Towards Learning the Foundations of Manipulation Actions from Unguided Exploration

by

Jonathan Emerson Juett

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Electrical Engineering)
in the University of Michigan
2021

Doctoral Committee:

Professor Benjamin Kuipers, Chair
Associate Professor Dmitry Berenson
Professor Odest Chadwicke Jenkins
Professor John Laird

Jonathan Emerson Juett

jonjuett@umich.edu

ORCID iD: 0000-0002-8985-8673

Acknowledgments

This work has taken place in the Intelligent Robotics Lab in the Computer Science and Engineering Division of the University of Michigan. Research of the Intelligent Robotics lab is supported in part by a grant from the National Science Foundation (IIS-1421168).

I thank my advisor, Ben Kuipers, for his guidance in this work and for introducing me to his insightful way of thinking about Artificial Intelligence and fundamental questions about the world. I also thank Chad Jenkins, John Laird, and Dmitry Berenson for their service on my committee and helpful feedback after the proposal.

I also gratefully acknowledge the ideas, assistance, and fellowship provided by everyone in the lab. Thanks to Zhen Zeng for the initial installation of the robot's simulated Palmar reflex and guidance on beginning work with the robot.

This work was made possible with support from my fiancée, Alyssa Hanson, as well as from my family and friends. Thank you for your encouragement in this and all things.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Human infants are not born with the ability to reach and grasp. But after months of typical development, infants are capable of reaching and grasping reliably. During this time, the infant receives minimal guidance and learns primarily by observing its autonomous experience with its developing senses. How is it possible for this learning phenomenon to occur, especially when this experience begins with seemingly random motions?

We present a computational model that allows an embodied robotic agent to learn these foundational actions in a manner consistent with infant learning. By examining the model and the resulting behaviors, we can identify knowledge sufficient to perform these actions, and how this knowledge may be represented.

Our agent uses a graph representation for peripersonal space, the space surrounding the agent and in reach of its manipulators. The agent constructs the Peripersonal Space (PPS) Graph by performing random motions. These motions are performed with the table but no other nonself foreground objects present to facilitate simple image segmentation of an unoccluded view of the hand. For each pose visited, a node stores the joint angles that produced it and an image of the arm in this configuration. Edges connect each pair of nodes that have a feasible motion between them. Later in the learning process, the agent may use learned criteria to temporarily remove a node or edge from consideration if motion to it or along it is expected to cause a collision given the current position of a foreground object being treated as an obstacle. The PPS Graph provides a mapping between configuration space and image space, and the agent learns to use it as a powerful tool for planning manipulation actions.

Initially, the only known actions are moves to selected PPS Graph nodes. The agent begins learning by executing move actions, and will continue to learn by applying the same learning method with other actions once it has defined them. The agent selects a random node as target, and observes the typical results of moving to it. The action is performed in the presence of at least one nonself foreground object, and the results of each action trial can be described in terms of the object's qualitative state and any changes it undergoes. Clustering of the results of all trials may identify the large cluster of typical results and perhaps some small clusters corresponding to autonomously observed unusual events. If there is at least one such cluster, the agent defines a new action with the goal of achieving the same type of unusual result. Once a new action is defined, the

agent learns features that help achieve the goal more reliably. This learning phase is the focus of this work. This phase resembles early action learning in human infant development, where a relatively small set of examples provides the necessary experience to learn to make the action reliable, though perhaps awkward and jerky in execution. These results prepare for a second learning phase to be carried out in future work, which corresponds with late action learning in humans, where actions become efficient with smooth trajectories. At the conclusion of this work, the move, reach, ungrasp, and place actions are fully reliable, and the grasp and pick-and-place actions are semi-reliable.

CHAPTER 1

Introduction

1.1 Overview

We investigate the question of how an autonomous agent can learn to represent and act in its spatial environment. We design a graph-based representation for peripersonal space, the space surrounding the agent and within its reach. Our agent constructs an instance of this representation with observations of its motor babbling motions. We present a learning method inspired by human infant development that allows the agent to identify the typical and unusual results of its actions, and to define new actions to repeat the unusual results reliably. We implement this method for use by an embodied robotic agent, and we evaluate the reliability of the learned actions with the physical robot. Our computational model also produces behaviors consistent with those observed in infants, and provides insights into their learning and a framework for testing hypotheses.

The remainder of this chapter provides additional information on the scope and contributions of this work. One of our goals is to contribute to the discussion of the questions ”Where does knowledge of space come from?” and ”How can an autonomous agent learn to move and act in its environment?” These questions are prompted by learning evident in human infants, which is discussed in Section 1.2. Our focus is on the learning of manipulation actions carried out in peripersonal space, specifically reaching, grasping, and placing. Infants learn these actions in the early and late phases discussed in Section 1.2.1. The presence of these separate stages suggests that there are also two separate learning processes, an early process that produces reliable actions and a late process that fine-tunes the actions to be executed more efficiently. Our agent performs the early learning process in this work, and learns with a much lower sample complexity than typical reinforcement learning or deep reinforcement learning methods, and without encoding the goals of the actions to learn in a reward function. The late learning phase is left as future work, and could be carried out with reinforcement learning much more efficiently given the information from the early phase.

While our learning method may be applied to define an action for any unusual event the agent has observed and make that action reliable, we assume for this work that the agent defines the reach, grasp, and place actions after observing their respective events. This constraint allows an efficient evaluation of the agent's success rate for these actions during and after learning. Section 1.3 elaborates on the selection of these actions, which were chosen for their importance for an independent agent and the clear demonstration they provide of our learning method. Our agent relies on the observation of unusual events and intrinsic motivation to learn to repeat them reliably (in particular, an intrinsic reward based on the rate of improvement of action reliability), and this learning method is discussed further in Section 1.4. Our theoretical model allows the agent to learn in a general environment with fully unguided exploration, but at this time we have implemented a model with some additional assumptions about the environment and guidance for the agent so that the learning could take place and be evaluated within the scope of this work. In future work, the model can be improved toward the ideal proposed version, primarily by removing assumptions that simplified visual tasks and object properties, by relaxing the schedule of learning phases to focus on actions or features, and by giving the agent a stronger feature generation capability. The current progress towards fully unguided learning is described in Section 1.5. This chapter concludes with a summary of the major contributions of this work in Section 1.6 and an outline of the remainder of the dissertation in Section 1.7.

1.2 The Learning Phenomenon we Study

Typical humans, from children to adults, are capable of moving and acting in their environments, manipulating objects efficiently to perform a variety of tasks even as the space and their own bodies may change over time. But where does the knowledge necessary for these skills come from? Two possible explanations propose that the knowledge is innate or provided by an expert designer in a one-time transfer prior to its application, as it would be for a carefully programmed robot. This is clearly not the case for humans.

Human infants begin with neither the spatial knowledge nor sensorimotor proficiency to execute the actions that they carry out as older children and adults, as evidenced by their behavior. Initially, babies engage in motor babbling, making seemingly random motions, and do not appear capable of achieving more coordinated goals. The responses to stimuli and the motions carried out by young infants are not consistent with innate spatial knowledge that would be necessary for manipulation actions. Specifically, infants can be observed to be incapable of reaching and grasping objects in their environment, even within their peripersonal space, or the space that should be in reach of their hands. The knowledge for these actions is not innate or present at these very early stages.

1.2.1 Early and Late Learning Phases

Unlike an artificial agent receiving new programming, the difference in capability between infants and adults is not realized in a single event, and occurs over several developmental stages in a long learning process. While behaviors vary between individuals, these stages can also be observed (especially by trained developmental psychologists) as clear landmarks between the initial motor babbling and a fully developed human.

Our work reflects the division of actions into their early and late forms that has been observed by developmental psychologists. We make a similar distinction, defining an early action as a method that can be learned according to a small number of features or patterns, but that is sufficient so that when executed in this manner the action's goal is reliably achieved. An early action may be inefficient and appear awkward due to a lack of optimization and practice. For example, early reaching exhibits a sequence of jerky submotions instead of a single smooth trajectory found in late reaching. For some actions, the technique will only be considered early when it is the very first one discovered, while others in our work have multiple early forms as additional salient, human-understandable features are iteratively added to the model for improved reliability. The changes between stages of early actions will focus on increasing the success rate of accomplishing the goal, often by allowing a prerequisite to be considered or more reliably met. Iterations of early actions are seldom different in terms of the efficiency with which underlying component motions are carried out. Our model only uses the knowledge representation of early actions, and the scope of this dissertation ends with reliable early reaching, semi-reliable early grasping, and reliable early placing.

Later forms of actions emerge as the agent gathers larger amounts of data, and the changes necessary to produce them can be derived from pixel-level features or practice that reveals precisely which motor commands will create the smoothest, most skilled motions. These learning stages may be best modeled by RL including deep reinforcement learning methods. Learning these actions from a minimal or random initialization has a high complexity in time and data, but future works with these techniques could use the results of our work, the learned spatial model and accompanying early action forms, as scaffolding to reduce the necessary training time. Late actions will be smoother and in some cases more reliable than their early counterparts. The defining difference between early and late is the types of learning methods and features that allow these improvements.

1.3 Why Reaching, Grasping, and Placing?

1.3.1 The Importance of the Grasp Action

An agent that can operate and survive independently must be able to observe its environment and perform actions to change the state of itself and its environment. Grasping is a fundamental action for the manipulation of objects. The grasp is a powerful and versatile tool for an agent due to the expanded level of control over an object. When an object is grasped, it enters a new qualitative state where it is temporarily bound to the hand, acting as a part of the agent's self. In particular, this binding can be observed through continued motion to the object, with each movement highly correlated with the direction and magnitude of the hand's movements.

When the grasp is performed successfully, the object's grasped state continues as long as the agent chooses, and the agent has more choice of how the grasp influences the environment due to this control of the time and place that the grasp ends. The grasped state continues until the agent performs an ungrasp action, and then the object transitions back into a state where it is no longer bound to the hand or moving with the self. The persistence of the effect is one of the key differences between the grasp and simpler techniques the agent can use to trigger an isolated motion from one quasi-static state of an object to another, ranging from small contacts to shift it or tip it over, to more forceful contacts that knock the object to the side or clear of an area. These bumps only give brief control of the object in a single direction, and sensitivity to differences in the amount and application of force during the brief contact make the control less reliable for achieving a predictable result. Using a push action to steadily apply a force allows control over a duration like a grasp, but the range of motion possible is limited to a surface. With a grasp, the agent can move the object in 3D space for the full duration, and can control the motion precisely due to the already-present knowledge of how to move the hand itself, which the object follows.

1.3.2 Reaching as a Precursor to Grasping

While the grasp is the key manipulation action for exercising this level of control and going on to several higher order actions, we also consider the reach action that causes the previously discussed bumps to be an important benchmark for an agent. Reaching is significant because grasping is highly complex, and can only be performed successfully when a number of conditions are met simultaneously. By contrast, reaching and causing a bump has only a singular condition of moving the hand to the same position in space as the object so that they collide and the object is moved out of the way. In a developing agent this is initially significant for the ability to influence nonself foreground objects. Observing the persistent change to an object caused by the bump at the end of a successful reach may be important to conceptualizing the existence of these objects in peripersonal space, as it will be efficient to model the change as the new appearance of an object that has been moved to a new state. This evidence of additional objects is unique to reaches and other actions that causes a persistent change, and will not be provided by motions that only move the hand. For example, a move that occludes an object causes a temporary change to its distinct patch in the visual field, but this also occurs for the background, and does not suggest the patch corresponds to an object that can be manipulated. Moves that do not occlude objects are also missing evidence of the objects' existence. Therefore, successful reaching is important for learning that it is possible to interact with the environment, and for identifying the specific parts of the environment that can be interacted with.

Reaching will continue to be important to the agent in our work because reaches provide a constrained set of experiences where grasps can be observed more frequently by chance, which is an essential step in our learning model. The complexity from grasping comes from the need to open and preshape the hand, then come into close proximity with the object with a suitable approach vector, and then close the hand securely around the object to fix it to the hand during subsequent motion. A situation where all of these conditions are met by random chance simultaneously will be too astronomically rare to observe - even if this unusual event occurs within the span of the agent's experience, without very large-scale data collection (which may employ multiple robots over a long period of time) there will certainly not be enough observations to find patterns between them and draw conclusions about how to perform the grasp action.

We incorporate two measures observed in infant learners to mitigate this complexity. The first is that as in the natural setting, the agent will learn the reach action as a precursor to the grasp action. When executing the reach action, the agent satisfies the same condition of bringing the hand to the object as is needed for the grasp approach. This is a helpful partial step due to its observable bump effect. Without performing a successful reach, the agent could have all of the other factors correct but not observe any change in the object. The reach is not dependent on any of the additional grasping conditions, and with these varied during the grasping attempts the agent will observe both the typical result of a quasi-static change and in rare (but less rare) cases the unusual result of a grasp and the associated change to follow the hand. As it is present in infants when they learn to reach and grasp, we also simulate the *Palmar reflex*, which in humans causes the fingers to close around an object pressed into the palm. This is an important learning aid since it also reduces the number of factors for the agent to put in place at once. In this case, the Palmar reflex closes the hand at the correct time, so that all that is necessary to complete a grasp is a successful reach with the correct approach direction and hand pose. As the agent becomes more skilled with grasping the Palmar reflex fades and the agent will need to compensate by consciously closing the hand, which will allow more selective grasps - not picking up objects unintentionally and finding the best grasp point before initiating the grasp - and will prevent an object that is being intentionally ungrasped from being caught reflexively.

1.3.3 Ending Grasp-Initiated Manipulation with Intentional Placements

The work in this dissertation also includes the learning of the place action, which is combined with the grasp action into an early move object action, which can also be called pick-and-place. The place action must begin with an object grasped in the hand, and consists of an ungrasp to release the object so that it comes to rest in a new quasi-static position within a specified qualitative location. In order to reliably place into the desired location, the agent may need to use the move hand action to perform the ungrasp at a suitable position.

