi}_a := \arg \min_{\theta_a, \phi_a} \|\mathbf{C}_a \theta_a - L_a\|_2^2 + \|\mathbf{P}_a \phi_a - L_a\|_2^2 \quad (3.5)$$

$$\text{s.t. } \mathbf{C}\theta_a = \mathbf{P}\phi_a \quad (3.6)$$

As before, $\mathbf{C}_a \in \mathbb{R}^{T_a \times d_c}$, $\mathbf{P}_a \in \mathbb{R}^{T_a \times d_p}$, and $L_a \in \mathbb{R}^{T_a}$ are matrices of importance weighted contexts, post hoc contexts, and losses respectively. The constraint, being valid for all actions, uses the full context matrix $\mathbf{C} \in \mathbb{R}^{T \times d_c}$ and post hoc context data matrix $\mathbf{P} \in \mathbb{R}^{T \times d_p}$. Using the constraint to define a transformation matrix $\phi_a = (\mathbf{P}^\top \mathbf{P})^{-1} \mathbf{P}^\top \mathbf{C} \theta_a := \mathbf{H} \theta_a$, we can solve for the weight estimate.

$$\hat{\theta}_a = [\mathbf{C}_a^\top \mathbf{C}_a + \mathbf{H}^\top \mathbf{P}_a^\top \mathbf{P}_a \mathbf{H}]^{-1} [\mathbf{C}_a^\top + \mathbf{H}^\top \mathbf{P}_a^\top] L_a \quad (3.7)$$

This formulation provides a normative reason to expect empirical improvements over the context-only setting. If the post hoc context model is known perfectly, it can recommend the correct action for a given context after only a single attempt, cutting down exploration by a factor of K . More formally, if we know $\phi_a^* \forall a$, then we know what the expected loss $\mathbb{E}[L] = \mathbf{P}\phi_a^*$ would have been for action a at all time steps, *including time steps where we*

did not take action a . In other words, we can rewrite the hard constraint from Equation 3.6 as

$$\theta_a = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{P} \phi_a = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbb{E}[L] \quad (3.8)$$

This surface defined by the hard constraint is just the solution to standard linear regression on all time steps. The upshot is that knowing the post hoc context model reduces the problem from bandit feedback to full feedback regression, which is a much easier problem.

3.5.2 Exploration: Modified LinUCB

Under the linearity assumption, i.e.,

$$L_a = \mathbf{C}_a \theta_a^* + \epsilon = \mathbf{P}_a \phi_a^* + \epsilon \quad (3.9)$$

We can show that $\hat{\theta}_a$ is an unbiased estimate of θ_a^* with a covariance bounded from above (via Cauchy-Schwartz and Jensen's inequalities) by

$$\Sigma_p := 2 (\mathbf{C}_a^\top \mathbf{C}_a + \mathbf{H}^\top \mathbf{P}_a^\top \mathbf{P}_a \mathbf{H})^{-1} \quad (3.10)$$

From here, following the same logic as [146], we can construct a lower confidence bound equivalent to Equation 3.3 and replacing Σ_a with Σ_p . This work focuses on comparing this modified algorithm to the baseline version of LinUCB as described in Section 3.2.3.

3.6 Validation on Synthetic Data

Before implementing this framework on the robot, we conducted two experiments with synthetic post hoc context to validate the potential benefits from this setting. Experiment 1 is designed to demonstrate that a low-dimension post hoc context vector can lead to faster learning, even if it contains no new information. It does so by varying the size of the synthetic context vector while keeping the size of the synthetic post hoc context vector fixed. Experiment 2 is designed to see if a well trained post hoc context model can effectively reduce the contextual bandit to the easier problem of full feedback online learning. The synthetic

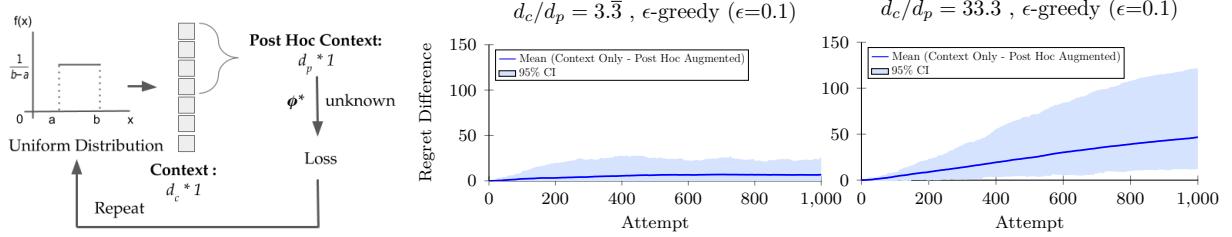


Figure 3.8: Experiment 1 shows the effect of dimensionality on the regret difference between the *Context Only* and *Post Hoc Augmented* model. The context vector is sampled from a uniform distribution and the post hoc vector is constructed using first d_p components. The full loss vector is defined as $l_t = p_t^\top \phi^*$ where ϕ^* is unknown to the model. We observe an improvement in regret difference as d_c is increased from 10 (Center) to 100 (Right).

post hoc context is constructed to be easy to learn, while the context vectors, derived from MNIST data, do not even adhere to the linear model assumption.

3.6.1 Experiment 1: Low Dimension Post Hoc Context

The setup for this experiment is outlined in Figure 3.8(Left). We first fix the number of actions $K = 10$ and the dimension of the post hoc context $d_p = 3$. We then generate a random, hidden pseudo-invertible post hoc context model $\phi^* \in \mathbb{R}^{d_p \times K}$. At each time step, we sample a context vector $c_t \sim [0, 1]^{d_c}$, which is shown to the bandit algorithm. The first d_p components are defined to be the post hoc context, and the full loss vector $l_t = p_t^\top \phi^*$ is computed accordingly. The algorithm incurs regret $l_t[a_t] - \min_a l_t[a]$ and is shown $l_t[a_t]$ and p_t .

Results For $d_c = 10$ and $d_c = 100$, we run the context-only and the post-hoc-augmented learners for 40 trials, 1000 attempts per trial, and record the cumulative regret. The results are shown in Figure 3.8(Center, Right). With the lower dimensional context, we observe that the two learners perform comparably, with a slight advantage to the post-hoc-augmented learner. However, with the higher dimensional context, the post-hoc-augmented learner

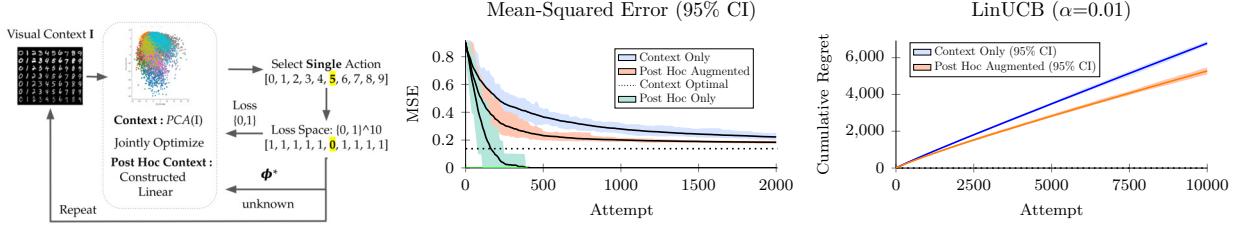


Figure 3.9: Experiment 2 is performed on the standard MNIST dataset to test learning speeds of various models. The context vector is reduced using the PCA and the post hoc context vector is constructed from the loss using ϕ^* which should be easy to learn. We see from the MSE plot (*Center*) that the *Post Hoc Augmented* model reaches it's optimal value much faster than the *Context Only* model. As soon as the post hoc model is learned the context model quickly approaches it's best possible value. Also from the Regret plots (*Right*) we can see the *Post Hoc Augmented* model achieving lower regret value than it's *Context Only* counterpart.

exhibits significantly better performance than the context-only learner.

The context and the post hoc context contain the exact same information about the loss vector, but the post hoc context does so with fewer dimensions, making each observation more informative. These results support the idea that lower dimensional post hoc context is beneficial. Even if it contains no new information, it can still be used to train an accurate context model more quickly.

3.6.2 Experiment 2: Easy-To-Learn Post Hoc Context

The setup for this experiment is outlined in Figure 3.9(Left). Context vectors are derived from the MNIST dataset [80], which consists of labeled 28×28 images of hand-written digits (0 to 9) split into a training set with 60k samples and a test set with 10k samples. At each time step, we sample an image, and then use a PCA (trained on the training set) to reduce it to a $d_c = 200$ dimension context vector. This vector is shown to the bandit algorithm, which returns an action $a_t \in [K = 10]$. For a given image, we can define the full loss vector $l_t \in \{0, 1\}^{10}$ to be 1 if the incorrect digit is guessed and 0 otherwise. As in Experiment 1, we construct a random linear post hoc context model $\phi^* \in \mathbb{R}^{d_p \times K}$ and use it to manually

construct a post hoc context vector $p_t \in \mathbb{R}^{d_p=10}$ from the full loss vector. This makes for a post hoc context model that is extremely easy to learn perfectly. The loss $l_t[a_t]$ and the post hoc context p_t is shown to the bandit algorithm.

Results First, to test learning speed, we ran a post-hoc-augmented learner and a context-only learner on the training set with random uniform exploration. Every 10 attempts, we freeze both linear context models $\hat{\theta}$ and record its mean square error (MSE) on 2k samples from the test set.

$$MSE := \frac{1}{|\text{Test Subset}|} \sum_{c_t, l_t \in \text{Test Subset}} \|\hat{\theta}^\top c_t - l_t\|_2^2 \quad (3.11)$$

For the post-hoc-augmented learner, we also recorded the MSE of the post hoc context model $\hat{\phi}$. All three are plotted in Figure 3.9(Center). Note that the best possible context model MSE (as determined by training on the full loss vectors from the entire training set) is 0.1383. We should expect no context model θ to beat that, even one augmented by post hoc context.

As expected, the noise-less post hoc model ϕ is learned perfectly as soon as it sees $d_p = 10$ linearly independent samples for each digit (which happens within ~ 500 total samples). At this point, the post-hoc-augmented learner significantly deviates from the context-only learner, and it has reached its plateau within another ~ 500 samples. Meanwhile, the context-only learner still did not reach its plateau after 2000 samples. These results support the idea that a perfect context model effectively reduces the problem from bandit feedback to full feedback, allowing for faster learning.

We then combined each learner with a LinUCB exploration strategy, ran them on the entire 10k-sample test set, and recorded the cumulative regret. These results are shown in Figure 3.9(Right), and demonstrate a significant improvement of the post-hoc-augmented bandit over the context-only bandit. This shows that an improved learning speed can translate to reduced regret, the metric that we care about in this setting.

3.7 Experiment: On-Robot Post Hoc Augmented Bandit

Our experimental setup and action space of motion primitives is identical to that of the previous experiment.

3.7.1 Offline Results and Tuning

The first step was to ensure that our post hoc context, the haptic data, was descriptive enough to potentially benefit the visual context model. To this end, we collected 115 samples of the robot skewering 3 food items, chosen to be representative of different haptic categories and optimal action (as determined in [49]): grape is classified as “hard skin” and has the optimal action TV90, strawberry is “medium” and prefers VS0 or TV0, and banana is “soft” and prefers TA0 or TA90. For each sample, we recorded the visual context c_t , post hoc context p_t , action taken a_t , loss $l_t[a_t]$, and food type name (e.g. “grape”).

Since this data, by necessity, was collected with bandit feedback, we impute the full loss vector \hat{l} using the doubly-robust estimator described in Section 3.3.1.