We choose the pick-and-place action as the final learning step within this dissertation for our agent, as demonstrating the acquisition of the knowledge required to perform each component of this action sequence is meaningful for multiple reasons. One of the straightforward sources of utility for the grasp action is realized from the extension to pick-and-place, where the object can be moved from an undesirable location where it may be in the way, or moved to a desirable location where it serves a new purpose. The reach, grasp, and place component actions of the pick-and-place also allow a suitable demonstration of our model's learning method, both on a series of increasingly complex actions (moving the hand, reaching to objects, and grasping objects) and on the place action, which shares fewer components. As a practical robot learning problem, these actions require the embodied agent to 1) learn to control the movement of its degrees of freedom in the hand and arms, 2) learn to use the robot's sensors and end-effectors to detect and manipulate objects, and 3) learn to form and execute plans using sequences of the component actions to succeed at a larger goal. This set of actions has also been observed for decades in human infants, and we can compare the behavior of our agent executing these actions with the behavior of the infants. Finally, this endpoint allows this work to easily serve as a starting point for other works, such as those with symbolic planners where the ability to move an object between locations is an assumed primitive action. The capability to pick-and-place can be combined with higher level search or reinforcement learning to solve problems like the Tower of Hanoi, or to perform real world tasks like cleaning and sorting.

1.4 Learning from Unusual Events

1.4.1 From First Motions to a Model for Moving and Acting

Human infants begin with seemingly random motor babbling motions, which are made even less predictable as the infant adjusts to the changing size of its own body and develops the strength and coordination of the muscles it must use. Our embodied robotic agent will begin with a similar set of capabilities. The agent quickly learns to use a proportional controller to set the velocities of its joints to move each to a desired set point, and then to use these control laws in parallel to move the entire arm from one configuration to another. During these first movements, the agent learns aspects of its own appearance and can gain a meaningful concept of self. A minimally complex model where the world is viewed as static can be improved upon by modeling the self. As the agent moves single joints and the full arm, it can identify the portion of its field of view that changes in directions and with timings correlated to the motor controls it has used. These portions are treated as the self, and the distal part of the self that moves most quickly can be defined as the end-effector or hand. The combination of the modeled appearance of the self and the rest of the environment as a static background is a much stronger prediction of the true appearance of the workspace.

After learning to recognize the self and to move the arm to any set point, the agent can carry out a large number of motor babbling moves to poses throughout the peripersonal space. Within these motions we assume that the agent observes its first typical result of its move arm action - when the arm is commanded to a new configuration, the position and appearance of the arm may change, but the static background model for the environment that is not occluded by the arm remains unchanged. These movements also give the agent the opportunity to observe the arm (the visually observable part of the self) in multiple poses. Rejection sampling is used to bias the poses generated as the targets of these motions, to ensure that each is safe and visible, protecting the fragile robot and facilitating learning from visual features. In our model, the agent will record the data from its visual and proprioceptive senses at each pose, and each pairing of sensory data will serve as a discrete mapping between the visual image space and the proprioceptive joint angle configuration space at the values of the observed pose. We assume that images are taken from a fixed perspective, and the agent does not use any degrees of freedom to adjust the position of the head or its gaze. This fixed perspective allows a straightforward comparison of stored and current percepts. The model associates the stored percepts within a graph node, and connects nodes with edges when the move between the poses they represent is feasible and safe. The determination of edge safety is initially based on edges that have been traversed during motor babbling - the motions that have already been made should continue to be safe and are suitable for edges - and the graph can be expanded to include additional edges for motions that are similar to those for the initial edges. The overall shape of the workspace is assumed constant, so nodes and edges that have been added will remain valid throughout all of the agent's learning and acting phases. In particular, the small table in front of the robot is present during motor babbling, so all proposed nodes inside or below the table's surface will be rejected by the sampling, and no motion along an edge will cross through the table. This prevents the agent's action trajectories from having significant collisions with the static background. The full collection of nodes and edges will be referred to as the Peripersonal Space (PPS) Graph, and will be used extensively as the agent identifies features relevant to making actions more reliable and plans motions to execute those actions.

1.4.2 Defining Actions from Observations of PPS Graph Motions

While the typical result of the arm motions is observed more frequently and can be understood early enough for the creation of the PPS Graph, the agent will be able to identify one or more types of atypical results as it continues to perform the motions. With enough executions of the move arm action and with a sufficient number of foreground objects present to make interactions frequent enough to be observed, the agent can determine a set of trials with qualitatively different results from the results in the typical set. In this case, in addition to the motion and corresponding visual of the arm, a nonself object will also have a new percept. As the unusual result type occurs much less frequently than typical results, and the individual outcome of each unusual result is more difficult to predict accurately with the agent's current knowledge state, an application of intrinsic motivation rewards the agent for observing the event more often, so that it can better understand the new phenomenon. By understanding the phenomenon, the agent can predict its result more accurately, and with a higher rate of increase of this accuracy than could be obtained for phenomenons that are already well understood or that are at this time too complex or stochastic in the current world model. In order to receive the intrinsic reward, our agent will formulate a new action defined by a goal of causing the type of qualitative state seen in the unusual event, and then must search for features that make the action reliable so that in practicing the action it encounters the event and can improve.

In the following paragraphs, we briefly discuss additional instances of this learning from unusual events pattern. In this pattern, the agent executes over multiple trials a motion it is innately capable of or an action it already knows how to perform. By observing the results of each trail, the agent can determine the typical result of that action. Using clustering, the agent can also identify one or more types of atypical results, which stand out as separate clusters with fewer members than the cluster of typical results. In this work, the agent only identifies a single unusual cluster at a time, but subsets may be found within this cluster in later iterations, as is done for the advancement from reaching to grasping.

1.4.2.1 Observing Bump Events and Defining a Reach Action to Repeat Them

The unusual events observed when a nonself object's visual percept changes will be collectively known as bumps. For bumps, one metric that reveals them as an unusual cluster of results is the intersection over union (IOU) of a visually distinct connected component of pixels in percepts taken before and after a move of the arm. In the typical case, the IOU is approximately 1, but for the atypical cases of bumps, the IOU will be significantly lower than 1, intuitively because it has been bumped to a new location and this appears in a at least somewhat different set of pixels in the agent's view. By observing that bumps can occur, the agent can derive the existence of objects, as recognizing that there are other objects within the foreground will be the most efficient and accurate way to model the interactions between these parts of the environment. Without the abstraction to objects that allows the data to be grouped in meaningful ways, the agent would be forced to consider these regions with pixel level data, where reasoning about their physical interactions would be intractable. While these objects will be able to change position like the self, the agent will be able to differentiate between the self and these other objects because they cannot be directly controlled and do not move in correspondence with the agent's motor commands (without the later capability of grasping to bind them to the self).

Distinguishing bumps as a set of results that qualitatively differs from the most common result leads to additional learning steps motivated by this unusual event. Due to its initial rarity and novelty, along with the presence of learnable conditions that make bumps conditionally more or less likely to occur, there is a high expected intrinsic reward for focused practice on a reach action that causes bumps. The agent uses the goal of causing a bump to define the reach action, which will be planned and executed using a sequence of one or more of the move arm actions that the agent is already capable of. By making this action reliable, the agent can maximize the reward it receives over time. An important detail is that the agent does not provide itself a reward each time that another event that can be described as a bump occurs, which would allow arbitrarily high rewards by simply executing more trials and without learning to produce bumps more reliably. Instead, the agent's intrinsic reward is based on the rate of increase in the reliability in the reach action over time. This creates the highest reward when a new action is significantly more reliable with the current feature set or in the current set of trials than in the previous sets, and relatively low rewards when there is little improvement, either when a new feature is unhelpful or the action was already very reliable in the previous sets.

Our agent identifies features that can be measured in past recorded trials and differentiate those with bumps from those without bumps. These same features can predict whether a certain sequence of arm movements is likely to cause a bump in a new trial, so the agent can evaluate all possible movement paths and choose one that is, according to its model, most likely to produce a bump. These features can be refined over time using the trials where the reach succeeds or fails, so that the feature values that are considered the most reliable are consistent with these results. The agent improves the reliability of the reach action through this refinement and also by adding features as they become evident. In particular, the agent will choose a motion that ends with the hand attempting to occupy the same region of space as the object, and among multiple options for doing this the agent will prefer moving the center of the hand to the center of the object. The learning process for the reach action has these positive qualities to note: 1) The agent learns a reliable reach action without any expert provided knowledge about what a bump or reach is, and without assuming any external utility to causing bumps, 2) The intrinsic reward is obtained when a reach successfully produces a bump, and not for efficiency. This focus on the goal only allows the agent to learn the early form of the action, which is far less data or time intensive than a process to learn a more efficient late or skilled action method, and 3) The agent learns reliable reaching with a small number of features (much smaller than the number of features in a deep network model), and each of these features corresponds to a high-level concept that is humanly understandable and thus useful in the discussion of how the necessary knowledge for the action is acquired and represented.

1.4.2.2 Learning Grasping from an Observed Unusual Result of Reaching

The pattern of observing typical and atypical results of known actions and then learning a new action to reliably repeat the atypical result can be applied for an arbitrary number of iterations. In each case, the agent is intrinsically motivated to improve the reliability of the new action to obtain the highest reward, which correlates with the fastest rate of improvement in understanding and reliability that it can achieve. In our work, the agent learns to grasp, place, and move objects through these additional iterations of the pattern. The following paragraphs will outline the learning process for each of these actions.

Once the reach action is suitably reliable, the agent can plan reach trajectories and see the typical result of their execution, which is a quasi-static change in the target object's visual percept. At this time the agent does not differentiate by the magnitude of this change as long as it is large enough to be differentiated from noise-based differences between multiple observations of a stationary object. At this time, we also do not allow a learning phase for identifying qualitatively different types of quasi-static bumps, though in future work the agent could apply the intrinsic motivation to identify and more reliably produce specific categories of bumps, such as those that do or do not change the orientation of the block via tipping.

In our current method, the agent begins a new stage by observing a sufficient number of accidental grasps to identify a cluster as an atypical result of reaching. This qualitative difference is more significant than the difference between types of quasi-static bumps by creating a new dynamic relationship between the object and the hand, and there is also a greater increase in flexibility and utility the agent gains by being able to manipulate objects with a grasp. Observing enough accidental grasps is aided in or experimental robot platform by the presence of a simulated Palmar reflex that closes the hand when it is suitably posed relative to the object to grasp it, and by first observing the activation of this reflex as an intermediate unusual event before requiring that the closing of the hand produces a successful grasp. Our intuition is that reliable grasping requires the fingers to be able to close around the object. We break this into three conditions:

1. the hand is fully open before the final motion of the grasp approach

2. the grasp approach direction is aligned with the vector along the axis of the forearm, pointing out parallel to the gripper fingers

3. the hand is oriented by the wrist such that the plane that passes through the gripper fingers is perpendicular to the major axis of the grasp target

When these conditions are met, the grasp approach allows the opening of the hand and the grasping region between the grippers to intersect with the target before the exterior of the hand, which would bump it away and prevent the grasp. Rotating the hand so that the grasp is attempted on a relatively small cross section of the hand provides the best chance that the object can fit between the grippers, which is also aided in our experimental robot platform by fully opening the hand to increase the size of this grasping region. Passing between the gripper fingers is necessary to trigger the Palmar reflex for an early reflexive grasp, and is also a prerequisite for later grasps with deliberate commands to close the hand. In addition to allowing the object to fit inside the hand, a wrist orientation perpendicular to the major axis of a rectangular prism target creates a reliable grasp point. In this situation, the grippers close evenly on two flat surfaces of the object for a firm connection that is less likely to squeeze the object out of the hand or allow it to slide out of the grip during subsequent motions. The agent experiments to identify features and ranges of values for those features that best predict whether a reach trajectory will meet these conditions and produce a grasp instead of a bump. Early grasps planned in this way become about 50% reliable.

1.4.2.3 Further Learning from Grasped Object Typical and Unusual Behavior

Once the agent has grasped an object, the typical result of the grasp involves the object moving with the hand, with the expectation that the percepts of the hand and object will be highly correlated. This will be the case for any moves within the PPS Graph and for almost all adjustments to the agent's degrees of freedom in the hands and arm. However, in the smaller set of cases where the agent chooses to modify the degree of freedom for the openness of the hand to a sufficiently higher percent open, the object is released from the hand to a new quasi-static position and does not exhibit motion correlated to the motion of the hand for future arm moves. As it ends the grasped property for the target object, This event will be called an ungrasp. Within the set of all grasps, a smaller set will be successful placing actions, where the agent ungrasps an object and its new quasi-static position is within the desired location. In this work, locations will be defined as qualitative regions, such as a visually distinct patch on a table. An object will be in one of these locations when it is in contact with the interior or boundary of the region, without requirements of a specific pose or coordinates. This definition is more appropriate for the early developmental agent and requires a lower level of precision that limited visual sensors can verify and limited motor coordination can achieve. In order to make the place action reliable, the agent must learn to fully open the hand for the best chance of the ungrasp succeeding, and to do so after moving the hand to a pose that is expected to maximize the probability the ungrasped block drops into the desired location.

1.4.3 Combining Known Actions to Form Higher Order Actions

In this work, the agent is assumed to be able to compose known actions together into new higher order actions. This can involve performing actions in parallel, as is done early on to create the move arm action from the action to move each degree of freedom individually, or sequentially, as is done to plan reach trajectories using several move arm actions, or for placing to combine a similar sequence of move arm actions with ungrasping. At the conclusion of this thesis we demonstrate the agent's ability to pick and place objects, which it learns to do by learning the component actions and then composing them in the proper order. Within the search space, the branching is reduced by which actions are applicable at each step of the process (for example, the agent does not consider ungrasping when no object is grasped), and the search can be further restricted to only branches that produce qualitatively distinct results from those offered by a shorter sequence. This allows the agent to quickly discover the correct method for pick and place, it first reaches to and grasps a target object, then moves the arm and hand together with the object, and finally places the object with a suitable ungrasp. The pick and place action will be semi-reliable, mostly constrained by the reliability of the grasp component action, and the agent's ability to perform it will demonstrate the results of multiple applications of the intrinsic motivation learning pattern and relate our work to those in other fields that begin with a reliable pick and place action.