Similarly to Experiment 2 (Section 3.6.2), we divided this data set into 80 training examples and 35 test examples. We then ran a context-only learner and a post-hoc-augmented learner on the training set with uniform exploration, freezing the context model $\hat{\theta}_a$ after each time step to measure its MSE on the test set. The difference in MSE is shown in Figure 3.10(Left). While the size and variance of the dataset makes it hard to show significance, the mean suggests that there may be a learning benefit to the post hoc context early on that fades over time.

We also used the full 115 samples to tune the exploration hyperparameter α for both LinUCB implementations. The optimal value for both the context-only and the post-hoc-augmented learners was $\alpha = 0.01$, which we used for the online experiment.

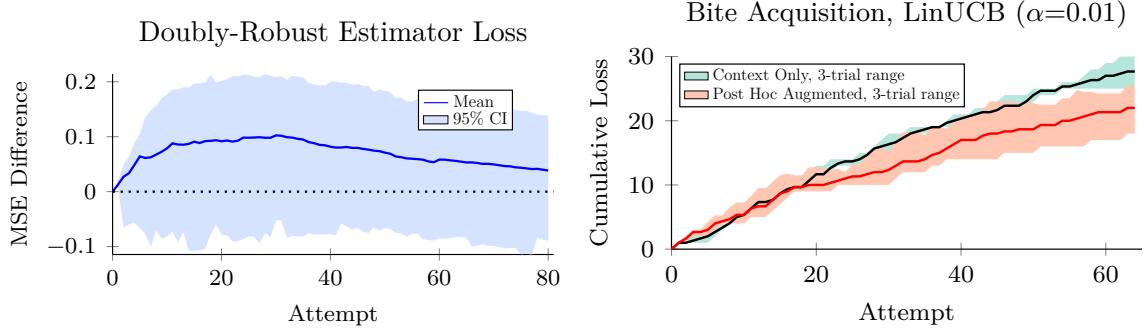


Figure 3.10: Results of the bite acquisition experiment using the Autonomous Dexterous Arm (ADA) (*Left*). The MSE Difference plot (*Center*) shows that there is an early benefit in learning with the post hoc context which reduces over time. Also from the Cumulative Loss plot (*Right*) it's evident that with *Post hoc Context Augmented* the cumulative loss incurred with increasing attempts is lower than its *Context Only* counterpart.

3.7.2 Online Experiment

For the online experiment, we ran both the post-hoc-augmented LinUCB and the context-only LinUCB algorithms on 8 previously unseen food types, 8 attempts per food type, for a total of 64 attempts per trial. The 8 food types included 2 “hard” foods (carrot and celery), 2 “medium” foods (strawberry and cantaloupe), 2 “soft” foods (banana and kiwi), and 2 “hard skin” foods (cherry tomato and grape). In the absence of noise, we would expect the robot to take about 48 attempts to figure out the optimal action for each food item: 6 actions for each of the 8 food items. Given that we expect the optimal action to be knowable from the haptic category, we would hope to decrease that convergence time by a factor of 2 with the post hoc context: 6 actions for each of the 4 haptic categories.

The results of the experiment are shown in Figure 3.10(Right). In this setting, where it is impossible to observe the full loss vector l_t , we record cumulative loss $\sum_t l_t[a_t]$ instead of cumulative regret. By this metric, we do see some improvement of the post-hoc-augmented agent over the context-only agent. Over 3 trials, the former experienced fewer failures, accumulating an average total loss of 22 compared with 27.667 for the context-only learner.

One key takeaway from this work is the success of post-hoc-augmented bandits in a

variety of empirical settings that potentially deviate significantly from the assumed linear model. This suggests that it may be fruitful to pair this augmentation with other empirically competitive exploration strategies such as Thompson Sampling. Overall, these results suggest that multi-modal feedback can be leveraged in interactive learning to allow for more data-efficient adaptive food acquisition.

Chapter 4

TRACTABLE ACTION SPACE: LEVERAGING HUMAN DATA

The previous chapter showed that food acquisition could be successfully modeled as a linear contextual bandit problem, allowing the robot agent to balance exploration and exploitation to identify an appropriate manipulation strategy for picking up previously unseen food items. However, the efficacy of this approach is predicated on having a set of possible manipulation strategies that includes at least one action that can pick up every food item. If no such action exists for a given food, a vanilla contextual bandit formulation is not going to find one spontaneously. One could consider introducing an action space so large as to encompass all possible robot motions. For example, the action space could be the set of all trajectories of a given length confined to the workspace. Of course, the issue such a space is that the learning problem becomes intractable without large quantities of data. As discussed in the introduction of Section 3, this is not tenable given the current state of food simulation.

Therefore, a key challenge for tractably adaptable food manipulation is the construction of an action space that is simultaneously (1) broad enough to cover all the food items the user might encounter and (2) small enough such that learning the optimal action for a new food items remains a possibility. Additionally, there are human interaction components of the problem to consider. As discussed in Section 2, erratic robot motion can lead to user discomfort, even if there is no danger of unsafe contact. Therefore, ideally, all the possible actions in the constructed space should avoid unpredictable motion.

Our key insights in this chapter are two-fold. First, even without a full imitation pipeline, context-free human demonstrations (i.e., only trajectory and force data, without the need for observing the food being acquired) can be used to define a distribution of reasonable manipulation actions. Secondly, if the resulting expert distribution can be represented in a

space such that, under some distance metric, similar actions perform similarly (or, in other words, the loss function $l : \mathcal{C} \times \mathcal{A} \rightarrow \mathbb{R}$ is continuous with respect to the action space), then that same distance metric can be used to identify and sample “spatially diverse” actions from within the expert distribution. The result is a small set of actions that represents the diversity of human food acquisition.

The work in this chapter was also in two publications [49, 60].

4.1 Related Work: Learning Grasps from Human Demonstration

As humans are generally expert tool users, a plethora of work has gone into transferring those skills to robots [22, 142, 39]. Some work focuses on higher-level task planning [141, 115]. Others learn more granular motions by generating dynamic motion primitives that a model learns to stitch together [92, 3]. Still others investigate a hand-design restricted action space for use with end-to-end models [31]. In contrast, this chapter looks at leveraging application-specific structure and human data to systematically restrict the action space prior to learning a model.

[125] simplifies the representation of the human action space for grasping rigid objects by performing in-the-wild data collection with a handheld gripper. Works that consider complete hand-like dexterous manipulation [154, 91] utilize pose estimation models to leverage very large datasets of non-annotated manipulation videos. Yamaguchi et al. [151] isolates the pouring application for fast policy learning. Ettehadi and Behal learn robot feeding movements from demonstration [46], but require kinesthetic demonstrations, which can be difficult for inexperienced robot users to provide.

Other works [125, 154, 91, 151, 46] utilize simulations or extensive human and environment data to learn offline RL policies that yield good generalization performance at test time. In contrast, this chapter exploits application-specific structure and the simpler contextual-bandit setting described in Section 3 to learn a simpler model that can be refined online without the need for simulation or large datasets.

4.1.1 Food Acquisition Qualitative Taxonomy

The action spaces presented in this chapter were inspired by the partial qualitative taxonomy described by Bhattacharjee et al. [17]. This taxonomy of human manipulation strategies is reproduced in Figure 4.1, and given the set of food items used, has a particular focus on skewering strategies. It identifies a few parameters of the motion that tended to differ across food items. For slippery items, a tilted approach increased the ratio of friction to gravity. For hard-to-pierce items, wiggling motions and partial contact increased the pressure on the food. Additionally, tine orientation was important for non-homogenous (e.g. hard-boiled eggs) or oblong food items (e.g. carrots) to make sure the food item was acquired in its entirety and did not roll away.

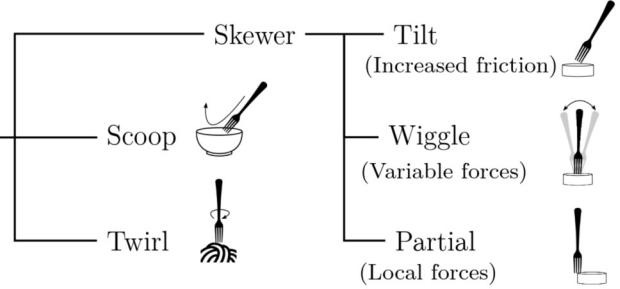


Figure 4.1: Partial qualitative taxonomy of human food acquisition from Bhattacharjee et al. [17]

4.2 SPANet: Analysis of a 2D Action Space

While the previously-described taxonomy provides a comprehensive list of human acquisition techniques, it does not provide a quantitative way to represent those motions so a robot can execute them. Our initial work narrowed in on one specific aspect: fork tilt. Variations in specifically the approach angle of the fork were sufficient to cover a variety of both hard and soft discrete food items like cut fruit and vegetables. The result is the action space presented in Figure 4.2(top), with 3 fork pitch and 2 fork rotation angles resulting in 6 total actions.

For pitch: “Vertical Skewer” (VS) has the handle of the fork orthogonal to the plane of the table and is the most consistent to execute. “Tines Vertical” (TV) has the tines of the fork vertical for higher pressure on hard-to-pierce items. “Titled Angle” (TA) has the handle of the fork at a 45° to the normal vector of the table, which is useful for softer items. For

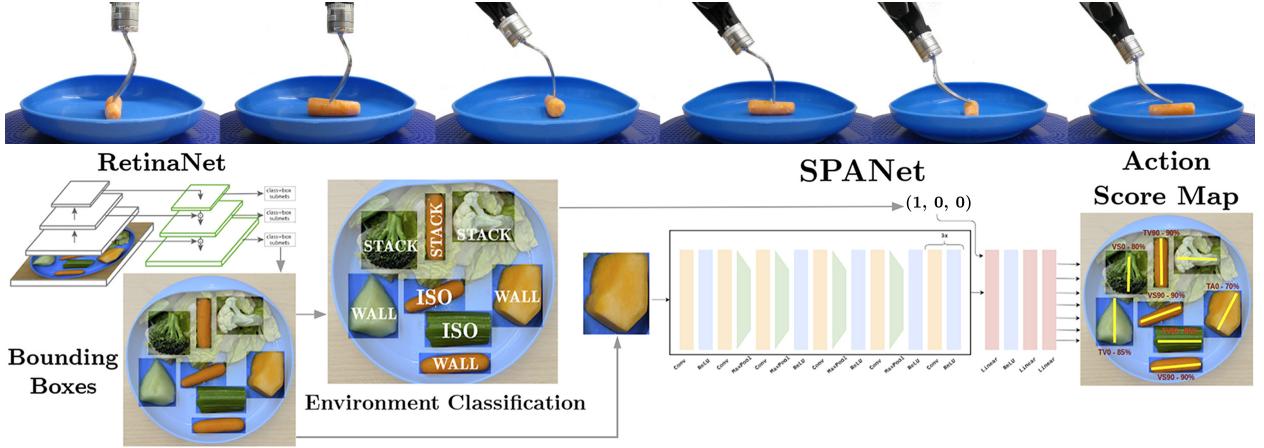


Figure 4.2: (*Top*): 6 Actions used in the SPANet classifier: 3 fork pitches by 2 fork rolls. (*Bottom*): SPANet classification framework. Using a full-plate image, RetinaNet outputs bounding boxes around food items. An environment classifier identifies items as being in one of three environments: isolated (ISO), near a wall (i.e., plate edge) or another food item (WALL), or on top of other food items (STACK). SPANet uses the bounding boxes and environment features to output the success probability for each bite acquisition action and the skewering axis for each food item.

roll: the fork could either be oriented such that the tines were parallel (“0”) or perpendicular (“90”) to the major axis of the bounding ellipse of the food item. The former was useful for items that exhibited changes along their length, like broccoli, while the latter was better for hard oblong food items, like baby carrots.