1.5 How Unguided is our Agent's Learning?

We aim for our agent to resemble the example of human infant learners, with methods that allow for the agent to learn from its own experience with minimal outside guidance or structure for the learning activities. The agent should be intrinsically motivated to study unusual events that are qualitatively different from the typical results of the agent's current set of actions. The agent will then expand its set of actions to include actions to repeat new phenomenon, and will focus on improving the reliability of actions with a pattern or contingency that can be observed with the agent's current senses and knowledge state, as these actions will be improved most quickly, yielding the highest expected intrinsic reward. While these are true for the agent in our ideal model, we have had to introduce some guidance into our learning methods for efficiency in order to demonstrate specific landmarks within the timeline of this work and the resources of our laboratory.

1.5.1 The Presence of Learning Stages Focused on Single Actions

A noteworthy difference from our ideal learning algorithm with a fully unguided agent is the presence of learning stages, where we limit the agent to practice a single action at a time, rather than a naturally interleaved learning of actions or processes that may improve the policies of multiple actions using the same trials (for example, the ideal unguided agent could opportunistically use information from the same trial or set of trials in continued practice toward making reaching fully reliable and to learn the first key considerations that make grasping reliable). In rough correspondence with an agent free to choose the action with the highest intrinsic rewards, we start the agent's focus on an action when its prerequisites have been learned well enough for the action to be learned efficiently, and end trials with that action when the agent experiences a plateau in the reliability of the action, which would lower intrinsic motivation to continue to a point that would drive a natural switch in focus.

Within a stage of focus on a single action, we also provide the agent with potential features, either as single features or as features that it can select from a relatively small group of options. We do not provide any information of how to use these features. The agent must process its current and stored past percepts without further input to determine the values of each feature, and the agent is not instructed on the values that indicate reliability, so it must still learn these aspects autonomously from experience and using general statistical analysis methods. However, we acknowledge that a completely unguided agent would have a larger capability for feature generation or would select features from a larger set. Another effect of the agent being partially reliant on the experimenters for feature generation is that the level at which action reliability plateaus may be influenced by the set of new features to test being exhausted. This - along with sensorimotor limitations and the higher difficulty of the grasping task - can explain that we were able to instruct the agent to move on from reaching after seeing a plateau at essentially 100 percent reliability, but in our current experiments the agent has had to end its focus on grasping with a semi-reliable grasp action (57.5% reliable).

A satisfactory level of reliability is different for different actions in our work to date, but we claim that this does not detract from our claims of foundational learning or unguided exploration. State of the art methods can be observed to have a similar result pattern of being almost completely reliable for reaching, but less reliable for grasping. We do not intend to compete with these methods, which often begin with much stronger information or learn from orders of magnitude more experiences. Our model is also meant to serve as scaffolding for a later model, so its final performance level before the additional learning with other methods is not critical to the success of this work. However, our agent will be expected to become more reliable at reaching and ungrasping than grasping, which is a much more complex action and requires more features and more precise control. Our final demonstration of the ability to move objects will be primarily limited by the bottleneck of grasping reliability.

The presence of learning stages is our current method for restricting the agent to the actions within our focus area. In a natural learning environment, the agent would be concurrently learning a hierarchy of actions, alternating making notes of observations that provide information for how to make actions in different branches of the hierarchy reliable. While an infant may learn to reach and grasp in the same time windows as it improves its vision, locomotive, or other skills, we prevent our agent from focusing on any of these tasks outside of our action hierarchy branches, those for reaching and grasping and for ungrasping and placing, and then the combination of those actions in sequence. This restriction is also necessary for the physical capabilities of our robot, which cannot adjust its viewing angle or focus of the eyes, and does not have the ability to crawl or walk. We also omit learning stages for low utility actions within these action hierarchies, such as learning to reliably occlude an object.

While it has been necessary for us to impose a set ordering of learning phases for practical considerations within this thesis work, this added structure is advantageous for evaluating the contributions of the work. Results of the natural learning pattern are more difficult to follow and record since any given action may be refined or not addressed at all within any set of trials. Further, we wish to demonstrate a specific set of actions can be learned to a reasonable level of reliability within a set period of time, which would not necessarily be possible if the agent was free to choose any action within the hierarchy at any time. By enforcing learning phases so that the agent addresses each action in the sequence and then moves on after reaching the reliability threshold for a satisfactory evaluation in this work, we can most efficiently reach these results. Learning a single action at a time also gives more reliable information about the sample complexity of each learning task, and allows for more clear reporting of the results and on which trials any breakthroughs have occurred.

1.5.2 Behavior Based on Assumed Intrinsic Reward Values

It should also be noted that our learning model does not include an explicit calculation of intrinsic reward. However, we claim that our agent is intrinsically motivated to learn actions to repeat unusual events more reliably as this would maximize an intrinsic reward such as the one proposed by Schmidhuber [42], where the reward is proportional to the rate of increase in performance of a task. This is justified within our use of learning phases because we are able to have the agent attempt and observe the action that is at the appropriate level of complexity for the agent to learn most efficiently about it, maximizing the reward. We always start phases when the agent has the necessary prerequisite actions already sufficiently reliable and the agent has the ability to consider all necessary features. We also end the learning phase consistently when the agent has no more room to improve, either because it is fully reliable or because the agent's set of features cannot significantly increase the reliability further in our methods.

We intend this work to contribute an application of intrinsic motivation by choosing the phases according to which actions should produce the highest intrinsic rewards as actions and predictions become more reliable at the fastest rate. This will be of value to the intrinsic motivation community, since it will demonstrate a new method that intrinsic motivation can be applied to where it allows for successful learning. It is not intended to advance the theory of intrinsic motivation or propose a new method for implementing intrinsic motivation, but these do not interfere with the goals of our work.

1.5.3 A Note on Assumed Sensorimotor Capabilities

The assumed visual capabilities of our agent and the built-in motor command processing of the physical robot could be considered to be sources of guidance. However, we claim these assumptions provide an amount of information that is relatively small compared to alternative methods. Further, while we do favor biologically plausible features and methods, our focus is not on the computer vision aspect of this learning problem, so we have made simplifications to allow our learning method to function without the implementation of one of the sophisticated techniques being developed in the state of the art of that field. For the motor command processing, we have demonstrated that our agent is capable of learning to bring individual joints or the arm to a stable set point using the built-in systems, and that the underlying methods are important for the safety and maintenance of the robot's operation, and also do not have a significant impact on the learning process we focus on.

1.6 Contributions

1.6.1 Demonstrated Learning

The Peripersonal Space (PPS) Graph. By unguided exploration of the proprioceptive and visual spaces, and without prior knowledge of the structure or dimensionality of either space, the learning agent can construct a graph-structured skeleton (the PPS Graph) that enables manipulator motion planning by finding and following paths within the graph. The graph representation requires only limited knowledge of the attributes of the nodes, and no knowledge of the dimensionality of the embedding space. The PPS Graph provides a discrete approximation of the agent's configuration space as sampled at the nodes, and can be extended with the local Jacobians of each node to interpolate the rest of the continuous space. We have demonstrated that the local Jacobian allows more precise control to complete actions more reliably, and in future works expect work with the local Jacobian and improvements to the dynamical model of hand motion to explain qualities of the early actions and to advance the agent toward the smooth and directed late actions of an adult.

Learning Reliable Reaching. By learning conditions to make a rare action (i.e., reaching to cause a bump of a block) reliable, the agent learns a criterion on perceptual images (stored and current) that allows it to select a suitable target node in the PPS Graph. Motion to that target node accomplishes a reliable reach. The PPS Graph representation accounts for reaching in a way that matches striking qualitative properties of early human infant reaching: jerky motion, and independence from vision of the hand.

By interpreting the target node and its neighborhood as a sample from a continuous space, the agent can approximate the local Jacobian of the hand pose in perceptual space with respect to the joint angles. This allows it to adjust the trajectory to make reaching more reliable.

Learning Reliable Grasping. At this point, reaching reliably displaces the target block. Occasionally, instead of quasi-statically displacing the block, the block continues to move, to follow the subsequent motion of the hand. The agent is intrinsically motivated to change its focus from learning to reach to learning to reliably repeat this newly observed phenomenon. Making this result reliable requires several distinct conditions. The innate Palmar reflex makes these rare events common enough to learn from. Conditions on gripper opening, wrist orientation, and approach direction can all be learned based on positive feedback from the unusual block motion.

Learning Reliable Placing. The agent learns to place in two stages. Successful grasps put the agent in a novel state of having an object moving with the hand. The agent learns to ungrasp by attempting changes to each degree of freedom. It finds that only adjusting the gripper aperture degree of freedom toward open reliably releases the object, returning to a familiar state with no object grasped. Starting with a firm grasp, movements between nodes or any other changes to the degrees of freedom do not result in an ungrasp.

After the agent has learned to ungrasp, it can refine this action to find the subset of ungrasps that place the object into a desired location. As the agent has limited visual and motor capabalities, the locations are relatively large and qualitatively defined, and do not require a precise location or particular orientation of the object. The agent learns that if it moves the hand, along with the grasped object, to a PPS Graph node that is close in image-space to the center of the location, and then opens the grippers for an ungrasp, that the object reliably comes to rest in the desired location.

Sequencing to Reliably Pick-and-Place. We show that the agent can perform a simple action-space search to combine the learned grasp, move, and place actions in sequence into a pick-and-place action. This action is semi-reliable, with its reliability mostly determined by the reliability of the grasp action, as it is the most complex and is the least reliable learned action given the current method and set of features. This action could be used as an action primitive in future robotics and learning works.

1.6.2 Significance for Developmental Psychology

There have been recent impressive results from unguided end-to-end learning of multiple games [45, 44]. Using very large-scale data collection and deep learning methods, a robotic agent may also learn the necessary coordination between vision and the motion dynamics of the hand to achieve a reliable grasp action [38, 31, 30]. While these results are very exciting, some limitations come from the need for vast amounts of training experience, and the lack of transparency and explainability of the learned knowledge. Our contributed learning method addresses these limitations and instead has these desirable unique properties:

- **Agent-defined Actions.** Our agent defines actions to learn based on its own observations. We avoid providing an error metric, a requirement for reinforcement learning methods and the back-propagation training of deep networks, as these metrics must build in some information about the action to be learned.
- **Suitable for Learning Multiple Actions.** Our learning method may be applied in several iterations for the agent to learn multiple actions. In each use of this pattern of observing unusual results for a known action and defining a new action to reliably repeat them, transfer learning can be used to achieve a baseline performance using the learned prerequisites of successfully completing the known action to attempt the new action. This removes the need to start from scratch or to create a new function or structure for each action.
- **Understandable Features.** Our model is fully transparent and the learned features are easily interpreted by humans. These understandable features and the methods that produce them allow us to add more to the discussion about how an agent learns actions in peripersonal space than if our method operated as a black box.
- **Low Sample Complexity.** All learning is performed within thousands of motions, with the majority of these taking place while motor babbling to build the Peripersonal Space Graph. Individual actions are learned within tens or hundreds of trials. Deep reinforcement learning and end-to-end networks produce highly reliable actions and can be argued to have little assumed knowledge and guidance, but have a well-documented issue of a sample complexity that is orders of magnitude higher than our method's.

We hope that our work on reaching, grasping, and placing in peripersonal space can illuminate the kinds of intermediate states that a developmental learner goes through. Those intermediate states make the structure of the knowledge more comprehensible, and the learning stages between them more efficient. Combining the strengths of these approaches could be important.

Our agent has learned to complete actions with trajectories that resemble early infant behaviors. Where these similarities are observed between the robot and infants, our computational model suggests hypotheses for features and parameters that are sufficient and may explain these behaviors. Our formalization may be used to design experiments that test these hypotheses.

1.6.3 Significance for Other Fields

Reinforcement Learning. The action trajectories planned by our agent produce positive examples with a much higher frequency than random motions. These initial successes can be provided as scaffolding to efficiently train a reinforcement learner to this level of performance. Further training could optimize the trajectories into a smooth, skilled motion.

Symbolic Action Planning. We have demonstrated an autonomous learning sequence that produces a pick-and-place action. Symbolic planners often treat pick-and-place as a primitive action, and our work provides a concrete justification for its use without assuming it will be innate or require guidance.

1.7 Outline of Dissertation Chapters

We present related works in Chapter 2. Chapter 3 provides the formalization for our model, including the general PPS Graph representation and a summary of all features that are learned by the agent to make each action reliable. Chapters 4 through 7 present experimental methods and results, some of which have already appeared in our published works [22, 23, 24]. Chapter 4 focuses on learning to move joints and the arm, and the creation of the Explored PPS Graph, the current implementation of the general model. Chapter 5 involves the observation of the bump unusual event and learning the reach action to repeat it. Chapter 6 describes the process of learning to grasp and the implications of moving the robot to a new environment. Chapter 7 presents experiments for learning to ungrasp and place, as well as sequencing grasping and placing to form a pick-and-place action that is the final demonstration of the agent's capabilities in this work. Chapter 8 concludes with a discussion of the contributions and importance of this work, as well as next steps and opportunities for future work.

CHAPTER 2

Related Work

2.1 The Human Model from Developmental Psychology

To discuss how an autonomous agent can learn about peripersonal space and to perform manipulation actions it is helpful to draw from the concrete and well-studied example of humans, who typically undergo these learning stages as infants. We can draw on the extensive literature of the field of Developmental Psychology, which documents decades of observations and experiments concerning the process infants take to learn specific actions like reaching and grasping. While individual infants exhibit varied onset times for the actions and may employ different early techniques, the order of the milestones and the qualitative advancements that mark arrival at a new learning stage are well understood [5]. For typically developing infants, the "pre-reaching" phase begins at birth and lasts for 15 weeks on average. During this time infants can move in response to the stimulus of a visual target, but these motions do not generally bring the hand in contact with the target. Reach onset typically occurs at about 15 weeks, when the infant can first use intentional reaching motions to make contact with the target with some reliability. Over the period from 15 weeks to 8 months the infant becomes more successful. This early reaching improves primarily in the frequency of achieving the goal state, with the trajectory for the reach remaining primitive and inefficient. Instead of a skilled technique of a single smooth motion, the early reaches are executed with a jerky sequence of submotions. The current literature provides multiple explanations for these submotions, and some may be corrective submovements [56], and some may be the result of underdamped oscillations as the infant moves to an equilibrium point near the goal using its minimally tuned motor control system [49].