To demonstrate the efficacy of this action space, we trained a convolutional neural network classifier, the *Skewering Position Action Network* (SPANet) to predict the best action for each food item given an RGB image of that food item and a description of the food’s environment (i.e. “isolated”, “wall”, or “stacked”) determined with classical computer vision techniques. The pipeline and architecture are described in Figure 4.2(bottom), and further detail and model architecture specifics can be found in our publication [49].

The main upshot of this approach were two-fold. First, the action space exhibited good performance on a variety of discrete food items, getting close to the 80% user-defined threshold for daily use. Second, due to the small size of the action space and some of the similarity

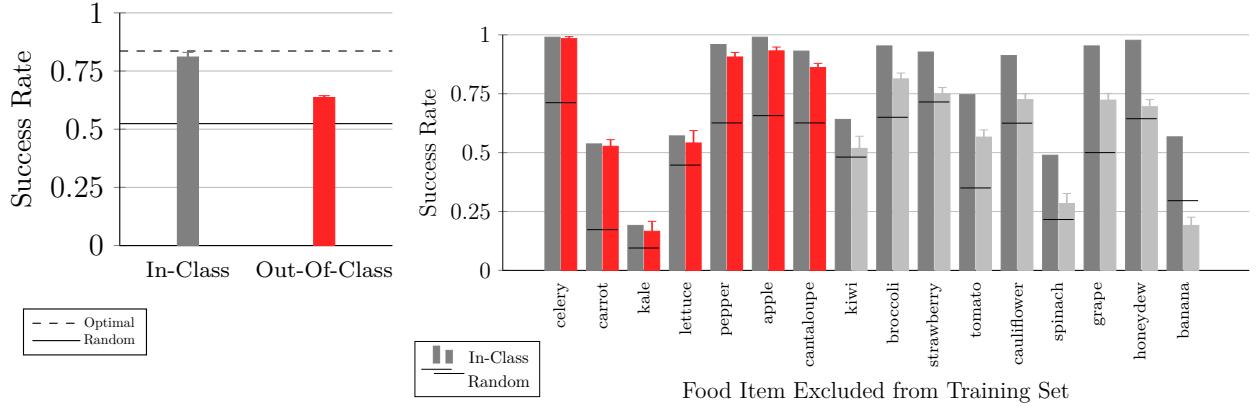


Figure 4.3: SPANet’s generalizability to unseen food items (Out-Of-Class) compared with known food items (In-Class). Each bar represents the expected success rate of SPANet’s proposal action, i.e., how well the best action proposed by SPANet would have performed according to the ground-truth data. *Left:* Overall comparison. The success rate of SPANet’s proposal action drops by 18% for unseen food items. *Right:* Per class comparison. For food classes from celery to cantaloupe, SPANet’s OOC predictions stay very close to IC predictions.

between visually-similar food items, SPANet had decent out-of-class performance as well, choosing a good action for previously-unseen food items at better-than-random rates. These results are summarized in Figure 4.3. SPANet was used as the feature extraction model and its action space used as the initial action space for our online learning feeding experiments in Section 3.

4.3 Acquisition Action Schema

SPANet focused exclusively on the “Tilt” component of the qualitative taxonomy. In order to fill out the rest quantitative, we developed an acquisition schema that describes an acquisition action space that is narrow enough to distill all of these taxonomic elements but flexible enough to capture variants of those elements (e.g., additional wiggling, or a different approach angle). The action is defined by 26 continuous parameters divided into three phases: approach, grasp, and extraction (Figure 4.4, left).

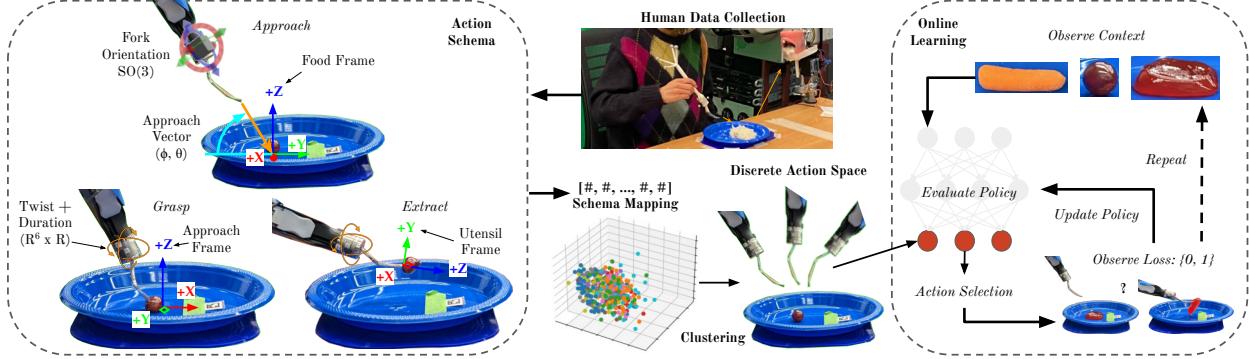


Figure 4.4: (Left): Visual description of the action schema. Robot motions are in orange. Reference frames are represented as three-color axes with X in red, Y in green, and Z in blue. (Right): General food acquisition pipeline. Human data is collected, mapped into the schema, and clustered into a discrete action space. This space is small enough to treat food acquisition as a contextual bandit to learn online the optimal action for new food items.

4.3.1 Approach (Pre-Grasp)

This phase captures the fork tilt and approach angle elements of the qualitative grasp taxonomy while discarding fork motions and parameters that are irrelevant to manipulation (e.g. distance from the food).

Frame Definitions: Define the world frame with an arbitrary origin and orientation such that -Z is the direction of gravity. We assume the existence of a food manipulation target bounded by an ellipsoid which may possibly intersect a flat plane defined by a table/plate/other surface parallel to the X-Y plane (from here just referred to as the plate). The projection of this ellipsoid onto the plate is the *bounding ellipse* of the food. The *food frame* is the default reference frame in which all parameters are defined unless otherwise specified. The origin is defined as the center of the bounding ellipse. +Z is aligned with that of the world frame and the X-axis is aligned with the major axis of the bounding ellipse.

Parameters: The approach consists of the following 9 parameters: Fork orientation ($SO(3)$), Approach polar and azimuthal angle ($[0, \frac{\pi}{2}] \times [0, 2\pi]$), Target approach point within

the food (\mathbb{R}^3), Force threshold (+ \mathbb{R}).

Implementation: During implementation, the utensil begins an arbitrary distance from the food and moves in a straight line towards the target approach point until either that point is reached or the force on the utensil exceeds the threshold. Note that the definition of the food frame introduces a π -rotation symmetry depending on which direction along the X-axis is +X. In this work, this symmetry is broken during the on-robot experiments based on which approach direction is within the robot's workspace and easiest for the on-board planning algorithm.

4.3.2 Grasp

This phase captures the wiggling, twirling, and in-food scooping motions of the qualitative taxonomy.

Frame Definitions: Define the *approach frame* as the food frame rotated by the azimuthal angle of the approach direction. Define the *utensil frame* with an origin at the very tip of the utensil (e.g., between the middle two tines on a fork) such that +Z points along the handle of the utensil and the X-axis goes across the face of the utensil. In other words, the Euler angles in this frame correspond with roll (Z), pitch (X), and yaw (Y). Using this frame instead of the food frame allows the approach and grasp to be parameterized independently. For example, approaching from the side instead of the front should not result in a grasp rotation yawing instead of pitching the fork.

Parameters: The grasp consists of the following 9 parameters: Angular velocity in utensil frame (\mathbb{R}^3), Linear velocity in approach frame (\mathbb{R}^3), Duration (+ \mathbb{R}), Force and torque thresholds (+ $\mathbb{R} \times +\mathbb{R}$).

Implementation: During implementation, the utensil will execute the provided velocities for the provided duration, or cut short if the force or torque thresholds are reached.

4.3.3 Extraction

This phase captures any stabilizing rotations that take place after the food is on the fork (e.g., rotating the fork so an acquired banana piece does not slip off).

Parameters: Similarly to grasp, the extraction consists of the following 7 parameters: Angular velocity in utensil frame ($\mathbb{R}^2 \times +\mathbb{R}$), Linear velocity in approach frame (\mathbb{R}^3), Duration ($+ \mathbb{R}$). While a force and torque threshold can be introduced, it is rendered unnecessary in this work by requiring the extraction motion to move against gravity away from the plate.

Implementation: Extraction is implemented the same way “grasp” is.

4.3.4 Euclidean Distance

A key element of this space is that it was designed such that each parameter had a single, independent effect on the final trajectory. For example, consider a basic scooping action along the $+X$ direction in the food frame. If the polar approach angle rotated by 90° , such that the food approaches from the $+Y$ direction, an in-food grasp motion defined in the food frame would continue in the $+X$ direction, abruptly changing the direction of the scoop (as well as the side of the fork leading the scooping motion). By defining the grasp motion in the approach frame, the single polar approach angle parameter rotates the entire trajectory, modifying it by only a single dimension. The upshot is that the Euclidean distance metric, which treats each dimension independently, becomes an appropriate measure of action similarity, a fact which will motivate the clustering approach described in the next section.

4.4 Capturing Human Bite Acquisition Strategies

Although the 26 dimensional action schema is large, we hypothesized that only specific points (acquisition actions) within this schema will actually be commonly used to acquire food items. To identify those points, we had able-bodied participants acquire a variety of food items and feed them to an actuated mouth. The food items were chosen based on

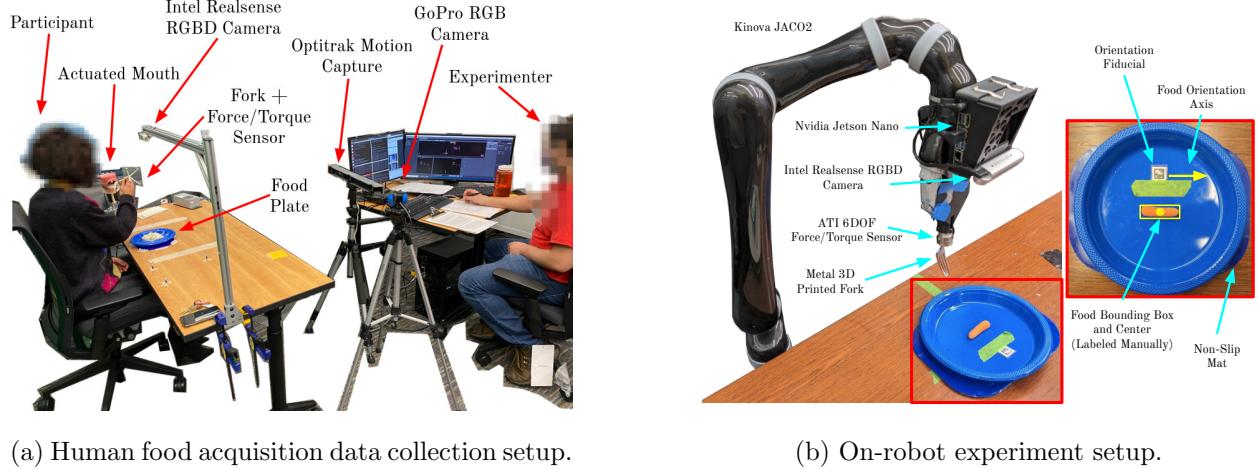


Figure 4.5: Food acquisition trials. For each trial, a single food item was acquired. Food perception (both center-of-mass and orientation) was performed with classical computer vision through a fiducial and color-based background rejection.

the meals an end-user with C1 quadriplegia¹ eats in a week, and included bread-y items like bagels and pizza, heterogeneous items like sandwiches rice and beans, gelatinous items like jello, and stringy items like noodles. As the users acquired the food, a motion capture system captured fork motion, a force-torque sensor on the fork captured haptic factors, and an RGB-D camera above the plate captured visual aspects of the food manipulation (Figure 4.5(a)). More details about the data collection can be found in the appendix of our conference publication [60].