Following the work of Piaget (1952) [36], there was a decades-long consensus that early reaching required vision of both the target object and the hand, and that the reach was performed as a visual survoing process to incrementally bring the images of the hand and the object together. However, this was challenged when Clifton et al. (1993) [11] showed that the movement pattern and success rate of early reaches is unchanged when the infant is denied vision of the hand. This study showed that when infants were in a dark room where the hand would not be visible, and then were presented with a glowing target object, reaches could be performed in the same manner with the same degree of success.

A follow-up work by Clifton et al. (1994) found that not only were infants capable of reaching equally successfully for a lit target in a dark room as with normal lighting conditions, but that the reaches were also performed with the same kinematics [13]. They propose that visual guidance of the hand may be a late development rather than an early one, and that it remains useful into adulthood for difficult and precise manipulations. They also conduct experiments with sounding objects in dark rooms, and find evidence that the infants can differentiate the noise of objects that are in reach or not. However, auditory information does not appear to allow for the same precise infant reaching, and most attempts missed the target completely [13]. A further study on 7-month old infants found that the visibility of the target does not appear to affect the timing of reach onset, and that infants will reach and grasp objects presented to them even if they are only glowing or sounding [12]. They also conduct an experiment where infants are presented with an object in normal lighting at most six times before being presented with the same object in the dark. They conclude that the infants are capable of quickly constructing the representations necessary to perform the reach in the dark successfully. However, they do acknowledge that their claim was not universally accepted, primarily due to objects that the success may be memory-based, due to the reach trials in the dark room being conducted just after the lights went out. Adolph and Berger (2005) [2] provide a thorough survey of significant observations of infant behaviors, including the presence of "balletic" and cyclical motions and timelines for when infants tend to exhibit reflexive and goal-directed behaviors. This survey includes an overview of other new studies in response to the findings of Clifton et al. [11, 13, 12]. Corbetta et al. (2014) present the result as a new clear consensus: "from their earliest attempts, infants can reach in the dark toward a glowing target without seeing their hand [11]." [14, p.1]

While the results of Clifton et al. [11, 13, 12] supported a new theory that vision of the hand is not important during early reaching, when infants are approximately one year old, vision is believed to become important again so that the hand can be configured and oriented to produce the desired interaction with the reach target once contact occurs [5]. By the age of eight months, the infant's reach motion becomes smoother and more similar to a typical adult reach, which can be described as "a single motor command with inflight corrective movements as needed" [5]. Many aspects of the evolution of reaching motion control can be explained by dynamical systems theory [49] and by optimal control and reinforcement learning [7].

While early reaching does not rely on having a current visual feed of the hand and visual servoing, vision does appear to be important for much of the process of learning and then using manipulation actions, especially for building spatial mappings before reaching and later in development for maintaining smooth motions and preshaping the hand for complex tasks later in development [5].

A theoretical branch of the literature discusses that it is necessary for the agent to build a useful mapping between the two- or three-dimensional visual space and the configuration space of the arm (which has a dimensionality determined by the number of degrees of freedom of the arm) before reach onset can occur. The agent's spatial model must also be able to represent additional features of itself, objects, and the environment before grasping can occur. The questions surrounding the mapping are well summarized by Corbetta et al. (2014) [14, p.2]: "*What remains unclear is how looking at the object and bringing the hand to that location occurs at first when infants perform their initial intentional attempts to hit the target. What visuo-motor mapping process allows this to happen?*" Several theories have been suggested for the nature of this mapping and how it is constructed by human infants.

The creation of such a mapping is described as *multisensory integration* by Bremner et al. (2008) [8], who focus on the sensory modalities of touch, proprioception, and vision. Their proposal includes two distinct neural mechanisms. One uses an egocentric frame of reference for object positions, and assumes a fixed initial body posture and arm configuration. The second is used with a world-centered frame of reference for object positions, and includes the ability to factor changes in the body and arm into its spatial relation mappings to remain effective with fewer assumptions.

Other studies on learning to reach highlight the importance of an agent understanding the relationship between its sensory percepts of the environment, including proprioception ("the feel of the arm") and vision ("the sight of the object"). This relationship is the focus of Corbetta et al. (2014) [14], where they describe and evaluate three theories of how this relationship is established. These are vision first, proprioception first, or vision and proprioception together. Their experiments provide weak support for the proprioception-first theory, but their results demonstrate strengths and weaknesses of all three alternatives.

Thomas et al. (2015)[50] made close observations of the spontaneous self-touching behavior that can be observed in infants during their first six months. Analysis of this behavior provides evidence for the existence of two separately-developing neural pathways. One pathway corresponds to the task of reaching, the act of moving the hand to contact the target object. The second is for grasping, and is responsible for shaping the hand to gain successful control of the object.

The contributions of these and many other researchers provide valuable insights towards an answer to the question of the nature of the mapping that allows reaching and grasping to be possible. However, the different distinctions made by each investigator have led to a number of different theoretical answers. It is difficult to determine which differences imply theories are competing, and which differences are simply the result of different settings and methods, and remain compatible. By subjecting theories of a behavior of interest (in this case, learning from unguided experience to perform reaches, grasps, and other manipulation actions) to additional evaluations it is possible to identify the most important distinctions and aspects of each theory that can work together to explain the phenomenon. This evaluation can be carried out with a carefully defined and implemented computational model that is capable of performing the behavior of interest. We present our model for use in such an evaluation in Section 2.3.1 and Chapter 3.

2.2 Robot Learning of Manipulation Actions

Expert researchers in the fields of computer vision [16] and robot manipulation [46] have demonstrated that methods that provide detailed knowledge to a robotic agent can lead to impressive results. In particular, these models incorporate precise prior knowledge of the physical properties of the robot and that allows the agent to reconstruct the geometry of the environment and its manipulators. These methods contribute highly reliable and skilled actions, but they do little to illuminate how an intelligent action with minimal externally provided knowledge can acquire these skills. The behavior of newborn infants highlights the differences in starting capabilities, as the baby begins with seemingly random and poorly controlled motions but can purposefully interact with its environment after a few months of experience. To develop methods with more resemblance to the infant's circumstances, robotics researchers have employed a variety of approaches to allow the robot to rely on learned information rather than assumed prior knowledge.

2.2.1 Robotic Models

One of these approaches still involves a model of the robot and its environmental physics that could be used for traditional motion planning with forward and inverse kinematics, but has this model be learned with suitable accuracy instead of provided. Hersch et al. (2008) [17] propose a model where a body schema is learned for a humanoid robot. This method assumes that the robot is given the topology of the network of joints and segments of the body and that the 3D position of each end-effector can be tracked reliably. Given these assumptions, the model can be constructed as a tree-structured hierarchy of frames of reference. Sturm et al. (2008) [47] provide their learning agent with a pre-specified set of variables and a fully-connected Bayesian network model. The learning process uses visual images taken while the robot motor babbles with the arm. Their image processing method requires visual markers on the robot to satisfy an assumption that the 6D pose of each joint can be determined from these images. Bayesian inference completes the model by eliminating unnecessary links and learning probability distributions over the values of the provided variables. In our work, weaker assumptions are made about the variables and constraints included in the model and visual perception provides much weaker information to the agent.

2.2.2 Deep Reinforcement Learning Models

Advancements in deep neural networks and processing have been utilized by researchers using deep learning and very large scale datasets. These powerful learning networks can be trained to encode the information necessary for reliable actions. In [38], the high number of unlabeled training examples (50,000) gathered over 700 robot-hours allows the model to avoiding biasing and overfitting from a smaller number of human-labelled examples. They use a convolutional neural network (CNN) to predict grasp points and achieve reliable grasp results. Levine et al. (2016, 2018) [31, 30] use a CNN to predict from input monocular images if a motion will produce a successful grasp. This model can be used in real time to perform successful grasps through servoing that relies on the direction of the motion associated with the most confident prediction of success. This model has an even larger dataset of 800,000 grasp attempts observed over the course of two months with up to 14 manipulators at a time. While these learning models are powerful and encode salient concepts such as grasp points and hand-eye coordination, the encoding of the learned features in network weights causes the learned features to be less easily interpreted than those of our model. These models demonstrate successful grasp learning, but it is unclear if they could be used to model the learning of natural agents, given that this number of clearly observed examples may not be consistent with infant learning. Our agent learns from significantly less examples to meet our time and resource constraints.

2.2.3 Neural Models

Another approach focuses on hypotheses about the neural control of reaching and grasping, with model structures determined by these hypotheses. The constraints of these models are represented by neural networks that are trained from experience. Some of these models are motivated by empirical data from the literature on human infants, including the computational model ILGM, proposed by Oztop et al. (2004) [35]. This model consists of neural networks representing the probability distributions of joint angle velocities. The model is specific to grasp learning and uses an assumption that the knowledge of how to reach is provided. Their focus is on the learning of an open-loop controller that is likely to terminate in a successful grasp, and the performance is evaluated using a simulated robot arm and hand with a Palmar reflex.

Chinellato et al. (2011) [10] proposed another architecture consists of two radial basis function networks linking retinotopic information with eye movements and arm movements through a shared head/body representation. Their network's weights are trained through experience with an arm with two degrees of freedom in a simulated 2D environment. Experiments demonstrated the behaviors produced had appropriate qualitative properties.

The embodied computational model proposed by Savastano et al. (2013) [41] is implemented as a recurrent neural network and uses a simulated iCub robot for evaluation. They demonstrate multiple phases of reaching with different levels of skill, corresponding to pre-reaching, gross-reaching, and fine-reaching of human infants. These phases also exhibit qualitative matches with observed behaviors of children, with a diminished use of vision in the first two phases and a proximal to distal bias in the degrees of freedom of the arm that are selected for use. In order to cause the transition between stages, the experimenters manually add specific links and change certain parameters in the network. While this produces desirable results, it leaves a question about how and why these changes would take place during development.

Roncone et al. (2016) [40] also evaluate a model of peripersonal space with an iCub robot with artificial skin. This model incorporates proprioceptive, visual, and tactile modalities and allows the agent to perform reaching and avoidance tasks. They conduct experiments with double-touch (touching two parts of the self together to produce a tactile sensation in both) with and without vision, and with a stimulus from an independently moving object that relies on vision. The resulting learned representations are used by a velocity controller to produce avoidance behavior that maintains a margin of safety around the robot and reaching behaviors to move toward the greatest activation. While the stimuli and sensors are biologically inspired, the controller relies on detailed kinematics of the iCub robot and formulas for frame of reference changes, as well as the high-resolution tactile information from the 4,000 skin taxels that many robots do not have access to, including the Baxter robot used in our work.

Serino (2019) [43] proposes a model of peripersonal space that allows the extent and shape of the space to vary with experience and the external state of the environment. The sources of variation supported by the model include physical restraints on the body, extended reach with tool use, and social interactions. This model also incorporates visual, tactile, and proprioceptive percepts to create representations. Rather than experimenting with action learning, Serino focuses on the importance of the peripersonal space representation for the safety of the agent as well as its influence on recognizing ownership of parts of the self, self-consciousness, and social considerations.

The computational model of reach learning presented by Caligiore et al. (2014) [9] is based on reinforcement learning, equilibrium point control, and minimizing the speed of the hand at contact. This model is implemented with a simulated planar arm with two degrees of freedom, and generates predictions that are compared with longitudinal observations of reaching by infants between ages of 100 and 600 days recorded in Berthier et al. (2006) [6]. The comparison demonstrates qualitative similarities between the predictions and the experimental data in the evolution of performance variables over developmental time. The quality they focus on is the irregular, jerky trajectories of early reaching documented in Berthier et al. (2011) [5], which they attribute to sensor and process noise, corrective motions, and underdamped dynamics as suggested in Thelen et al. (1993) [49]. In our PPS Graph model, we suggest the irregularity of motion along graph paths is at least a partial cause of the irregular motions observed in infant behavior (and that the irregular motions are not the result of real-time detection and correction of errors in the trajectory, as this explanation would be inconsistent with Clifton et al. (1993) [11]). We accept that another component of the irregularity may be due to process noise and underdamped dynamics during motion along individual PPS Graph edges. While we show that it is possible to produce underdamped motions with our robot by adjusting the control mode and parameters, we have not yet evaluated these motions within our model. Our work also demonstrates that our model supports learning for a physical embodied robotic agent with a humanoid arm and corresponding higher number of degrees of freedom.

2.2.4 Sensorimotor Learning

Several more recent results have been produced by an approach that is more similar to ours as it focuses on sensorimotor learning without explicitly programmed-in skills, exploration guidance, or supervision through labeled training examples. Each work using this approach, including ours, requires the task or environment to be simplified via assumptions to allow the learner to make progress that can be observed and evaluated in a reasonable time given the current state of the art. While these methods speak most directly to their simplified problems, each contributes an important component of the solution to the problem of learning manipulation actions in the full complexity real-world setting and advances the discussion of how such learning may take place.

The developmental robotics results of Law et al. (2014a,b) [28, 29] are closely related to our work. In both cases, a graph-structured mapping is learned between proprioceptive and visual sensors, and therefore between the configuration space and a work space or image space observed with those sensors. Both our method and theirs also rely on an application of a form of intrinsic motivation that encourages learning from unusual events, which the agent can identify and learn to repeat reliably. In the case of our work, the theoretical model we present has no externally-imposed guidance or schedule for the order of learning tasks. In this case, the learning process would only be sequenced developmentally in terms of prerequisite actions, such as reaching being learned before grasping since a grasp includes a reach to approach the target. Our works differ significantly in their simplifications from this ideal model. Law et al. (2014a,b) [28, 29] provide an explicit schedule of “constraint release” times that are designed to follow the observed stages identified in developmental psychology literature, whereas we make assumptions about the agent’s intrinsic motivation to determine if the agent will advance to the learning stage for the next action or continue to practice the current action of interest with a new feature considered.