4.4.1 Dataset

We published the data gathered here [101] to facilitate future research in food acquisition. This dataset consists of 496 bite acquisition trials, totaling over 1.25 hours of food acquisition data across 9 participants and 13 unique food items.

¹C1 quadriplegia refers to paralysis of all four limbs as a result of an injury to the first, or top-most, cervical vertebrae.

4.4.2 Human Data Analysis

Extracting an Action Schema Point

For each bite’s acquisition data, we extracted a point within the acquisition action schema that was close to that motion. We developed the procedure by iteratively extracting action schema points from bites, and then visualizing the actual participant’s motion superimposed with the extracted action on a random subset of bites to determine how to improve the extraction procedure. The result of this extraction was the creation of an expert distribution of actions within our acquisition schema action space. By sampling from this distribution, the resulting actions are more likely to have predictable motions that are less likely to make the user uncomfortable. Complete implementation details are covered in our conference publication [60] and our code repository².

Clustering Actions

Just sampling from the expert distribution directly would likely yield redundant actions. If many human examples coalesce around a single strategy, we certainly want that strategy in our final action space, but we only need one copy of it. Ideally, we would employ “spatially diverse” sampling, such that each sample is as different from the others as possible (under the Euclidean metric, as our schema was designed with that metric as an appropriate measurement of similarity) while still remaining within the expert distribution. This is exactly the effect of clustering algorithms.

We ran k-medoids on the entire dataset of 407 extracted actions, where each action has 26 dimensions. Notably, the clustering did not consider aspects of the action that would be outputted by the perception system, such as food reference frame, and only included the aspects of the acquisition motion that might generalize across food items. This resulted in $k = 11$ actions (corresponding to the within-cluster-sum-of-square-distances elbow point). A full quantitative parameterization and video of each action are provided in the supplementary

²https://github.com/personalrobotics/corl23_towards_general_food_acquisition

materials. Qualitatively, we observed emergent behavior consistent with findings in previous work [17], such as in-food wiggling, tilted extraction, and the use of vertical tines for high force.

From Human to Robot Actions

Although this procedure outputs representative actions for the motions that participants took when acquiring food items, it also learns some aspects of motion that are particular to the morphology of a human arm. For example, participants' motions tended to approach food from the right, since they were right-handed and feeding a person to their left. However, since the robot approaches food from above and feeds someone behind it (sitting in the wheelchair), it no longer needs to be constrained to right-to-left motion. As described in Section 4.3, the definition of the food frame is ambiguous with respect to a π rotation about the axis of gravity. Since the food location was fixed, we manually broke this symmetry by choosing the orientation that was easiest for the on-board planning algorithms. Further, some in-food grasp motions that humans executed, specifically tiny rotations of the fork (3° or less), produced negligible motion of the robot with significant planning and collision checking time; so we truncated those rotations to 0° .

4.5 Experiment: Action Space Evaluation

This experiment was designed to test the utility of the discrete 11-action space on a variety of food items. Previous work in assisted feeding suggests that, depending on level of mobility impairment, users can generally tolerate up to 20% failure rate in food acquisition [16]. Our hypothesis was two-fold: (1) *Coverage*: for each food item, at least one action would meet or outperform baseline performance and meet the 80% user threshold. (2) *Minimal Bad Actions*: each action would have acceptable performance on at least one type of food. If the action set lacks coverage, it is likely too small to adequately acquire new food items in the home, while if there are many bad actions, it is likely too big and will be difficult to use for online learning.

Our hardware setup is summarized in Figure 4.5(b), and details, including the hardware description, trial description, and success metric are fleshed out in Appendix A.

4.5.1 Experiment Design

We evaluated our action space on 14 diverse food items. Some food items were identical to those used during the human acquisition data collection (Section 4.4): fries, broccoli, mashed potatoes, spinach mix, and jello (cut into $\sim 1.5\text{cm}$ slices to obviate the need for cutting). Some food items had similar properties to those in the human data collection with different visual characteristics: powdered doughnut holes, white rice, white bread sandwich, and flat noodles. Finally, some food items were new: baby carrots, grapes, half-strawberries, banana slices, and kiwi slices.

The baseline action set consisted of 3 skewering techniques described in Section 3.3. *Vertical skewer* (VS) orients the handle of the fork to be orthogonal to the table and moves straight down applying up to 15N of force before moving straight back up. *Tines vertical* (TV) orients the tines of the fork to be orthogonal to the table and again applies 15N of force straight down before moving back up. Finally, *tilted angle* (TA) orients the handle of the fork 45 degrees off the table normal with the fork flat facing up and approaches the food at that same angle, moving straight upwards after skewering.

For each food item, we perform 10 trials with each of the 3 baseline and 11 human-informed actions for a total of $14 \text{ actions} \times 14 \text{ food types} \times 10 = 1960$ trials. At about 1 minute per trial, data collection took about 33 hours.

4.5.2 Results

These results are summarized in Figure 4.6. All error bars represent Wilson Binomial Proportion 95% Confidence Intervals ($n = 140$ in aggregate, $n = 10$ per food item). Overall, the best action for each food item from the human-informed set significantly outperforms both the best action from the baseline set and the user-defined benchmark with a success rate of 94.6% ($p < 0.05$ necessarily by non-overlapping confidence intervals).

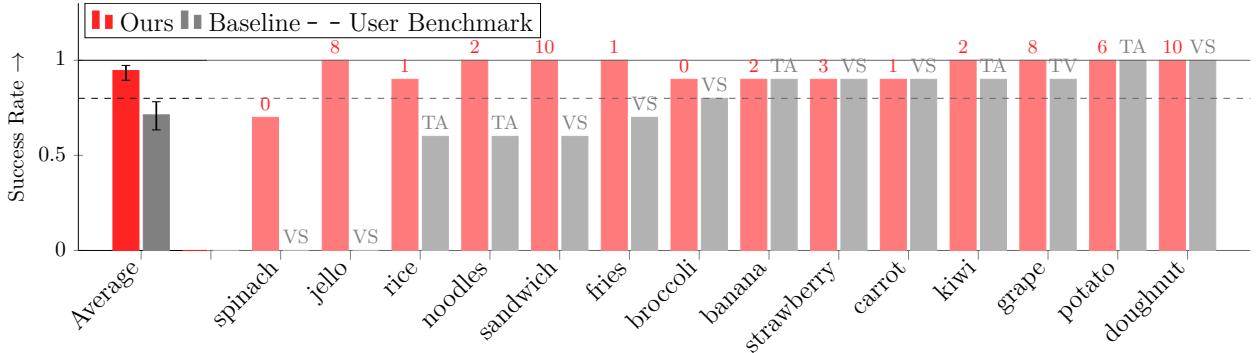


Figure 4.6: Best action for each food item from both the baseline and our action spaces. The specific action is labeled above each bar.

Coverage All food items except for spinach exhibited a success rate of 90% or higher within the new action space, exceeding the 80% user benchmark. Single-leaf spinach, difficult to acquire due to its thinness, came close with a 70% success rate with the best human-informed action. The nearly complete coverage suggests that this action space is large enough to handle the variety of food items necessary for in-home deployment. Additionally, reducing k maintained a subset of the $k = 11$ medioids down to at least $k = 5$, and so coverage is achieved for $k \geq 8$, as below that, the action space does not include the only action that covers jello.

Bad Actions Almost all human-informed actions exhibited good performance on at least one food item. Actions 0, 1, 2, 3, 6, 8, and 10 were the optimal action for spinach, carrot, banana, strawberry, potato, jello, and sandwich respectively. While not an optimal action for any food item, actions 4, 5, and 7 exhibited $\geq 70\%$ success on at least one food item. The only exception was action 9. This action captured the “cutting” motion that humans used on the full, undivided jello cups. Therefore, only the side of the fork comes into contact with the food, making success less likely. That 10/11 actions exhibited good performance suggests that this action space is not excessively large.

Baseline Comparison Carrots, grapes, strawberries, bananas, and broccoli exhibited good ($\geq 80\%$) performance with the optimal baseline actions that were designed for them in previous work [53], with the human-informed actions performing as well or slightly better. Sandwich, fries, noodles, and rice exhibited 60 – 70% baseline performance, with insufficient in-food contact (e.g. not enough of the fork present inside of every sandwich layer) as the primary failure mode. These failures were remedied by the increased in-food grasp motion of the human-informed actions. Finally, jello and spinach were completely impossible for the baseline actions to acquire. Jello needed a significant rotation during extraction to prevent the heavy chunk from slipping off the fork. Spinach needed a significant lateral force during the grasp phase to wedge the fork between the flat leaf and the plate. Finally, the human-informed actions were generally able to acquire a greater mass of rice (258mg vs. 212mg) and potato (890mg vs 5930mg) than TA, the only baseline action with any form of scooping-like motion.

4.6 *Experiment: Online Action Selection*

As described in Section 3, we assert that online learning is a necessary component of any in-home robot food acquisition system to handle previously-unseen food items. This is supported by the extensive in-lab evaluation time required for the previous experiment. In this experiment, we evaluate the ability of an off-the-shelf online learning procedure to identify sufficiently good human-informed actions for acquiring previously unseen food items. Our hypothesis is that, given that there are only 11 actions (and as few as 4 can cover the space), such a system should be able to reach the user benchmark on the order of 11 trials for each new type of food. At about 1 minute per trial, in an assistive feeding context, this would happen well within the bounds of a 20-30min meal.

4.6.1 *Learning System*

As in Experiment 1, this experiment constituted a series of trials. We cycled through all 14 food items, with 1 trial per item. A group of 14 trials, one per food item, constitutes a

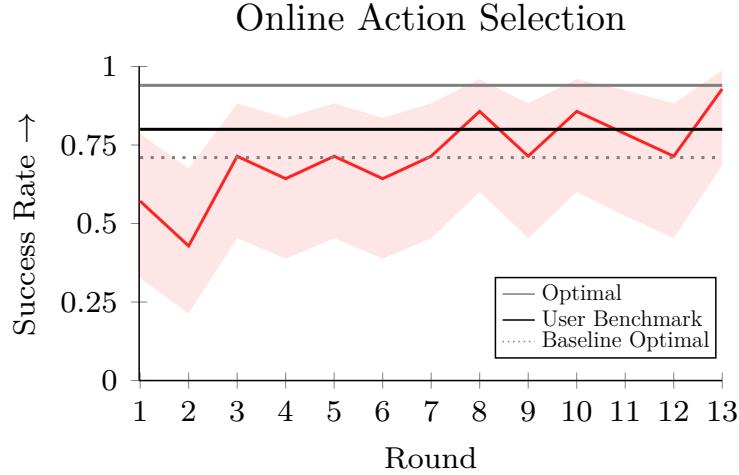


Figure 4.7: Acquisition success rate for each round of 14 trials (1 per food item) using LinUCB. Error bars represent the 95% confidence interval.

round. Data collection ceased once the success rate over the course of a round exceeded 90% (approaching the 94% optimal). For each trial, food orientation, manual perception, action execution, and success definition, are identical to what is described in Section 4.5.1.

Our online learning procedure models food acquisition as a contextual bandit [20, 58] with visual context and augmented with haptic post hoc context based on related work [59]. Our exploration scheme is LinUCB [146]. A full description of our post hoc augmented contextual bandit approach can be found in Section 3.7.

4.6.2 Results

These results are summarized in Figure 4.7, where we plot the success rate across all 14 food items in each round (with Wilson 95% Confidence Interval, $n = 14$). Since most human-informed actions perform well on most food items, we see that we get a 50% success rate even in the first round. By round 8, the action performance is on par with the user benchmark. And by round 13, we have approached the expected optimal performance with this action space (i.e. the Average shown in Figure 4.6). At one minute per trial, this suggests that this

system can successfully learn an acceptable acquisition action for 2-4 new types food within the span of at 30min meal, assuming that all foods of a given type have a similar success rate for each action. And that number is likely higher for foods with similar haptic properties. Most food items converged to a sufficiently good action (e.g. Action 1) with 5 rounds. The food items that took the longest to learn were the haptically “unusual” food items like jello, rice, noodles, and mashed potato, which exhibited particularly poor performance on the skewering actions that worked well on the firmer food items.

Chapter 5

CONCLUSION

This dissertation aims to develop a food manipulation system that enables an in-home robot-assisted feeding system to adapt to any food item a user might want to eat without the need for excessive in-home data collection or erratic, unpredictable exploration motions. Our approach makes quite a few assumptions based on the structure of the problem that feedback from users to enable us to reach this north star. In general, our design philosophy has been to start with the simplest approach that works, and only increase data or algorithmic complexity as needed. As a result, while we do believe the key insights present in our work can apply more generally, the actual algorithms and empirical studies are fairly specific to the assisted-feeding setting. This final chapter reviews the design choices and assumptions we made and proposes some future work in both the robot-assisted feeding space and the more general food manipulation space that could be done to apply this work more generally.

5.1 Future Work

5.1.1 Dynamic Action Space

In Chapter 4, we presented a methodology to use human trajectory data to identify a subset of food acquisition actions that can acquire a wide variety of food items for robot-assisted feeding applications. The 11 actions we distill from our publicly available dataset [101] are sufficient to pick up the 14 food types including hard carrots, soft bananas, slippery jello, noodles, compound sandwiches, and continuous mashed potatoes. And the set is so small that we can reasonably expect to determine the optimal action for 2-4 food items over the course of a 30-minute meal.

The foods selected for both the human data collection and on-robot experiments were

motivated by surveying a participant and co-designer with mobility impairments about their eating habits. We believe they cover a wide variety of rheological contexts, but even with this variety, there may still exist food types that are sufficiently distinct as to not be covered by the actions presented here. Future work in the home can help identify such foods. These can possibly be addressed with online action space expansion. For example, the caregiver can provide some kinesthetic demonstrations that can be mapped into the action schema. If these new actions are sufficiently far from existing actions, they can be added to the dataset. If they are close to other actions, they can identify a discontinuity in the schema space that can either be more densely sampled, or used as motivation for a modification of the schema space to mitigate or remove they discontinuity.

5.1.2 Extending the Bandit Setting

In Chapter 3, we mapped the food acquisition problem into a contextual bandit framework, augmented with post hoc haptic context. This approach has a lot of benefits: the contextual bandit framework is well-studied, data efficient, and has been shown to be very effective in fairly general domains. But in the domain of robotics, there are also some limitations inherent to the vanilla setting, more saliently, there is no feedback provided to the bandit agent until the end of the entire trajectory. The trade-off between data efficiency and action-extensiveness manifests in part from the fact that the contextual bandit framework is in a sense situated on a sort of spectrum between supervised learning and full reinforcement learning. On one hand, the contextual bandit *is* supervised learning between the context vector \vec{c} and the loss vector \vec{l} , but instead of the full loss vector being revealed, only a single component (i.e. the index of the selected action) is revealed. On the other hand, the contextual bandit *is* reinforcement learning in the specific case where each episode is of length $T = 1$.

Future work can look into using ideas from across this spectrum to improve bandit performance in the food acquisition setting. On the side of supervised learning, one idea is to leverage the slate bandit [37], where the discrete action index is replaced with a set of

discrete indices called a *slate*. The loss vector becomes effectively modeled by a loss matrix or other combination of contributions from each element of the slate. In the context of food acquisition, the discrete set of robot motions can instead become a discrete set of approach, grasp, and extraction that can be mixed-and-matched and stitched together. This could enable a exploration of a more extensive action space. Other methods in this vein could include: (1) adding an action to ask the caregiver or the user for help, which maps to the concept of bandits with budgeted information retrieval [149, 55]; (2) if the user does not care what food they want, have the robot select a food item based on the context that will be likely to yield the most information, as measured by some statistical optimality criterion, as studied in the field of optimal experimental design [124].

On the side of reinforcement learning, the $T = 1$ assumption could be slightly relaxed to increase the extensiveness of actions or effectively enable closed-loop feedback to the agent while maintaining a tractable data requirement. As an example of the former, consider “premanipulation” actions, like pushing the food around the plate. While this costs time, it could lead to a state that is easier for acquisition, such as pushing food towards the edge of the plate or grouping smaller food items together. With this framework, we can consider a reduced RL setting, where at each time step in a given episode the agent can choose from either a library of premainpulation actions that prolong the episode with a time cost or an acquisition action that ends the episode with the final binary cost. There are at least two approaches for solving this setting. A *model-based* approach looks into estimating how a given premanipulation action will modify the state of the plate. This can potentially be done with a simulator or an optical-flow-based computer vision algorithm. With a model, action selection can be solved tractably as a tree search. See recent work that uses a similar approach for noodle acquisition [129]. Alternatively, a *model free* approach could bypass this at the cost of more data collection by exploring the set of premanipulation actions, potentially online, and propagating the acquisition cost backwards with standard value iteration. This setting demonstrates how future work can move in the direction of reinforcement learning without losing data tractability.

5.1.3 Integrating Food Transfer and other Feeding Tasks

The scope of this dissertation is limited to the problem of food acquisition, but a full robot-assisted feeding system needs to also comfortably transfer the food from the utensil to the user’s mouth. Previous work has identified ways in which the acquisition and transfer problems are coupled [54]. For example, it may be better to acquire a long food item like a carrot or french fry at the edge instead of the center so that it is easier for the user to remove it from the fork. These heuristics can be modeled into the loss function used in the acquisition phase.

Looking at bite transfer in isolation, the motion that the robot takes to get to the user’s mouth is itself a trade-off between keeping the food balanced on the utensil, orienting the food so it is easy to grab, user preferences for how they want to interact with the robot, and the environment the robot is in (e.g. avoiding blocking the user’s line of sight). Some previous work looks into heuristics that can be used when planning the transfer trajectory [13], but an online learning system could allow such optimization to adapt to different environments and changes in user preferences.

In general, there are so many features that go into a robot-assisted feeding system with varying levels of technical complexity and user-expressed importance. We direct interested readers to Nanavati, et al. [100], Figure 7, for a partial list. General food acquisition is only one part of this system, and we hope that the work here can be used as a building block for more comprehensive systems going forward.

5.2 Closing Thoughts

“Autonomous systems exist because of the people that build them. As roboticists, we are responsible for ensuring that those systems serve real societal needs” [66].

Robotics as a field can have a fascination with full generality. Many works push towards the goal of manipulating arbitrary objects to arbitrary poses, or navigating around arbitrary environments to solve arbitrary tasks. This is undoubtedly interesting work that represents

the future of the field. However, such a broad view comes with the risk of overlooking more specific but no less important problems that robots should be able to solve *today*.

This dissertation focuses on one such problem: robot-assisted feeding. We use the assumptions, user preferences, and other structures specific to this setting to narrow the general food manipulation problem, making it tractable with current state-of-the-art algorithms and techniques. The result is a system with a more limited scope, but a hopefully large impact on the millions of people around the world who need help feeding themselves. We hope that other roboticists are encouraged to tackle similarly specific but important problems to realize all the positive potential robotics can have on the human condition.

BIBLIOGRAPHY

- [1] Yasin Abbasi-yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In J Shawe-Taylor, R S Zemel, P L Bartlett, F Pereira, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2312–2320. Curran Associates, Inc., 2011.
- [2] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. *arXiv preprint arXiv:1402.0555*, February 2014.
- [3] Baris Akgun, Maya Cakmak, Karl Jiang, and Andrea L Thomaz. Keyframe-based learning from demonstration: Method and evaluation. *International Journal of Social Robotics*, 4:343–355, 2012.
- [4] Peter K Allen and Kenneth S Roberts. Haptic object recognition using a multi-fingered dexterous hand. In *IEEE International Conference on Robotics and Automation*, pages 342–347, 1989.
- [5] Redwan M Alqasemi, Edward J McCaffrey, Kevin D Edwards, and Rajiv V Dubey. Wheelchair-mounted robotic arms: Analysis, evaluation and development. In *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pages 1164–1169. IEEE, 2005.
- [6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.
- [7] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The non-stochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.
- [8] Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53. ACM, 2004.
- [9] Hamsa Bastani, Mohsen Bayati, and Khashayar Khosravi. Mostly exploration-free algorithms for contextual bandits. *arXiv preprint arXiv:1704.09011*, 2017.

- [10] Jenay M Beer, Arthur D Fisk, and Wendy A Rogers. Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of human-robot interaction*, 3(2):74–99, 2014.
- [11] Beeson automaddak feeder, 2019. <https://abledata.acl.gov/product/beeson-automaddak-feeder-model-h74501>, [Online; Retrieved on 18th April, 2019].
- [12] Michael Beetz, Ulrich Klank, Ingo Kresse, Alexis Maldonado, Lorenz Mösenlechner, Dejan Pangercic, Thomas Rühr, and Moritz Tenorth. Robotic roommates making pancakes. In *IEEE-RAS International Conference on Humanoid Robots*, pages 529–536. IEEE, 2011.
- [13] Suneel Belkhale, Ethan K. Gordon, Yuxiao Chen, Siddhartha Srinivasa, Tapomayukh Bhattacharjee, and Dorsa Sadigh. Balancing efficiency and comfort in robot-assisted bite transfer. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4757–4763, 2022.
- [14] Bestic, 2019. <https://www.camanio.com/us/products/bestic/>, [Online; Retrieved on 18th April, 2019].
- [15] Tapomayukh Bhattacharjee, Maria E Cabrera, Anat Caspi, Cakmak Maya, and Siddhartha S Srinivasa. A community-centered design framework for robot-assisted feeding systems. In *International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 2019.
- [16] Tapomayukh Bhattacharjee, Ethan K. Gordon, Rosario Scalise, Maria E. Cabrera, Anat Caspi, Maya Cakmak, and Siddhartha S. Srinivasa. Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding. In *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 181–190, 2020.
- [17] Tapomayukh Bhattacharjee, Gilwoo Lee, Hanjun Song, and Siddhartha S Srinivasa. Towards robotic feeding: Role of haptics in fork-based food manipulation. *IEEE Robotics and Automation Letters*, 2019.
- [18] Tapomayukh Bhattacharjee, James M Rehg, and Charles C Kemp. Inferring object properties with a tactile-sensing array given varying joint stiffness and velocity. *International Journal of Humanoid Robotics*, pages 1–32, 2017.
- [19] Zeungnam Bien, Myung-Jin Chung, Pyung-Hun Chang, Dong-Soo Kwon, Dae-Jin Kim, Jeong-Su Han, Jae-Hean Kim, Do-Hyung Kim, Hyung-Soon Park, Sang-Hoon