The Reachable Space Map proposed by Jamone et al. (2012, 2014) [21, 20] is defined in terms of gaze coordinates (head yaw and pitch, plus eye vergence to encode depth) during fixation. The control system moves the head and eyes to place the target object at the center of both camera images, and the required configuration maps to the point in space that must be reached to. Aspects of this relationship between retinal, gaze, and reach spaces were previously investigated by Hülse et al. (2010) [19]. The Reachable Space Map assigns a value $R \in [0, 1]$ reflecting the ease of reaching targets. $R = 0$ indicates that a target is unreachable, $R = 1$ indicates that the target may be reached with all joints having significant range in both directions before reaching the joint limits, and fractional values of R indicate that the object is reachable but only with some joints close to their limits. These intermediate R values are used as error values to motivate the use of other degrees of freedom determining the pose of the body (for example, the waist and the legs) to increase the reachability of target objects. The Reachable Space Map would be a valuable future addition to our framework, but each implementation of the PPS Graph [22, 23, 24] is learned at a developmentally earlier stage of knowledge before goal-directed reaching has a meaningful chance of success. The Explored PPS Graph [23, 24] is learned as the agent visits poses generated by non-goal-directed motor babbling. As each pose is visited, the agent records the set of joint angles and the image(s) of the arm, producing a discrete mapping between configuration and visual spaces at the points sampled with exploration. Only after the PPS Graph is completed is this information applied to the task of learning to reach reliably. Our work also differs in that we assume that all degrees of freedom other than those of the arm (including those to control the gaze and position of the head and body degrees of freedom our Baxter Research Robot does not have) are fixed.

Ugur et al. (2015) [51] demonstrate autonomous learning of behavioral primitives and object affordances, leading up to imitation learning of complex actions. However, they start with much stronger assumptions about peripersonal space than our model. Their model includes a 3D Euclidean spatial representation, and assumes that motions will be specified with the starting, midpoint, and endpoint coordinates of the hand in that 3D space. Our agent is only provided with the raw vector of joint angles sensed with proprioception and perceives coordinates of the hand in image space from the perspective of a fixed-position RGB-D camera. The PPS Graph represents a learned mapping between those spaces, and does not provide information about the 3D Euclidean space. The egocentric Reachable Space Map [20] could be a step toward an alternative 3D model of peripersonal space.

Other works rely on senses other than vision, such as Hoffman et al. (2017) [18]. They present a computational model that includes haptic and proprioceptive sensing, but not vision. Their model integrates empirical data from infant experiments and was evaluated on a physical iCub robot with artificial tactile-sensing skin. They model a process where infants are prompted to learn to reach to different parts of their bodies with buzzers placed on the skin. The results reported for the experiments with infants are used to derive constraints for the computational model. The authors present the work as only a partial success due to the disparities between the empirical data, conceptual framework, and robotic modeling. As a result of the limited integration of these aspects of the work, they describe their model as closer to traditional robot programming than the sensorimotor learning model they aspire to.

Luo et al. (2018) [32] create a three-phase model inspired by the first four months of human infant learning to reach and that is closely related to our work. The first phase corresponds to pre-reaching with motor babbling during an infant's first two months, and as suggested by Corbetta et al. (2014) [14] the agent first learns to guide the arm with proprioception. Their robot is given simulated proprioception that is represented with an autoencoder that was trained during the motor babbling motions. The second phase corresponds to learning to fixate visual attention on objects during the third month of infancy, and relies on haptic feedback from the environment. The third phase combines the movement of the arm with visual fixation on the target object to produce reaching. This model is also compatible with the finding by Clifton et al. (1993) [11] that reaching does not rely on current vision of the hand. The model is evaluated with a physical PKU-HR6.011 (a small humanoid robot) and two simulated robots. Their method is shown to produce very reliable actions in the PKU-HR6.011 evaluation, where the agent performs 99% successful reaching, 98% successful grasping, and 94% successful placing of one cuboid object onto another.

The results of Luo et al. (2018) [32] are highly impressive, especially when noting that they obtained a grasp action that is almost fully reliable, while the grasp action produced in our related method remains only semi-reliable. We believe that the higher performance is possible due to key differences from the assumptions of our method and in the information available to the agent. In particular, their agent is provided with knowledge of the mathematical structure of the robot's kinematics chains, and learns the appropriate tuning of coefficients for these matrix functions and the neural models that use them for forward and inverse kinematic reasoning. By contrast, our agent does not attempt to learn a general forward or inverse kinematics model, nor is it assumed to have an implicit understanding of the structures of the kinematics chains. Their agent is given the capability to extract the 3D position and orientation of the targets of its actions from its vision by exploiting knowledge that each target is a cuboid with color-coded faces. While our model does rely on distinctive colors to simplify image segmentation and processing, our agent is only allowed to produce less informative representations of the hand and other objects, such as binary image masks and depth ranges. In future work, we expect that the consideration of additional features or a reinforcement learning approach for late action learning could increase the reliability of our agent's grasp action without adopting stronger assumptions.

Developmental robotics with sensorimotor approaches inspired by infant learning remains an active field of research. Kumar et al. presented the Dev-PSchema model in 2018 [27]. This model performs bootstrap learning from simple action primitives with proprioception of the hand up to increasingly complex new actions based on intrinsic motivation principles. The selection of a task and target object are based on excitation rewards calculated from the similarity, novelty, and habituation of each choice. Adjusting the weight given to each excitation parameter allows variations in behavior similar to those between individual infants. Evaluations are performed with a physical and simulated iCub robot. Adjustments to the weights are also shown to produce preferences for novel or familiar objects (or novel objects with more or less properties in common with familiar objects), and equal weightings allow the agent to swap between actions quickly and combine novel action chains. We believe this approach may be applicable for our future work to allow the agent to produce its own learning sequence, removing the necessity of guidance from learning phases or specific actions to focus on. In very recent work in 2021, Kumar et al. presented continued development of the Dev-PSchema model [26]. The model is extended to be even more open-ended, with the agent creating general skills that were not observed or practiced in its learning experience in order to accomplish new user-defined goals. The sequencing of high-level actions to produce these solutions is expected to be a step toward learning tool use. These results also inspire opportunities for future work to evaluate our agent's learning and performance of high-level actions, including those that depend on the component actions learned in this dissertation.

2.3 Our Representation for Peripersonal Space

Our work with peripersonal space is inspired by prior works completed by members of our lab that considered learning problems in other spaces. In those previous works, a process was described that allows a mobile robot with uninterpreted sensors and end-effectors to construct a useful model of its own sensorimotor system [37]. The agent can use this type of model to distinguish mobile or quasi-static objects from the static background environment. Simple and human-understandable models of actions that can transform the states of objects can also be learned through this framework [33]. Mugan and Kuipers (2012) took the initial steps of investigating how an autonomously exploring agent could construct its own hierarchical models of actions for a manipulating robot [34].

2.3.1 The Peripersonal Space Graph

We present a graph-based model for representing peripersonal space. The nodes store configurations of visited poses as a set of arm joint angles and corresponding visual features for the hand in each pose, and the edges connect nodes if motion between their poses is feasible. As is typical with sensorimotor model approaches, in order to decrease the scope of the learning problem, our current model adds assumptions to simplify the environment, objects, and images. This allows a more restricted focus on specific questions which can be modeled in greater detail, and from these models insights and improvements can be seen. These results can be combined with future work that considers factors that are over-simplified at this time. Because our model is implemented and evaluated on a specific robot (a Baxter Research Robot in our works to date), the current model also reflects aspects of this robot's specific perceptual and motor systems that are not realistic for a human infant.

We have presented multiple implementations of the Peripersonal Space (PPS) Graph. In our work published in Humanoids 2016 [22], we manually construct a small Learning PPS Graph where the agent learns to reach, and show that this reach action performs well in the large Sampled PPS Graph which has configurations randomly sampled throughout the range of each joint. This is also where we first develop our algorithm for learning from unusual events based on intrinsic motivation [3]. When the results of a known action qualitatively differ from the observed typical results, the agent defines an action, and then identifies the features and feature values that allow it to plan to repeat the unusual event reliably. In this case of our Humanoids 2016 work [22], the agent observes that in rare cases random motions bump into blocks and change their position, and learns features to reliably reach and bump the blocks. At this time, we simplify the high level process with assumptions about the actions the agent will be intrinsically motivated to study and assume the results of searches over the set of possible features in some cases where the search has not yet been performed. In the ideal model we propose, the agent will be entirely responsible for feature generation and selection, and will autonomously direct its attention to actions in any order when they show the greatest expected intrinsic reward for being ready to learn.

The Explored PPS Graph is constructed and used in our works published in IROS 2018 and Frontiers in Neurorobotics 2019 [23, 24]. Instead of independently sampled configurations from the full configuration space, the poses for the nodes are generated using a motor babbling method that appears random, but with biases toward keeping the hand visible, as in von Hofsten (1982, 1984) and van der Meer (1995, 1997) [54, 55, 53, 52]. This bias can be enforced with rejection sampling, and generating a new next pose if the currently proposed pose is predicted to fall outside the view of the robot's camera. Due to the use of a new vision system, we assumed the agent began with a reliable reach action to focus on learning only the grasp action. Successful grasping requires two low-probability events to occur simultaneously. First, the direction of the hand's motion and the orientation of the hand must allow the object inside the hand, and second, the hand must be closed to create a firm connection with the object. These events occurring together by chance is too rare for the agent to observe a usable number of accidental grasps as an unusual event to repeat. We simulate the Palmar reflex, a reflex found in human infants that causes the hand to firmly grasp an object that is pressed into the palm [15], in order to remove one of the rare conditions for this event. The agent is then able to observe some accidental grasps, and learns to grasp semi-reliably. The agent's actions are also improved in this work with the introduction of local Jacobian estimates, to extend the PPS Graph from a discrete mapping between configuration space and image space to a locally continuous mapping centered at each observed pose. Modifications to trajectories using these estimates can interpolate between nodes to more precisely match the image-space position of the target, improving the reliability of the action. In our Frontiers 2019 work [24], we show that reaching and grasping can be learned in the same framework with two applications of the learning from unusual events procedure, and also that learning to reach does not require a smaller Learning PPS Graph (the number of bumps that can be observed in the larger randomly explored space covered by the Explored PPS Graph is sufficient).

In order to extend the set of learned actions to include placing, we define visually distinct qualitative locations that the object may be placed into. Our final demonstration of the agent's learned skill in this work is to combine a grasp and a place in sequence to perform a pick-and-place with a target object. In situations where the shape and orientation of the object allow it to be grasped again after the placement, this allows the agent to generate additional tasks to practice for itself. This gives the potential to learn from self-directed repetitive play, as in Schmidhuber (2011) [42].

We observe important qualitative consistencies between the trajectories for the move and reach actions learned by our model and the documented behaviors of human infants. In particular, the granularity of motions possible with the relatively sparse set of nodes and existence of only relatively short edges causes the trajectory to exhibit the jerky motions documented by von Hofsten (1991) [56], and offers support for a potential explanation for their cause. The agent's use of only stored visual percepts of the hand (and a current percept of the target object) while planning is consistent with the finding of Clifton et al. (1993) [11], where infants were observed to be equally capable of reaching for a lit object in a dark room, and thus were not reliant on current vision of the hand. Our model and its results are intended to be suitable for use as scaffolding for other agents that learn to perform smooth, skilled actions from the starting point of reliable but infant-like early action policies. A form of apprenticeship learning [1] could use demonstrations of our agent's actions to direct the attention of a reinforcement learner to a neighborhood of the action space that produces reliable actions, and should converge to skilled actions much more quickly.

2.3.2 Probabilistic Road Maps

As our PPS Graph is built with random motions and defines nodes with points in the robot's configuration space, it can be considered to be a type of Probabilistic Road Map (PRM), a method for robotic motion planning first introduced by Kavraki et al. (1996) [25]. The PRM method finds nodes near the desired start and goal positions and can efficiently find a path between them using a graph search, a process similar to our agent's procedure of choosing a reliable final node and travelling to it along graph edges. Reducing the problem of learning to move by using the graph to guide most of the motion between these start and end nodes allowed the learning phase to be shortened to approximately one minute, and the planning phase for each subsequent move could be completed in well under one second [25].

The PPS Graph is especially inspired by a variant of the PRM, the Visual Road Map (VRM) developed by Ramaiah et al. (2015) [39]. In the VRM, each node is annotated with images taken from different perspectives of the robot in the pose the node's configuration yields. In our Humanoids 2016 work [22], our agent's visual sensor used 3 webcams to gather a similar set of vectors of images to annotate each node. Both the VRM [39] and our Sampled PPS Graph [22] created the set of nodes by sampling points in the configuration space, though in our more recent work [23, 24], PPS Graph nodes are instead generated by a random walk, with small deltas added to each joint angle to arrive at the configuration for the next node. This approach has a stronger resemblance to the motor babbling and cyclical motions of young infants documented by Adolph and Berger (2005) [2]. In Ramaiah et al. (2015) [39], the robot is given expert knowledge in the form of a procedure for ruling out nodes and edges where collisions will occur, and significant focus is given to visualizations and analysis of the space based on the graph without these components. In our work, the agent is not given any pre-defined selection criteria for nodes or edges and must learn to use the information stored in the PPS Graph.

CHAPTER 3

Defining a Model of Peripersonal Space

3.1 What Spaces are We Modelling?

In order to discuss our model, we must first clarify the types of space we represent in this work, and which of many possible definitions we use. The scope of our work is also described by spaces that we do not focus on. Those spaces are important to other branches of literature, but differ from those in our model and our research question in significant ways.

Our agent learns to reach, grasp, and place objects with its end-effector. As a result, our main focus is on peripersonal space, the near surroundings of the agent including objects that are in arm's reach for the robot and that can be manipulated with the hand. Our Baxter Research Robot is stationary, and the range of its reach for defining what is within peripersonal space is measured from a single position. For simplicity, we also consider only a single arm that will be used for all motions and all action learning. The limited size and lack of locomotion differentiate the agent's peripersonal space environment from large scale and navigational spaces. In our experiments, the agent can observe its entire environment (albeit through limited sensors) at any time step, while the large scale spaces have at least some of the environment beyond the sensory horizon, requiring adjustments to the agent's position or gaze to observe additional regions.

As the set of reachable positions in space, the agent's peripersonal space will be primarily defined by the set of all possible points in configuration space of the robot's arm and hand. This is a continuous 8-dimensional space with values corresponding to the angular positions of 7 arm joints and the percent openness of the parallel gripper fingers. The Peripersonal Space Graph Model uses a sparse, discrete sampling of the set of feasible configurations to approximate the full continuous configuration space. This approximation is improved to a locally continuous model by calculating the local Jacobian in the neighborhood of each sampled point and using it to interpolate between those points.

Peripersonal space is also observed visually by the agent. We have shown that the number and type of cameras is not critical to our model, as the learning process succeeds in producing reliable actions with 2D RGB images from three webcams in [22] and with images from a single RGB-D camera in [23, 24]. However, our model does assume that the one or more cameras used will have fixed viewing angles and focuses, differentiating this space from those with gaze coordinates, such as in [20, 21] where the perspective can be changed with degrees of freedom in the neck, head, and eyes, as well as from visual space in studies using selective focus or foveated vision. While our approach is likely to support use of other sensors (perhaps laser range finders), we have used RGB or RGB-D cameras to create an image-space representation of the agent's peripersonal space as it would sense visually. It is important to note that any points, vectors, or binary masks defined with image-space coordinates will not correspond with the standard 3D coordinates of the robot's environment, and that the agent will not be provided with or compute a transformation from image-space to the 3D metrical workspace.

Our agent is best able to understand peripersonal space and perform actions in it by considering the relationship between its configuration space and vision, in particular focusing on the relationship between the configuration of the arm and the position and other features of the hand in the visual percepts taken at the same point in time. In order to preserve the relationship between an observed configuration and observed image, the agent will define a graph node using the set of joint angles and annotate the node with the image(s) taken at that time, which can be analyzed further to find the portion corresponding to the hand and the values of any features derived from it. By recording a large number of these paired observations in a set of nodes for the Peripersonal Space Graph, the agent learns a discrete mapping between the proprioceptive and visual descriptors of its peripersonal space. This model is expanded to a locally continuous approximation of the full space by computing a local Jacobian for each node, calculated as a best fit for the recorded changes in joint angles and resulting changes in hand position in image-space between the node and each of its neighbors.

The agent can also visually observe points in its workspace that are not associated with a known arm configuration. The agent uses a static background assumption to locate the arm in the image. This also allows for the observation of nonself foreground objects, which (in our experiments) are distinctively colored rectangular prism blocks within this work. (This simplifies the image processing required and leaves the determination of grasp points on objects with more complex shapes for future work, when the agent would be learning late grasping and become able to account for those features.) The agent can estimate the configuration space coordinates of an object by searching for a node where the stored image of the hand appears in an overlapping position with the object in image-space, and retrieving the joint angles stored for that node. The estimate can be improved using that node's local Jacobian to account for the displacement between the hand's center and the object's center in image space. Note that this process is distinct from inverse kinematics, as it uses a relationship between configuration space and image space rather than a 3D metrical space or six-dimensional pose. Further, the mapping uses the PPS Graph that the agent has built from experience, rather than the globally continuous mapping of an inverse kinematics function provided by prior knowledge.

A rectangular table is placed in the agent's environment centered in front of the robot. The table is centered with the agent, and the entire tabletop is within the agent's peripersonal space - though as our agent only uses its left hand, reaching to the far right side requires the arm to be more extended, which can be achieved only with a somewhat narrower range of joint angle combinations. For safety and to preserve the robot's relatively fragile arm, we prevent motions that would cause the arm to collide with the table, limiting the agent's peripersonal space. We also constrain the agent's motor babbling movements to points in space where the hand will be observable by the fixed camera. These constraints give the agent the same set of experiences as it would have if it did attempt these motions and chose not to preserve the positions it failed to reach without a collision (which would cause pain for a natural agent and would not be desirable to repeat) or that were not observable, and would not be able to appear desirable for manipulation actions. These safety measures are implemented using forward kinematics and the planned next configuration, though this formula and technique are not visible to the agent, and the checks are only performed as the agent learns to move the arm and explores the space, and do not limit its options as it plans other actions (for example, the agent will never need to attempt to plan a reach to an object that is beneath the surface of the table or that is not in a visible region of space).

The table's surface provides the agent with a workspace where it will interact with other objects in the foreground. As is the case with those smaller objects, the agent will not have a geometric model of the table or its coordinates or dimensions in three-dimensional space. The agent will primarily observe the table visually, and can describe the workspace as a region in image-space, or with features derived from this region, such as its center in image-space coordinates. As with objects, the agent can use the discrete mapping available from the nodes of the PPS Graph to approximate the configuration-space coordinates of a point in the workspace as the joint angles of the arm stored in a node where the stored image of the hand overlaps that point in image space.

As the table is present for the entire duration of the agent's learning, it is observed in each visual percept and is typically treated as part of the static background. However, qualitatively different locations in this workspace can be identified. In a real-world setting, these locations may have various properties or relationships with objects in the environment that would provide utility for objects being inside or outside these locations. In our experiments, the locations are manually defined regions on the tabletop with distinctive colors so that they may be easily identified by the agent using simple image processing methods to identify contiguous pixels with RGB values in a pre-programmed range around the expected color. These locations afford the same descriptions as the full tabletop and general points in the workspace - after being observed visually an estimate of their position in configuration space can be made from the PPS Graph. It should be noted that the locations used are relatively large, both to preserve the similarity to regions with a desired quality in natural scenarios, and also to fit the sensory and motor capabilities of the agent in early action learning, when the motors cannot produce and the sensors cannot verify exact poses. For an object to be considered in a location, it does not need to be in a specific position or orientation. Within the scope of this work, an object will be considered to be in a location if the object's visual percept (specifically its binary mask based on the RGB image) has any overlap with the percept of the location. In future works as the agent moves towards late actions and becomes more skilled, the locations will require more precise positions and orientations, and once locations are small enough, they could include positions on top of other objects to facilitate stacking.

3.2 Representing Peripersonal Space

The agent's peripersonal space is the space within its arm's reach, and where any object within that space can be interacted with using the robot's manipulators. Since the space is defined as the part of the environment within reach, it can be best described as a continuous set of six-dimensional poses $\mathbf{p}$ of the hand. However, like a human infant, our agent cannot measure $\mathbf{p}$. Instead, the agent observes the hand with its two sensory modalities, proprioception and vision.

For each feasible pose $\mathbf{p}$, there exists one or more configurations of the arm $\mathbf{q}$ that produce it. One such configuration is given by the *inverse kinematics* function

$$f^{-1}(\mathbf{p}) = \mathbf{q} \tag{3.1}$$

While this method is effective in contemporary robotics works for producing desired poses, f^{-1} is defined using expert knowledge of the structure of the arm, and would not be known to an infant. The function would also be prohibitively difficult for the infant to learn. Along with the mathematical complexity, the infant would not have a method for precisely measuring the geometry of the arm and the interactions of the joints, and would also be hindered by the changing size and strength of the arm and the changes in its vision. To retain similarity to the learning and behavior of infants that are limited in this way, our agent is not given and does not attempt to learn inverse kinematics.

However, the current configuration $\mathbf{q}$ of the arm is available to the agent at any time through proprioception. The arm's configuration is detected as a vector of the arm's seven joint angles

$$\mathbf{q} = \langle q^1, q^2 ..., q^7 \rangle \tag{3.2}$$

where q^j reports the current angular position of the jth joint in radians. This vector is always reported with a consistent ordering that matches the proximal to distal ordering of the joints[1],

[1] The vector contains shoulder (s), elbow (e), and wrist (w) joints, ordered s0, s1, e0, e1, w0, w1, w2. All of

but the agent does not use any assumption that this ordering describes the structure of the arm. Proprioception also allows the agent to sense the aperture of the gripper fingers a on a linear scale from 0 when closed as narrowly as the hardware allows to 100 when fully open to the maximum width. The arm has eight degrees of freedom, and the combination of $\{\mathbf{q}, a\}$ describes the full internal state of the arm at a given time.

Many contemporary robotics works also make use of *forward kinematics* to map from a potential set of joint angles to the pose it would produce with the equation

$$f(\mathbf{q}) = \mathbf{p}. \tag{3.3}$$

However, like its inverse f^{-1} for inverse kinematics, f is a highly complex function and relies on information that would not be available or consistent as a natural agent grows and learns to move. Our agent will also not use forward kinematics, and will instead use external senses to gather information about the arm's position and interactions in the environment. For our agent, the only external sense will be vision. Typical human infants would also be able to observe and describe the relationship between the hand and the environment with additional modalities, notably auditory and tactile senses, which can provide additional information about location and contacts with objects.

While the arm is sensed at a current arm configuration $\mathbf{q}$, the agent can make a concurrent visual observation, an image or set of images $I_{\mathbf{q}}$. The appearance of $I_{\mathbf{q}}$, including key features like the position of the hand in the image(s), will depend on the setting of the eight degrees of freedom in $\{\mathbf{q}, a\}$, as well as a set of environmental factors, E_t, present at time t, such as the presence of novel or moving objects in the background or the lighting conditions. This suggests a function g_1 to predict I_q,

$$I_{\mathbf{q}} = g_1(\mathbf{q}, a, E_t). \tag{3.4}$$

We can consider a function ϵ_1 that describes the effect of the factors in E_t on $I_{\mathbf{q}}$, which is dependent on the current time t, and can be assumed to be independent of the agent's actions. This suggests a new function g_2 that describes the agent's contribution to the appearance of $I_{\mathbf{q}}$ that it makes by setting $\mathbf{q}$ and a, such that

$$I_{\mathbf{q}} = g_2(\mathbf{q}, a) + \epsilon_1(t). \tag{3.5}$$

Our agent does not model the types of environmental factors in E_t or their consequences given

the Baxter Research Robot's joints are rotational, and this vector contains the four twist joints s0, e0, w0, and w2 and the three bend joints s1, e1, and w1. The full hardware specifications of the arm and joints can be found here: http://sdk.rethinkrobotics.com/wiki/Hardware_Specifications. Note that the vector of joints is one-indexed due for convenience when recording and looking up joint angles in MATLAB.

by $\epsilon_1(t)$. For the simplicity of our experiments and the infant-like learning agent, the agent uses a static background model, which assumes that as the time t varies, the factors in E_t produce a near-constant contribution $\overline{\epsilon}_1$, that is,

$$\forall t, \epsilon_1(t) \approx \overline{\epsilon}_1. \tag{3.6}$$

We have taken steps to ensure that the agent's use of a static background assumption can be reasonably justified. All of the visible background is as simple as possible, and visually distinct (i.e. differently colored) from the robot's arm and hand to eliminate the need for sophisticated image processing methods. As much as possible, the background was kept static and the lighting was kept consistent. Later, when the agent wishes to interact with non-self quasi-static foreground objects, these will also have to be observed visually, but we simplify the agent's first visual experiences with the assumption that no foreground objects are present until the agent begins to observe that it can bump them and then learns to reach for them. These early observations of I_q are the only ones that are stored for long term use in our current implementation, and none of them are complicated with foreground objects, which are only brought into the agent's field of view later.

These measures keep the variance of the environmental factors sufficiently low so that their impact can be treated as a constant $\overline{\epsilon}_1$. This also provides a convenient initialization for $\overline{\epsilon}_1$, as it can be set to the value of $\epsilon_1(t)$ observed for time $t = 0$, the first observation made. In this work our agent is not provided with and does not learn the functions g_2 and ϵ_1 to attribute parts of the appearance of I_q to its own actions or the environment, but the combination of equations 3.5 and 3.6 into

$$I_{\mathbf{q}} \approx g_2(\mathbf{q}, a) + \overline{\epsilon}_1 \tag{3.7}$$

is used to justify a more theoretical assumption that whenever the agent returns to the same settings of $\mathbf{q}$ and a, it will expect to see the same image(s) $I_{\mathbf{q}}$.

This assumption can still be used even in cases where there is a significant change to the environment, as the agent does not need to explicitly model states or transitions for parts of $I_{\mathbf{q}}$ that do not correspond with the dynamic self (the hand and arm). Since changes to E_t primarily change the appearance of the background of the image(s) and the agent has assumed this background is static, the same information and utility are available with an approximation of $I_{\mathbf{q}}$ that only depends on $\mathbf{q}$ and a, as long as the approximation is accurate in the region of $I_{\mathbf{q}}$ that corresponds with the self. Along with foreground object regions in later learning phases, these self-regions are the only parts of $I_{\mathbf{q}}$ that any features will be extracted from.

The agent will also use the assumption that changes to the gripper aperture a do not significantly change the appearance of I_q, even though any change that does exist will be localized in the region of the hand. This can be shown empirically when changing each degree of freedom, changes to a will change the size and center of the hand region less per unit than changes to any joint angle in

$\mathbf{q}$. While in practice this allows the contribution of a to be dropped, this suggests a new formula using a function g_3 for the effect of the arm configuration when separated from the unmodeled contributions of the gripper aperture and background environment,

$$I_{\mathbf{q}} = g_3(\mathbf{q}) + \epsilon_1(t) + \epsilon_2(a). \tag{3.8}$$

Our image processing algorithm for recording I_q for long-term storage requires having the grippers fully open with the the gripper aperture setting $a = 100$, so the agent will use this setting whenever it first moves to a new configuration $\mathbf{q}$. For this reason, any contribution of the grippers can be approximated as a constant $\overline{\epsilon}_2 = \epsilon_2(1)$. This allows new approximations for $I_{\mathbf{q}}$ that vary only with $\mathbf{q}$, justifying the use of $\mathbf{q}$ only as a subscript for the visual percepts:

$$I_{\mathbf{q}} \approx g_3(\mathbf{q}) + \overline{\epsilon}_1 + \overline{\epsilon}_2 \tag{3.9}$$

$$I_{\mathbf{q}} \approx g_3(\mathbf{q}) + \epsilon_1(0) + \epsilon_2(1). \tag{3.10}$$

Recalling that the terms $\overline{\epsilon}_1$ and $\overline{\epsilon}_2$ are either small or affect only irrelevant portions of $I_{\mathbf{q}}$, we can define a similarity between images that requires only their hand regions to be approximately equal (and thus yielding similar values of features extracted or calculated from those regions). When an arm configuration is repeated, the resulting image will be similar to the original image,

$$\mathbf{q}_1 = \mathbf{q}_2 \implies I_{\mathbf{q}_1} \sim I_{\mathbf{q}_2}, \tag{3.11}$$

as $g_3(\mathbf{q})$ itself will be similar to $I_{\mathbf{q}}$,

$$I_{\mathbf{q}} \sim g_3(\mathbf{q}), \tag{3.12}$$

using the same metric that compares only the region corresponding to the agent's self.