- Kang, et al. Integration of a rehabilitation robotic system (kares ii) with human-friendly man-machine interaction units. *Autonomous robots*, 16(2):165–191, 2004.
- [20] Alberto Bietti, Alekh Agarwal, and John Langford. A contextual bandit bake-off. *arXiv preprint 1802.04064*, February 2018.
 - [21] C Blanes, M Mellado, C Ortiz, and A Valera. Technologies for robot grippers in pick and place operations for fresh fruits and vegetables. *Spanish Journal of Agricultural Research*, 9(4):1130–1141, 2011.
 - [22] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on robotics*, 30(2):289–309, 2013.
 - [23] M. Bollini, J. Barry, and D. Rus. Bakebot: Baking cookies with the pr2. In *The PR2 workshop: results, challenges and lessons learned in advancing robots with a common platform, IROS*, 2011.
 - [24] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.
 - [25] Matthew W Brault. Americans with disabilities: 2010. *Current population reports*, 7:70–131, 2012.
 - [26] PN Brett, AP Shacklock, and K Khodabendehloo. Research towards generalised robotic systems for handling non-rigid products. In *International Conference on Advanced Robotics*, pages 1530–1533. IEEE, 1991.
 - [27] Steven W Brose, Douglas J Weber, Ben A Salatin, Garret G Grindle, Hongwu Wang, Juan J Vazquez, and Rory A Cooper. The role of assistive robotics in the lives of persons with disability. *American Journal of Physical Medicine & Rehabilitation*, 89(6):509–521, 2010.
 - [28] Tadhg Brosnan and Da-Wen Sun. Inspection and grading of agricultural and food products by computer vision systems—a review. *Computers and Electronics in Agriculture*, 36(2):193–213, 2002.
 - [29] Gerard Canal, Guillem Alenyà, and Carme Torras. Personalization framework for adaptive robotic feeding assistance. In *International Conference on Social Robotics*, pages 22–31. Springer, 2016.

- [30] Alexandre Candeias, Travers Rhodes, Manuel Marques, Manuela Veloso, et al. Vision augmented robot feeding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.
- [31] Lawrence Yunliang Chen, Huang Huang, Ellen Novoseller, Daniel Seita, Jeffrey Ichnowski, Michael Laskey, Richard Cheng, Thomas Kollar, and Ken Goldberg. Efficiently learning single-arm fling motions to smooth garments, 2022.
- [32] Vivian Chu, Ian McMahon, Lorenzo Riano, Craig G McDonald, Qin He, Jorge Martinez Perez-Tejada, Michael Arrigo, Trevor Darrell, and Katherine J Kuchenbecker. Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems*, 63:279–292, 2015.
- [33] PY Chua, T Ilschner, and DG Caldwell. Robotic manipulation of food products—a review. *Industrial Robot: An International Journal*, 30(4):345–354, 2003.
- [34] SOFA Cutting and Mesh Refinement, 2023. <https://www.sofa-framework.org/applications/plugins/cutting-mesh-refinement/>.
- [35] Intel RealSense D415, 2023. <https://www.intelrealsense.com/depth-camera-d415/>.
- [36] Fred D Davis, Richard P Bagozzi, and Paul R Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8):982–1003, 1989.
- [37] Maria Dimakopoulou, Nikos Vlassis, and Tony Jebara. Marginal posterior sampling for slate bandits. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 2223–2229, 2019.
- [38] Kexiang Ding and Sundaram Gunasekaran. Shape feature extraction and classification of food material using computer vision. *Transactions of the ASAE*, 37(5):1537–1545, 1994.
- [39] Praveen Kumar Donepudi. Reinforcement learning for robotic grasping and manipulation: a review. *Asia Pacific Journal of Energy and Environment*, 7(2):69–78, 2020.
- [40] Alin Drimus, Gert Kootstra, Arne Bilberg, and Danica Kragic. Classification of rigid and deformable objects using a novel tactile sensor. In *International Conference on Advanced Robotics*, pages 427–434, 2011.

- [41] Cheng-Jin Du and Da-Wen Sun. Learning techniques used in computer vision for food quality evaluation: a review. *Journal of food engineering*, 72(1):39–55, 2006.
- [42] Miroslav Dudik, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, March 2011.
- [43] Håkan Eftring and Kerstin Boschian. Technical results from manus user trials. In *Proc. ICORR*, volume 99, pages 136–141, 1999.
- [44] Gen Endo and Nobuhiro Otomo. Development of a food handling gripper considering an appetizing presentation. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4901–4906. IEEE, 2016.
- [45] F Erzincanli and JM Sharp. Meeting the need for robotic handling of food products. *Food Control*, 8(4):185–190, 1997.
- [46] Nabil Ettehadi and Aman Behal. Implementation of feeding task via learning from demonstration. In *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pages 274–277. IEEE, 2018.
- [47] Nabil Ettehadi and Aman Behal. A learning from demonstration framework for implementation of a feeding task. *Encyclopedia with Semantic Computing and Robotic Intelligence*, 2(01):1850001, 2018.
- [48] M Evans. Magpie: It’s development and evaluation. Technical report, Technical report, 1991.
- [49] Ryan Feng, Youngsun Kim, Gilwoo Lee, Ethan K. Gordon, Matt Schmittle, Shivaum Kumar, Tapomayukh Bhattacharjee, and Siddhartha S. Srinivasa. Robot-assisted feeding: Generalizing skewering strategies across food items on a plate. In Tamim Asfour, Eiichi Yoshida, Jaeheung Park, Henrik Christensen, and Oussama Khatib, editors, *Robotics Research*, pages 427–442, Cham, 2019. Springer International Publishing.
- [50] Micro Egg Switch for a Wheelchair, 2023. <https://www.quickie-wheelchairs.com/Wheelchair-Parts-Accessories/Assorted-Wheelchair-Parts/Power-Wheelchair-Joystick-Accessories/Micro-Egg-Switch-Wheelchair-Switch/40247p>.
- [51] Dylan J Foster, Alekh Agarwal, Miroslav Dudík, Haipeng Luo, and Robert E Schapire. Practical contextual bandits with regression oracles. *arXiv preprint arXiv:1803.01088*, March 2018.

- [52] Barbara Frank, Rüdiger Schmedding, Cyrill Stachniss, Matthias Teschner, and Wolfram Burgard. Learning the elasticity parameters of deformable objects with a manipulation robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1877–1883, 2010.
- [53] D. Gallenberger, T. Bhattacharjee, Y. Kim, and S.S. Srinivasa. Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding. *ACM/IEEE International Conference on Human-Robot Interaction*, 2019.
- [54] Daniel Gallenberger, Tapomayukh Bhattacharjee, Youngsun Kim, and Siddhartha S Srinivasa. Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2019.
- [55] Kyra Gan, Esmaeil Keyvanshokooh, Xueqing Liu, and Susan Murphy. Contextual bandits with budgeted information reveal. *arXiv preprint arXiv:2305.18511*, May 2023.
- [56] Mevlana C Gemici and Ashutosh Saxena. Learning haptic representation for manipulating deformable food objects. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 638–645. IEEE, 2014.
- [57] T M Gill, C S Williams, and M E Tinetti. Assessing risk for the onset of functional dependence among older adults: The role of physical performance. *Journal of the American Geriatrics Society*, 43(6):603–609, 1995.
- [58] Ethan K. Gordon, Xiang Meng, Tapomayukh Bhattacharjee, Matt Barnes, and Siddhartha S. Srinivasa. Adaptive robot-assisted feeding: An online learning framework for acquiring previously unseen food items. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9659–9666, 2020.
- [59] Ethan K. Gordon, Sumegh Roychowdhury, Tapomayukh Bhattacharjee, Kevin Jamieson, and Siddhartha S. Srinivasa. Leveraging post hoc context for faster learning in bandit settings with applications in robot-assisted feeding. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10528–10535, 2021.
- [60] Ethan Kroll Gordon, Amal Nanavati, Ramya Challa, Bernie Hao Zhu, Taylor Annette Kessler Faulkner, and Siddhartha Srinivasa. Towards general single-utensil food acquisition with human-informed actions. In *7th Annual Conference on Robot Learning (CoRL)*, 2023.
- [61] Eric Heiden, Miles Macklin, Yashraj Narang, Dieter Fox, Animesh Garg, and Fabio Ramos. DiSE Ct: A Differentiable Simulator for Parameter Inference and Control in Robotic Cutting, March 2022. *arXiv:2203.10263 [cs]*.

- [62] Laura V Herlant. *Algorithms, Implementation, and Studies on Eating with a Shared Control Robot Arm*. PhD thesis, The Robotics Institute, Carnegie Mellon University, 2016.
- [63] Hiroki Higa, Kei Kurisu, and Hideyuki Uehara. A vision-based assistive robotic arm for people with severe disabilities. *Transactions on Machine Learning and Artificial Intelligence*, 2(4):12–23, 2014.
- [64] M Hillman and A Gammie. The bath institute of medical engineering assistive robot. In *Proc. ICORR*, volume 94, pages 211–212, 1994.
- [65] Irit Hochberg, Guy Feraru, Mark Kozdoba, Shie Mannor, Moshe Tennenholz, and Elad Yom-Tov. Encouraging physical activity in patients with diabetes through automatic personalized feedback via reinforcement learning improves glycemic control. *Diabetes care*, 39(4):e59–e60, 2016.
- [66] Brian Hou. *Robot Motion Planning with Uncertainty and Urgency*. PhD thesis, University of Washington, 2023.
- [67] ATI-IA Wireless F/T Interface, 2023. https://www.ati-ia.com/products/ft/ft_WirelessFT.aspx.
- [68] Kinova JACO, 2018. <https://www.kinovarobotics.com/en/products/robotic-armseries>.
- [69] Jamie, 2019. <https://www.focalmeditech.nl/en/content/jamie>, [Online; Retrieved on 18th April, 2019].
- [70] Alberto Jardón, Concepción A Monje, and Carlos Balaguer. Functional evaluation of asibot: A new approach on portable robotic system for disabled people. *Applied Bionics and Biomechanics*, 9(1):85–97, 2012.
- [71] S. Javdani, H. Admoni, S. Pellegrinelli, S.S. Srinivasa, and J.A. Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *International Journal of Robotics Research*, 2018.
- [72] Mohsen Kaboli, Philipp Mittendorfer, Vincent Hügel, and Gordon Cheng. Humanoids learn object properties from robust tactile feature descriptors via multi-modal artificial skin. In *IEEE-RAS International Conference on Humanoid Robots*, pages 187–192, 2014.

- [73] Dae-Jin Kim, Rebekah Hazlett-Knudsen, Heather Culver-Godfrey, Greta Rucks, Tara Cunningham, David Portee, John Bricout, Zhao Wang, and Aman Behal. How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(1):2–14, 2011.
- [74] Jetson Nano Developer Kit, 2023. <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>.
- [75] Predrag Klasnja, Eric B Hekler, Saul Shiffman, Audrey Boruvka, Daniel Almirall, Ambuj Tewari, and Susan A Murphy. Microrandomized trials: An experimental design for developing just-in-time adaptive interventions. *Health Psychology*, 34(S):1220, 2015.
- [76] Michael C Koval, Jennifer E King, Nancy S Pollard, and Siddhartha S Srinivasa. Robust trajectory selection for rearrangement planning as a multi-armed bandit problem. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2685. IEEE, 2015.
- [77] Vijay Kumar, Tariq Rahman, and Venkat Krovi. Assistive devices for people with motor disabilities. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1997.
- [78] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.
- [79] Tor Lattimore and Csaba Szepesvari. *Bandit Algorithms*. Cambridge University Press, 2019.
- [80] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2, 2010.
- [81] 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.
- [82] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A Contextual-Bandit approach to personalized news article recommendation. *arXiv preprint arXiv:1003.0146*, February 2010.
- [83] Liftware handles, 2019. <https://www.liftware.com/>, [Online; Retrieved on 18th April, 2019].

- [84] I. F. Lin and H. S. Wu. Does informal care attenuate the cycle of adl/iadl disability and depressive symptoms in late life. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 66(5):585–594, 2011.
- [85] T Lin, P Goyal, R Girshick, K He, and P Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, October 2017.
- [86] P Lopes, R Lavoie, R Faldu, N Aquino, J Barron, M Kante, and B Magfory. Icraft-eye-controlled robotic feeding arm technology. *Tech. Rep.*, 2012.
- [87] Wen-Tao Ma, Wei-Xin Yan, Zhuang Fu, and Yan-Zheng Zhao. A chinese cooking robot for elderly and disabled people. *Robotica*, 29(6):843–852, 2011.
- [88] Veronique Maheu, Philippe S Archambault, Julie Frappier, and François Routhier. Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities. In *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.
- [89] Alessandra Malito. Grocery stores carry 40,000 more items than they did in the 1990s, 2017.
- [90] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1077–1084. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [91] Priyanka Mandikal and Kristen Grauman. [DexVIP: Learning Dexterous Grasping with Human Hand Pose Priors from Video](#). In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 651–661. PMLR, 08–11 Nov 2022.
- [92] Ren Mao, Yezhou Yang, Cornelia Fermüller, Yiannis Aloimonos, and John S Baras. [Learning hand movements from markerless demonstrations for humanoid tasks](#). In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 938–943. IEEE, 2014.
- [93] Dale McConachie and Dmitry Berenson. Bandit-based model selection for deformable object manipulation. *arXiv preprint arXiv:1703.10254*, 2017.
- [94] Meal buddy, 2018. <https://www.performancehealth.com/meal-buddy-system>, [Online; Retrieved on 25th January, 2018].

- [95] Meal-mate, 2018. <https://www.made2aid.co.uk/productprofile?productId=8&company=RBF%20Healthcare&product=Meal-Mate>,[Online; Retrieved on 25th January, 2018].
- [96] The mealtime partner dining system, 2018. <http://mealtimepartners.com/>,[Online; Retrieved on 15th February, 2018].
- [97] Juan Pablo Mendoza, Reid Simmons, and Manuela Veloso. Online learning of robot soccer free kick plans using a bandit approach. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*, 2016.
- [98] R Morales, FJ Badesa, N Garcia-Aracil, JM Sabater, and L Zollo. Soft robotic manipulation of onions and artichokes in the food industry. *Advances in Mechanical Engineering*, 6:345291, 2014.
- [99] My spoon, 2018. <https://www.secom.co.jp/english/myspoon/food.html>,[Online; Retrieved on 25th January, 2018].
- [100] Amal Nanavati, Patrícia Alves-Oliveira, Tyler Schrenk, Ethan K. Gordon, Maya Cakmak, and Siddhartha S. Srinivasa. Design principles for robot-assisted feeding in social contexts. In *2023 18th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2023.
- [101] Amal Nanavati, Ramya Challa, Ethan K. Gordon, and Siddhartha S. Srinivasa. A Dataset of Food Manipulation Strategies for Diverse Foods, 2022.
- [102] Isira Naotunna, Chamika Janith Perera, Chameera Sandaruwan, RARC Gopura, and Thilina Dulantha Lalitharatne. Meal assistance robots: A review on current status, challenges and future directions. In *2015 IEEE/SICE International Symposium on System Integration (SII)*, pages 211–216. IEEE, 2015.
- [103] Neater eater robot, 2019. <http://www.neater.co.uk/neater-eater-2-2/>,[Online; Retrieved on 18th April, 2019].
- [104] Nelson, 2019. <https://www.focalmeditech.nl/en/content/nelson>,[Online; Retrieved on 18th April, 2019].
- [105] Obi, 2018. <https://meetobi.com/>,[Online; Retrieved on 25th January, 2018].
- [106] Eiichi Ohara, Ken’ichi Yano, Satoshi Horihata, Takaaki Aoki, and Yutaka Nishimoto. Tremor suppression control of meal-assist robot with adaptive filter. In *2009 IEEE International Conference on Rehabilitation Robotics*, pages 498–503. IEEE, 2009.

- [107] Yutaro Ohshima, Yuichi Kobayashi, Toru Kaneko, Atsushi Yamashita, and Hajime Asama. Meal support system with spoon using laser range finder and manipulator. In *2013 IEEE Workshop on Robot Vision (WORV)*, pages 82–87. IEEE, 2013.
- [108] Oreo separator machines. <https://vimeo.com/63347829>, [Online; Retrieved on 1st February, 2018].
- [109] Alex Owen-Hill. How Robots With Vacuum Grippers Improve Food Safety, 2019.
- [110] Daehyung Park, Yuuna Hoshi, Harshal P Mahajan, Ho Keun Kim, Zackory Erickson, Wendy A Rogers, and Charles C Kemp. *Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned*. *Robotics and Autonomous Systems*, 124:103344, 2020.
- [111] Daehyung Park, You Keun Kim, Zackory M. Erickson, and Charles C. Kemp. Towards assistive feeding with a general-purpose mobile manipulator. *CoRR*, abs/1605.07996, 2016.
- [112] Anupam Pathak, John A Redmond, Michael Allen, and Kelvin L Chou. A noninvasive handheld assistive device to accommodate essential tremor: a pilot study. *Movement Disorders*, 29(6):838–842, 2014.
- [113] Chamika Janith Perera, Thilina Dulantha Lalitharatne, and Kazuo Kiguchi. Eeg-controlled meal assistance robot with camera-based automatic mouth position tracking and mouth open detection. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1760–1765. IEEE, 2017.
- [114] Chamika Janith Perera, Isira Naotunna, Chameera Sadaruwan, Ranathunga Arachchilage Ruwan Chandra Gopura, and Thilina Dulantha Lalitharatne. Ssvep based bmi for a meal assistance robot. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 002295–002300. IEEE, 2016.
- [115] Claudia Pérez-D'Arpino and Julie A Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4058–4065. IEEE, 2017.
- [116] Travers Rhodes and Manuela Veloso. Robot-driven trajectory improvement for feeding tasks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2991–2996. IEEE, 2018.

- [117] Amrita Sawhney, Steven Lee, Kevin Zhang, Manuela Veloso, and Oliver Kroemer. [Playing with food: Learning food item representations through interactive exploration](#). In *Experimental Robotics: The 17th International Symposium*, pages 309–322. Springer, 2021.
- [118] Alexander Schneider, Jürgen Sturm, Cyrill Stachniss, Marco Reisert, Hans Burkhardt, and Wolfram Burgard. Object identification with tactile sensors using bag-of-features. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 243–248, 2009.
- [119] Sebastian Schröer, Ingo Killmann, Barbara Frank, Martin Völker, Lukas Fiederer, Tonio Ball, and Wolfram Burgard. An autonomous robotic assistant for drinking. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6482–6487. IEEE, 2015.
- [120] ATI-IA Force-Torque Sensor, 2018. https://www.ati-ia.com/products/ft/ft_models.aspx?id=Nano25.
- [121] Weiwei Shen, Jun Wang, Yu-Gang Jiang, and Hongyuan Zha. Portfolio choices with orthogonal bandit learning. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [122] Ashwin A Shenoi, Tapomayukh Bhattacharjee, and Charles C Kemp. A crf that combines touch and vision for haptic mapping. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2255–2262. IEEE, 2016.
- [123] Susan M Shortreed, Eric Laber, Daniel J Lizotte, T Scott Stroup, Joelle Pineau, and Susan A Murphy. Informing sequential clinical decision-making through reinforcement learning: an empirical study. *Machine learning*, 84(1-2):109–136, 2011.
- [124] Byran Smucker, Martin Krzywinski, and Naomi Altman. Optimal experimental design. *Nat. Methods*, 15(8):559–560, 2018.
- [125] Shuran Song, Andy Zeng, Johnny Lee, and Thomas Funkhouser. [Grasping in the Wild: Learning 6DoF Closed-Loop Grasping From Low-Cost Demonstrations](#). *IEEE Robotics and Automation Letters*, 5(3):4978–4985, 2020.
- [126] Won-Kyung Song and Jongbae Kim. Novel assistive robot for self-feeding. In *Robotic Systems-Applications, Control and Programming*. IntechOpen, 2012.

- [127] Yuta Sugiura, Daisuke Sakamoto, Anusha Withana, Masahiko Inami, and Takeo Igarashi. Cooking with robots: designing a household system working in open environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2427–2430. ACM, 2010.
- [128] Priya Sundaresan, Suneel Belkhale, and Dorsa Sadigh. [Learning Visuo-Haptic Skewering Strategies for Robot-Assisted Feeding](#), November 2022. arXiv:2211.14648 [cs].
- [129] Priya Sundaresan, Jiajun Wu, and Dorsa Sadigh. [Learning Sequential Acquisition Policies for Robot-Assisted Feeding](#), October 2023. arXiv:2309.05197 [cs.RO].
- [130] Yoshihiko Takahashi and Shinichiro Suzukawa. Easy human interface for severely handicapped persons and application for eating assist robot. In *2006 IEEE International Conference on Mechatronics*, pages 225–229. IEEE, 2006.
- [131] Shinya Takamuku, Gabriel Gomez, Koh Hosoda, and Rolf Pfeifer. Haptic discrimination of material properties by a robotic hand. In *IEEE 6th International Conference on Development and Learning (ICDL)*, pages 1–6, 2007.
- [132] Kanya Tanaka, Shenglin Mu, and Shota Nakashima. Meal-assistance robot using ultrasonic motor with eye interface. *International Journal of Automation Technology*, 8(2):186–192, 2014.
- [133] Liang Tang, Romer Rosales, Ajit Singh, and Deepak Agarwal. Automatic ad format selection via contextual bandits. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1587–1594. ACM, 2013.
- [134] Tecla, 2019. <https://gettecla.com/>, [Online; Retrieved on 30th September, 2019].
- [135] Noriyuki Tejima. Evaluation of rehabilitation robots for eating. In *Robot and Human Communication, 1996., 5th IEEE International Workshop on*, pages 118–120. IEEE, 1996.
- [136] Michael J Topping and Jane K Smith. The development of handy 1. a robotic system to assist the severely disabled. *Technology and Disability*, 10(2):95–105, 1999.
- [137] Mike Topping, Helmut Heck, Gunnar Bolmsjo, and David Weightman. The development of rail (robotic aid to independent living). In *Proceedings of the third TIDE Congress*, 1998.