The next subsection describes how the agent will represent what it learns about g_3 as it builds a peripersonal space graph, a process that samples $(\mathbf{q_i}, I_{\mathbf{q}_i})$ pairs and stores them in graph nodes. Once the graph is completed, the agent will possess a discrete approximation of g_3. Since the agent does not attempt image generation, its model will be limited to the sampled values for the images themselves, but for simpler features such as the center of the hand region in an image, it will also have the ability to linearly extend the approximation to be locally continuous around each node.

3.2.1 The Peripersonal Space Graph

3.2.1.1 Overview

The Peripersonal Space (PPS) graph $\mathcal{P}$ is a collection of nodes and edges, representing a state of knowledge about the mapping g_3.[2] A node $n \in \mathcal{P}$ represents an observation $(\mathbf{q}, I_p)$. An edge $(n_i, n_j) = e_{ij} \in \mathcal{P}$ represents an *affordance* (i.e., an opportunity) for safe motion between $\mathbf{q}(n_i)$ and $\mathbf{q}(n_j)$. In our approach, the learning agent uses its unguided experience to define the PPS graph. We will first focus on the graph used for the latest iteration of our experiments, the Explored PPS Graph.

3.2.1.2 Nodes

Each node n_i of the PPS graph represents a state of the arm, defined in terms of its joint angles $\mathbf{q}_i$, so it represents a point in configuration space. The stored configuration $\mathbf{q}_i$ includes the angular position of each of the seven arm joint angles in a vector $\langle q_i^1, q_i^2, \ldots, q_i^7 \rangle$, as reported by proprioception when the node was recorded. Our current implementation does not use the gripper aperture a in the definition of a node, as it is required to be 100 during the process of building the PPS Graph for visual processing and the agent will assume changes to that degree of freedom do not significantly change the position or appearance of the hand. Keeping a out of the definition of the node will allow the agent to later freely set the gripper aperture degree of freedom to make grasps more reliable but still plan its motion through recorded nodes. The same will apply after a grasp for the return trajectory, when the aperture has changed but it will be convenient for the agent to consider itself to be revisiting the same nodes, so that it can track the target object's motion along with the motion of the hand to confirm the grasp.

The physical structure of the robot and its perceptual system also define a mapping from the pose of the hand to the visual percept, and the nodes will record observed configuration-image pairs, a discrete point in the mapping g_3. In order to do this, in addition to storing the configuration $\mathbf{q}_i$ that defines it, each node is also annotated with the perceptual image(s) of the hand and arm in the otherwise empty workspace. This involves storing the raw image(s) of $I_{\mathbf{q}_i}$ that were captured as the robot first visited the configuration $\mathbf{q}_i$, which for the Explored PPS Graph consist of two images, an RGB image with the standard range of 0-255 for the three color values of each pixel, and a depth image of disparity values from 0 to 80. Each image has dimensions of 640x480 pixels, and are taken at the same time and from the same perspective using a Kinect RGB-D camera mounted above the Baxter Research Robot's head screen.

Along with the raw images, the annotation includes processed images and measurements that

[2] Strictly speaking, a graph $\mathcal{P} = \langle \mathcal{N}, \mathcal{E} \rangle$ consists of two sets, one for nodes and one for edges. For notational simplicity, we will use $n_i \in \mathcal{P}$ and $e_{ij} \in \mathcal{P}$ as abbreviations for $n_i \in \mathcal{N}(\mathcal{P})$ and $e_{ij} \in \mathcal{E}(\mathcal{P})$.

provide a more abstract and usable representation of the agent's pose in the environment. While these will be primarily used by the agent instead of the raw images $I_{\mathbf{q}}$, they act as a lower-dimensionality manifold for the full image space, allowing the agent to still learn about the relationship between the configuration of the arm and its visual appearance. The relationships that will be the agent's focus will not be precisely g_3, but will be a simpler mapping to the manifold spaces and abstractions and should preserve the agents ability to learn the consequences of changes to the joint angles in terms of features that could also be observed from the raw image.

The first type of manifold used is a binary image, or mask, where a pixel is valued 1 if it is part of a region of interest, determined by whether the pixel at the same row and column in the raw image $I_{\mathbf{q}}$ is part of the segment for that object or region. The largest of these will be $R(I_{\mathbf{q}})$, corresponding to the agent's whole visible self, including every pixel of the image that is part of the robot. It is assumed that the agent will be able to identify each of these pixels, either with knowledge of the colors that appear on the robot's hand and arm, or by comparison with an image taken of the environment with no part of the self visible. This binary image will be referred to as the robot mask, and the robot mask that is the result from processing a specific stored image $I_{\mathbf{q}_i}$ will be denoted as $r_i = R(I_{\mathbf{q}_i})$. The robot mask generally indicates a large region that is a significant portion of the image, and it will primarily be useful for conservatively planning trajectories to achieve obstacle avoidance when additional objects are present and serve as obstacles, as was the case in [22]. Smaller masks focused on the region near the end effector will typically be more helpful for reaching and grasping tasks where more precision is required.

The agent will also segment the "palm" of the hand, the space between the grippers. While this region does not structurally resemble a human palm, we refer to it as the palm since it is the interior region of the hand most relevant for grasping, and is the site of the Palmar reflex[15]. To aid in this segmentation, a colored block of the same size as the space between the grippers while fully open is placed in the robot's hand, and is held for the duration of the creation of the PPS Graph. The RGB image in $I_{\mathbf{q}}$ is processed to find the largest segment of this distinctive color, and this segment becomes the binary image for the palm, or palm mask, $P(I_{\mathbf{q}})$. The palm mask for the image $I_{\mathbf{q}_i}$ stored in a specific node n_i may be referred to as $p_i = P(I_{\mathbf{q}_i})$.

A third binary image can be extracted from each $I_{\mathbf{q}_i}$, the hand mask, or $h_i = H(I_{\mathbf{q}_i})$. The hand mask includes the palm as well as the portions of the base of the hand and both gripper fingers that are in view with the current pose of the hand. The hand is all parts of the agent's self near the palm, so our image processing calculates this region from the intersection of the robot mask and a dilation of the palm mask with a 20 pixel radius,

$$h_i = r_i \cap \text{Dilation}(p_i, \text{Disk}_{20}) \tag{3.13}$$

The hand mask will be useful for confirming successful grasps, as the smaller palm mask may be fully covered by the grasped object instead of showing intersection or adjacency. The slightly larger size also makes the hand mask a candidate as a predictive feature for learning to reach and bump objects. Most bumps occur with the end-effector, and the smaller size allows the agent to better understand the region swept through along an edge, so this mask can also be used while reaching for obstacle avoidance considerations that will be less restrictive than those based on the full robot mask.

Since $I_\mathbf{q}$ only has a single perspective, the agent must also use the depth image to act reliably in a 3D environment. The agent will have the ability to find the disparity value at each pixel within a region of interest determined by a segment in the RGB image. $D(r_i)$, $D(h_i)$, and $D(p_i)$ will be the range of depths occupied by the robot, the hand, and the palm, respectively. Using a mask and the depth range for the depth mask, for example p_i and $D(p_i)$, together provides the agent with a conservative overestimate of the space occupied, as the agent can assume the entire range of depths is occupied across the entire mask, forming a type of hull. The combination of 2D coordinates (u, v) in the RGB image and the disparity value d from the depth image will give the agent a 3D coordinate system (u, v, d) for its visually-observed environment. While the image space defined by these coordinates is three-dimensional, it is important to note it is distinct from the standard 3D Euclidean space of the workspace environment. Along with the different axes, the units are different, as distances in image space would be measured in pixels by pixels by disparity.

The agent will derive additional features from these binary images and depth ranges, including the center of mass and vectors indicating direction. These will be introduced in section 3.4 as they become relevant to specific actions which will be learned in chapters 4, 5, 6, and 7.

Our current graph, the Explored Peripersonal Space Graph, contains 3,000 nodes. Even with this number, the nodes provide a sparse set of discrete observations of the mapping between configuration space and image space. Since the configuration space in which nodes are defined is 7-dimensional (one per joint), we can compare this number of nodes to an evenly spread 7-dimensional grid of nodes. Since $\sqrt[7]{3000} \approx 3.14$, the agent would only have about 3 settings of each joint angle in the grid. Even with a similarly sparse explored set of nodes, in this dissertation we show that this set of nodes is sufficient to facilitate learning a number of early manipulation actions.

3.2.1.3 Edges

An edge is defined by the nodes at its endpoints, such that $e_{i,j}$ is an edge connecting n_i and n_j. The PPS Graph is not complete, or fully connected, and an edge $e_{i,j}$ exists if and only if the motion from n_i to n_j is safe and feasible using linear interpolation between the configurations of the nodes. This feasibility is either determined by construction, as the agent has moved along the edge and observed it to be safe, or by assumption, with the agent treating edges as safe if they are shorter or similar in length to an edge known to be safe by construction. The length of an edge is the Euclidean distance between the configurations at its endpoint nodes, considered in joint space.

$$||e_{ij}|| = d(n_i, n_j) = ||\mathbf{q}_i - \mathbf{q}_j||_2 \tag{3.14}$$

Edges of the PPS Graph are directed, as some implementations of the graph and representations of learned actions may use a video percept of the motion, where the video for $e_{i,j}$ would be roughly the reverse of the video for $e_{j,i}$. More generally, due to the actuation of the robot and physics of the arm, the region the arm moves through is significantly different for some $e_{i,j}$ and $e_{j,i}$. While the edges are directed, the current Explored PPS Graph implementation is symmetric, so $e_{i,j}$ exists if and only if $e_{j,i}$ does.

$\mathcal{E}(\mathcal{P})$ will denote the set of all edges in a specific PPS Graph $\mathcal{P}$, and $\mathcal{E}$ will be used for a general set of edges, or as shorthand for $\mathcal{E}(\mathcal{P})$ when the graph the set belongs to is clear. The Explored PPS Graph has 111,717 edges, that is, $|\mathcal{E}(\text{Explored PPS Graph})| = 111,717$. While this is a seemingly large number of edges, note that it is only about 2% of the 4,448,500 edges a complete graph with 3,000 nodes would have.

3.2.1.4 Neighborhoods

Nodes of the PPS Graph are considered to be neighbors if an edge exists between them, allowing the agent to plan a move between them with path length one. The neighborhood $N(n_i)$ of a node n_i is the set of all neighbors of n_i,

$$N(n_i) \equiv \{n_j \mid \exists e_{i,j}\}. \tag{3.15}$$

The data recorded for neighborhoods in the PPS Graph can be used to extend the relationship between $\mathbf{q}$ and $I_{\mathbf{q}}$ (or processed images and features derived from $I_{\mathbf{q}}$) beyond the discrete observations stored in the nodes. This allows the agent to model peripersonal space with locally continuous estimates around each node. This technique will be used in section 3.4.2 and Chapter 4 as the agent learns to more precisely move the hand and predict its position, which will also be important in the later actions for reaching and grasping.

In this context of the locally continuous estimates, $N(n_i)$ will also be said to include the region in configuration space around q_i, where these estimates will not require error-prone extrapolations. These points will include all of the configurations stored by the neighbor nodes, and any less distant configurations, providing an alternative definition of a neighborhood around n_i in continuous configuration space,

$$N_{\mathbf{q}}(n_i) \equiv \{\mathbf{q} \neq \mathbf{q}_i \mid \{\exists n_j \in N(n_i) \mid ||\mathbf{q} - \mathbf{q}_i||_2 \leq ||\mathbf{q}_j - \mathbf{q}_i||_2\}\} \tag{3.16}$$

When the appropriate definition is clear from context, so both may be referred to as the neighborhood of n_i, and when the difference must be specified the presence or absence of the subscript **q** will indicate the type of neighborhood.

3.2.2 Alternative Graph Models

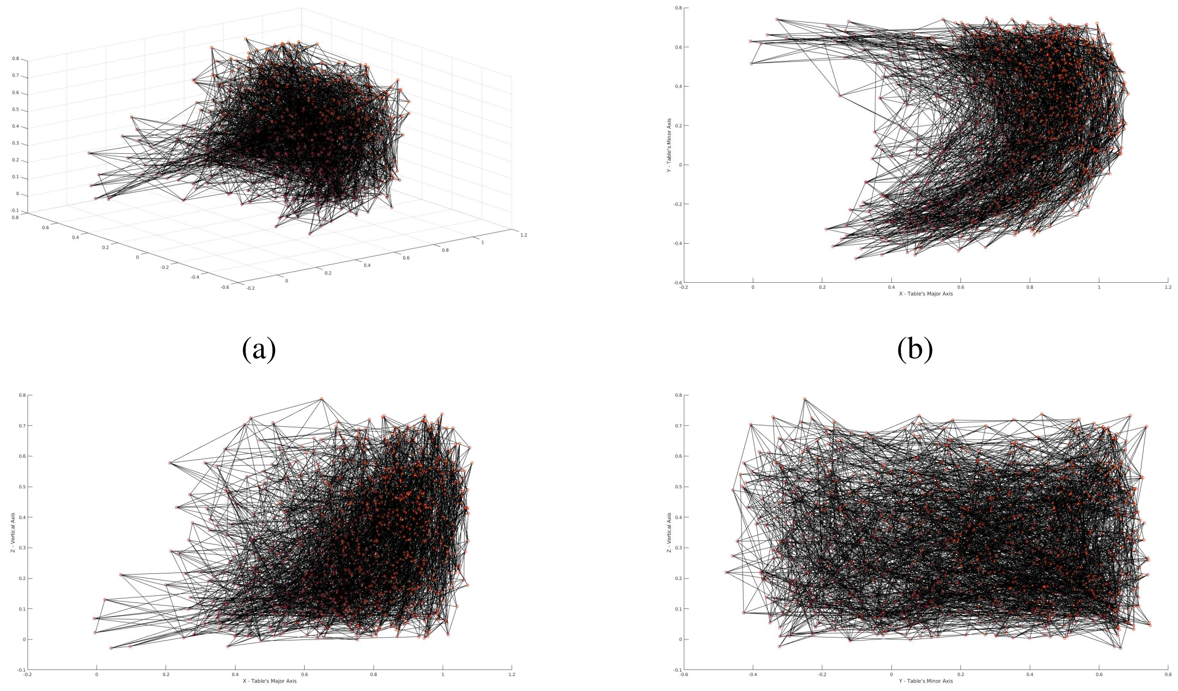
Throughout this work, Peripersonal Space Graphs were created with four different methodologies. Most of the experimental results reported in this dissertation were gathered while the agent built and used the Explored Peripersonal Space Graph that has been described up to this point in the section. Before this method, the agent constructed the Sampled PPS Graph and the Learning PPS Graph, and we have designed and constructed the Improved Explored PPS Graph, which was created using additional assumptions so that the images recorded would allow new features to be observed and existing feature values to be observed more accurately, which should improve action reliability in future work. These other methods are described in the remainder of the section. It should also be noted that more than four graphs were created during this work, with additional graphs created for some of the methodologies as needed to account for changed camera positions or new environments, such as a new location for the robot or a table with different dimensions being used for the surface in its workspace.