- [138] Loek A Van der Heide, Bob Van Ninhuijs, Arjen Bergsma, Gert Jan Gelderblom, Dick J Van Der Pijl, and Luc P De Witte. An overview and categorization of dynamic arm supports for people with decreased arm function. *Prosthetics and orthotics international*, 38(4):287–302, 2014.
- [139] Claire Vernade, Alexandra Carpentier, Tor Lattimore, Giovanni Zappella, Beyza Ermis, and Michael Brueckner. Linear bandits with stochastic delayed feedback. *arXiv preprint arXiv:1807.02089*, July 2018.
- [140] Maria Vila Abad, Gerard Canal Camprodón, and Guillem Alenyà Ribas. Towards safety in physically assistive robots: eating assistance. In *Proceedings of the 2018 IROS Workshop on Robots for Assisted Living*, pages 1–4, 2018.
- [141] Naoki Wake, Riku Arakawa, Iori Yanokura, Takuya Kiyokawa, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. [A Learning-from-Observation Framework: One-Shot Robot Teaching for Grasp-Manipulation-Release Household Operations](#). In *2021 IEEE/SICE International Symposium on System Integration (SII)*, pages 461–466, 2021.
- [142] Chao Wang, Xuehe Zhang, Xizhe Zang, Yubin Liu, Guanwen Ding, Wenxin Yin, and Jie Zhao. Feature sensing and robotic grasping of objects with uncertain information: A review. *Sensors*, 20(13):3707, 2020.
- [143] Xinxi Wang, Yi Wang, David Hsu, and Ye Wang. Exploration in interactive personalized music recommendation: a reinforcement learning approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 11(1):7, 2014.
- [144] Zhongkui Wang, Ryo Kanegae, and Shinichi Hirai. [Circular shell gripper for handling food products](#). *Soft robotics*, 8(5):542–554, 2021.
- [145] Zhongkui Wang, Yuuki Torigoe, and Shinichi Hirai. [A Prestressed Soft Gripper: Design, Modeling, Fabrication, and Tests for Food Handling](#). *IEEE Robotics and Automation Letters*, 2(4):1909–1916, October 2017. Conference Name: IEEE Robotics and Automation Letters.
- [146] Chu Wei, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.
- [147] Tomos G Williams, Jem J Rowland, and Mark H Lee. Teaching from examples in assembly and manipulation of snack food ingredients by robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 2300–2305. IEEE, 2001.

- [148] Winsford feeder, 2018. <https://www.youtube.com/watch?v=KZRFj1UZ1-c>, [Online; Retrieved on 15th February, 2018].
- [149] Huasen Wu, Rayadurgam Srikant, Xin Liu, and Chong Jiang. Algorithms with logarithmic or sublinear regret for constrained contextual bandits. *Advances in Neural Information Processing Systems*, 28, 2015.
- [150] Akihiko Yamaguchi and Christopher G Atkeson. Combining finger vision and optical tactile sensing: Reducing and handling errors while cutting vegetables. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 1045–1051. IEEE, 2016.
- [151] Akihiko Yamaguchi, Christopher G Atkeson, Scott Niekum, and Tsukasa Ogasawara. Learning pouring skills from demonstration and practice. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 908–915. IEEE, 2014.
- [152] Akira Yamazaki and Ryosuke Masuda. Autonomous foods handling by chopsticks for meal assistant robot. In *ROBOTIK 2012; 7th German Conference on Robotics*, pages 1–6. VDE, 2012.
- [153] Akira Yamazaki and Ryosuke Masuda. Various foods handling movement of chopstick-equipped meal assistant robot and there evaluation. In *International Conference on Social Robotics*, pages 158–167. Springer, 2012.
- [154] Yezhou Yang, Yi Li, Cornelia Fermuller, and Yiannis Aloimonos. Robot learning manipulation action plans by “watching” unconstrained videos from the world wide web. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29(1), 2015.

Appendix A

PORTABLE DEPLOYMENT-READY SYSTEM

The primary application of the work presented in this dissertation is a robot-assisted feeding system capable of extended in-home use. To that end, the algorithmic design and users studies in the main body of manuscript have always been accompanied by significant engineering and infrastructure work. This appendix documents that work so that other researchers can utilize our design principles, hardware, and software for their own systems, with the hope that collectively we will get closer to achieving the goal of long-term independent feeding for those with mobility impairments.

Specifically, the following three sections will detail:

- Design principles that guided our system development
- An overview of our hardware, both purchased and customized
- An overview of our software, including details of both our ROS2-based architecture and our finite-state-machine and behavior-tree-based logic design

At the time of publication, more details can be found on our project website¹, and all of the code have written is publicly available on Github². Individual relevant repositories will be linked as footnotes in subsequent sections.

¹<https://robotfeeding.io>

²https://github.com/personalrobotics/ada_feeding

A.1 Design Principles

A.1.1 Safety

The most important imperative of the system is that it should not cause any physical harm to the user. This is particularly salient for feeding given the intimacy of the task and the potential danger of the tools involved. A fork is sharp! From a software perspective, hard safety constraints are represented by quantitative system invariants, which need to be maintained for the system to be allowed to function. The following 2 sections will highlight these invariants as they come up.

With regards to verification, on one hand, with a distributed system, each component needs to be able to independently verify each invariant to ensure compliance. On the other hand, from a code reliability perspective, distributing the invariant checking comes with the need to independently verify every component of the system, which introduces a larger surface area of code that is prone to errors. How we manage this trade-off with our “watchdog” system is described in more detail in Section A.3.

A.1.2 User Control

A major goal for the robot-assisted feeding system is to engender a sense of independence for users who may feel self-conscious about being fed by a caregiver. This motivates us to ensure that the user always feels like they have control over the system. This can be partially achieved through customization, for example: robot speed, resting positions, interface mode. As discussed in Chapter 2, preferences can vary greatly between users, and this can be addressed in part by adding plenty of options. Additionally, our system is constructed such that the user interface is the seat of the logic, driving all of the other systems. The user should be able to pause the robot at any time, not just as a safety feature, but also for comfort and convenience. In general, the user should have full control over their path through the feeding state machine.

A.1.3 Portability

Ideally, the feeding system should seamlessly fit into users' eating routines and environments. A user should be able to deactivate the system and put down the feeding utensil when eating is done so that they can use their robot arm for other tasks. If the system is mounted on the user's wheelchair, the user should be able to take it anywhere their chair can go without the need to hook up to wall power or a stationary desktop. From a software perspective, the system should be able to function at least minimally without the need for reliable internet connectivity or external computation.

A.2 Hardware Design

The base of the system is a Kinova Gen2 (JACO) with a 2-finger gripper and 6 degrees of freedom that can be powered directly from the user's wheelchair battery [68]. The arm has been modified by drilling a hole just above the last joint and running a 5V DC barrel connector out from the internal power bus. The connection allows for continuous rotation of the wrist joint and powers an Nvidia Jetson Nano developer kit mounted on the wrist [74], which in turn manages an Intel Realsense D415 [35] RGBD camera over USB-C. For both user safety and system portability, no wire besides the robot-internal power bus runs from this eye-in-hand system to the rest of the wheelchair or any other system component. This also ensures that there is nothing limiting the robot's full range of motion. The enclosure for the Jetson Nano, the mount and stabilizer for the Realsense camera, and the interface with the wrist of the robot are all 3d printed ABS.

When the feeding mode is active, the robot gripper is wrapped around a customized fork assembly. The tines of the fork are 3d-printed stainless steel and bolted onto a 6-DOF ATI Nano25 force-torque transducer [120]. The sensor is powered and read by ATI's battery-powered Wireless F/T interface [67], which can optionally be charged over USB by the Jetson Nano. When not charging, there are no wires running from the fork assembly to any component of the robot. This allows the robot to let go when feeding is completed, allowing it

to partake in other tasks. Since the fork tines represent one of the most potentially dangerous components for the user, many of the safety invariants revolve around the connectivity of the Wireless F/T whenever the robot is gripping the assembly. The fork handle and physical interface with the Wireless F/T are both 3d printed ABS.

Finally, the primary computation device is “Lovelace,” a Lenovo Legion 5 laptop (with an Nvidia RTX 3060 6GB GPU) mounted on the back of the user’s wheelchair, and the primary networking component is a Cradlepoint IBR900 Ruggedized router. Both systems can be connected directly to the 24V DC power provided on most wheelchairs, with the laptop specifically drawing 65W over USB-C. This is sufficient for charging the laptop battery between feeding sessions and for the GPU to be running with minor memory throttling for at least 3 hours during a feeding session. Both the computer and the router are mounted on our custom setup with acrylic plates, but can also be placed in a mesh backpack with proper air circulation and hung from any wheelchair for maximum flexibility. The router provides a wireless access point for both the Jetson Nano and the Wireless F/T and connects over Ethernet to Lovelace; it can optionally also connect to any upstream wireless internet access point to allow for updates and time synchronization. The laptop in turn connects to the robot over USB and a standard accessibility button (e.g. a “Micro Egg Switch” [50]) over a 3.5mm aux cable. This button acts as a user-accessible emergency stop button, and a key system safety invariant is ensuring its connectivity.

A.3 Software Design

Our software stack is built on ROS2³ and the ros2-control framework. At the lowest level, we have configured hardware driver plugins that can swap between the real robot, an IsaacSim⁴ simulation environment, and a “mock” system that just echos joint commands as joint states. These plugins allow the entire system to operate both in sim and real without any other modifications.

³See https://github.com/personalrobotics/ada_ros2

⁴See <https://developer.nvidia.com/isaac-sim>

ros2-control Above the hardware drivers is a collection of ROS2 controllers that handle trajectory execution and higher-frequency PID control. They expose ROS2 actions, parameters, and topics to higher level code. Our custom controllers⁵ incorporate “force gating”, where execution will be aborted if the force or torque captured by the ATI Nano25 exceeds a configurable threshold in any direction.

MoveIt2 The glue that holds our higher level code together is MoveIt2⁶. It interfaces with the Open Motion Planning Library (OMPL) for planning calls, maintains a “world” of objects for collision checking, enables plugins for re-timing, forward and inverse kinematics calls, and exposes a suite of ROS2 services and action servers to control it all. We maintain a fork of pymoveit2⁷ to interface with these services and action servers. The MoveIt2 Servo node also allows for obstacle-away Cartesian motion.

Behavior Tree Framework At the highest level, individual robot commands are logically organized into behavior trees using `py_trees`⁸. Some trees handle multi-part complex actions, such as AcquireFood, which computes the food and approach reference frames, executes an element of our action schema (see Chapter 4), and returns the robot to the resting position. Other trees are simpler, such as MoveAbovePlate, which executes a single MoveIt2 planning and execution call.

Watchdog As discussed in the previous section, utilizing invariant verification for our safety system exposes a trade-off between coverage, i.e. making sure that every component of the system is performing this verification, and complexity. We partially address this issue with a “watchdog” system. All of our invariant checking takes place within a single ROS2 node. This includes:

⁵https://github.com/personalrobotics/pr_ros_controllers

⁶<https://moveit.picknik.ai/main/index.html>

⁷<https://github.com/personalrobotics/pymoveit2>

⁸<https://py-trees.readthedocs.io/en/devel/>

- Verifying that the Emergency Stop button is connected and has not been pressed
- Verifying that the F/T sensor is connected and returning sane readings

If this verification succeeds, the watchdog node will publish an all-clear message to the rest of the ROS2 system. All of the system invariants are now defined in one place and easier to check manually. This also simplifies the safety logic of the rest of the system: if an all-clear message has not been received in a certain period of time, shut down.

App Design Our primary user interface is a React App that can be accessed via the browser of any computer, phone, or tablet⁹. It has been designed to maximize user control. Many people with mobility impairments already utilize a variety of interfaces to interact with smart devices, such as a sip-n-puff, chin joystick, voice interface, or head-activated buttons. By operating as a web app on these devices, the feeding app can integrate with all of these systems without requiring additional configuration. Additionally, unlike previous iterations of the feeding system, the app represents the “seat” of the logic for our system. The finite state machine driving the app utilizes ROS2 action servers, services, and topics to initiate and monitor and preempt robot actions, read from the sensors, write system parameters, and generally dictate the logical flow of the system. The only elements that by-pass the app control are the emergency stop and other safety components. This enables the user to have full control of their system at all times.

⁹https://github.com/personalrobotics/feeding_web_interface