3.2.2.1 The Sampled PPS Graph

The first large-scale PPS Graph created for this dissertation work, It is shown in Figure 3.1 and fully discussed in [22], where the figure also appears. This graph had two primary differences from the Explored PPS Graph. First, at the time the vision system was based on three independently-placed webcams with fixed viewing angles, instead of a single RGB-D camera. As a result, $I_{\mathbf{q}}$ consisted of a vector of three two-dimensional images. The agent relied on color threshold assumptions to identify the self and the foreground objects, but used a different set of extracted features for this version of $I_{\mathbf{q}}$. While these images arguably provided more information and are simpler to use than RGB-D, the independently placed cameras had low biological feasibility. We changed to an RGB-D camera in our newer works as this system more closely resembles stereo vision of natural agents (particularly of typically developing human infants) and allows a more straightforward discussion of how such an agent might learn from its visual experiences.

Second, the configurations for each node of the graph was generated by sampling an angle q^j for each joint j, with the sample drawn uniformly from the entire range of the joint[3], rather than sampling a small delta to keep each joint near its previous angular position. For the safety

[3] We artificially reduced the range from the absolute limits set by the hardware to reduce strain on the robot that may occur with repeated motion near the extreme values.



(a) (b)

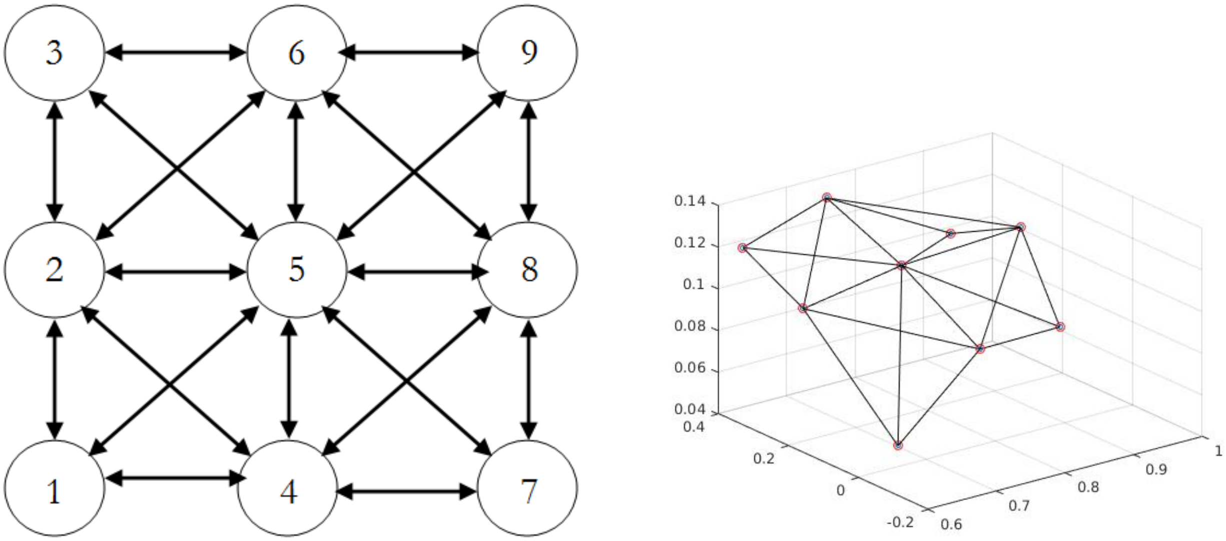
(c) (d)

Figure 3.1: (a) The Sampled PPS Graph with 1001 nodes and 6460 directed edges, with nodes plotted according to the 3D positon of the end-effector. *The agent only has access to a topological abstraction of this structure.* (b)-(d) 2D projections of the Sampled PPS Graph. Note that the random configuration space sampling procedure has produced a dense, well-covering structure, especially in the region most natural to sweep the arm through.

of our experimental robot while making these larger motions, we used an oracle (not accessible to the learning agent) to detect and reject collisions and other hazardous states. The oracle was implemented using the MoveIt! motion planner [48].

The original Sampled PPS Graph used in [22] contained 1001 nodes. Edges of the graph were created after all nodes were sampled, with edges added so that each node was connected to its five nearest neighbors in configuration space (the five connections that would result in the shortest edge lengths). Since the edges are directed, additional edges were added to ensure the graph was symmetrical even when the nearest neighbor relationship was not, for a total of 6400 edges.

The Sampled PPS Graph was used for evaluation of reaching and obstacle avoidance after the agent learned these actions using the Learning PPS Graph. The agent attempted to find a trajectory to reach a target object and avoid bumping an obstacle object for 50 trials. Completing the reach was weighted as the more important goal, and the agent reached the target in 45 trials (90%), avoided the obstacle in 37 trials (74%), and was successful at both goals in 34 trials (68%).



(a)

(b)

Figure 3.2: The Learning PPS Graph: (a) The topology of the abstract model used by the agent. (b) The metrical structure of the graph, illustrated by plotting the 3D end-effector positions at each node. This structure is not available to the agent.

3.2.2.2 The Learning PPS Graph

The Learning PPS Graph was the first graph we constructed and was small in size, serving as both a proof of concept and providing a highly constrained learning environment for the agent. It is discussed in more detail in the first publication related to this dissertation work [22]. Figure 3.2, which also appeared in [22], provides a visualization of the graph's topology (that was available to the agent), and a more standard metrical visualization in Euclidean space (that was not available to the agent).

The graph contained only nine nodes, each of which was recorded after the arm was manually positioned by the experimenter in a desirable pose. These poses were chosen to facilitate learning, with each a short distance above the table's surface to make interactions with blocks in the workspace more frequent. They were also arranged so that the end-effector positions were in a grid, and where the change in position was mostly achieved with straightforward adjustments of the proximal joints, so that the motion between the nodes would not be complicated by additional rotations and any resulting bumps would be easily anticipated by human observers. The vision system and features gathered for each node were the same as for the Sampled PPS Graph.

The graph contained 40 directed edges, connecting adjacent nodes in the 3x3 grid, including diagonal adjacency. The existence of these edges was defined for the agent. Due to the small number of edges, the agent could record a video of motion along the path and derive a binary mask for the entire region travelled through. In the Sampled PPS Graph, the much higher number of edges required this mask to be approximated by a convex hull of the binary masks for the nodes. The more accurate edge masks and simplified motions within the single-layer grid of poses were particularly helpful for obstacle avoidance. After the learning phases were complete, the agent evaluated its performance in the Learning PPS Graph before moving on to the evaluations with the Sampled PPS Graph. On 50 trials with the same goals, the agent reached the target in 45 trials (90%), avoided the obstacle in 43 trials (86%), and completed both goals successfully in 38 trials (76%).

Later, we demonstrated that the entire process of learning to reach and then to grasp could be carried out within the larger scale and more complex Explored PPS Graph (Section 3.2.1 and [24]), contradicting our intuition that this may be impossible or prohibitively inefficient without first learning at least some features in a small, expert-designed graph, and making future Learning PPS Graphs unnecessary.

3.2.2.3 Improvements to the Explored PPS Graph

Since the Explored PPS Graph that was built for [23] and also used in [24], we have generated ideas to improve the theory and implementation of the PPS Graph. One that we have begun to work with uses the observation that like the gripper aperture a, changes to the degree of freedom for the most distal joint w2 (represented as q^7 in the joint angle vector) also have very low impact per unit on the image $I_{\mathbf{q}}$, significantly lower than the effect per unit of change to any of the other joints. This can be observed by the experimenter and treated as an assumption for the agent, or learned autonomously by the agent as it calculates Jacobian estimates and identifies the low coefficients for w2.

In our previous work [23, 24], the agent identified that the orientation of the hand must be correctly aligned for the target object to reliably grasp it. The agent also learned to adjust only the orientation of the hand by rotating w2 rather than any other degree of freedom. While the hand orientation is dependent on all of the joint angles, this is justified by observing that it is best to adjust the orientation with only w2, as it has very little effect on the overall position of the hand. Altering any of the other joints to orient the hand would have a side effect of moving the hand, but the change in hand position for w2 is negligible. In [24], we assume that the agent performs the analysis that leads to the preference for rotating the hand with w2 just in time for the final iteration of attempting to make the grasp action more reliable by adding a new feature. The Improved Explored PPS Graph is differentiated from the Explored PPS Graph by using this assumption during the construction of the graph.

As the agent builds the Explored PPS Graph, w2 is treated like every other joint, so the agent only observes the hand with only one setting of w2 at each node n_i, q_i^7, and has little information about how the hand would be oriented if q^7 was changed. The agent was able to make a rough prediction of the value necessary for reliable grasping, using the q^7 setting from the nearest neighbor position with a successful grasp. For the Improved Explored PPS Graph, the agent instead collects six observations of the hand with values of q^7 spread evenly throughout the range. These additional images provide much more accurate information about how rotating the wrist with the hand in the same position will affect the orientation of the grippers. When these images are processed to compute binary images, the agent is also able to more finely segment the hand. While the images for the graph were collected, the robot's grippers were wrapped with distinctively colored materials, allowing them to be distinguished from each other and from the rest of the hand using additional assumptions about the segments that should be located and the range of colors their pixels should consist of. Instead of only a full hand mask h_i and palm mask p_i, the agent creates separate binary images for each gripper, the base of the hand, and the "palm" as the space between them. It is important that the agent differentiate between the grippers and separately track them so that it can attempt to avoid unnecessary half-turns of the hand between the penultimate and final nodes of a trajectory. Such a motion would leave the hand with a similar overall mask, but it could disrupt grasp attempts if the interaction with the target occurs during the rotation. We hypothesize that these changes will improve the reliability of the grasp by allowing the agent to better predict when the grippers will be correctly aligned, but we have not yet conducted an experiment to verify this hypothesis.

We also propose additional changes that have not yet been implemented for a constructed graph. If the need for visual simplicity while building the graph can be removed, along with the corresponding assumptions that no foreground objects will be present while the graph images are being recorded, the graph can made with more sophisticated techniques. Human infants show a tendency to direct their motion toward visible objects [4]. While these motions seldom result in contact with the object, this pre-reaching behavior during otherwise random, near-cyclic motions [2] biases the agent to visit poses closer to the object, with higher likelihood to be useful for later reach attempts for objects in the same region. With the capability of building the graph with objects present, the agent could have a similar bias towards an object instead of motor babbling with deltas drawn randomly from a uniform distribution.

Allowing for nodes to be recorded even with objects present could also allow the graph to continue growing in later learning stages. In our current implementation, the graph grows until a chosen number of nodes (3000 for the Explored PPS Graph) and then is fixed in that state for all future learning stages. Regardless of the presence of objects, it would be possible to identify relatively sparse regions of the graph, or ”holes” in configuration space where the distance from a point to the nearest node is significantly more than average. Action performance tends to drop in these sparse regions of the graph, as the agent has a lack of observations within the regions, and must extrapolate further from its past experiences to plan motions into unexplored spaces. The agent could choose to continue motor babbling in these regions, or use use any technique that adds nodes to increase the density of the graph where it is most sparse. Considering any objects present, and estimating their position in configuration space, the agent could also use such a technique to add nodes near these objects, making the assumption that the regions containing the foreground objects will be most likely to contain target objects in the future, so nodes in those regions will be especially valuable. Having the highest density in these action-relevant regions gives the agent information from more observations, and also provides more choices for motion along the graph edges, both of which should make the motions near the object more precise. Finally, the agent can determine if the success rates of its reach action are high enough to imply the graph is dense enough of average, and if it covers enough of the total peripersonal space. This may allow for the agent to initially collect a smaller amount of nodes, and add more until the learned reach action is reliable.

3.3 Locations in Workspace

To move the hand and to complete reaching and grasping tasks, the agent does not need to represent the specific appearance of the background of its environment. The assumption that each node stores the configuration for a feasible pose and that edges only exist when the motion between the endpoint nodes is safe allows the agent to avoid any permanent environmental obstacles as it moves the hand. We have also shown that it is possible, at least for some PPS Graph implementations, to temporarily remove nodes and edges where the agent would collide with foreground obstacles [22]. To reach and grasp objects, the agent only relies on current vision of the target, and does not use vision of the background or even current vision of the self.

This is also the case when the agent learns to ungrasp, releasing it from its control without regard for where it comes to rest. This action relies only on adjusting the gripper aperture a to sufficiently open the hand, which will not vary with the hand's position or relationship to anything else in the environment. The agent can use vision to track the object as it moves along with the hand to verify it is grasped, but it will also be aware that the grasp was initiated by the Palmar reflex and the restriction on how far the grippers could close. The knowledge that the object is currently grasped provides the affordance for an ungrasp. Vision of the object no longer moving with the hand can be used to verify that the ungrasp was successful.

However, once the agent is interested in placing, it must represent locations in the environment to determine if the object's position has particular properties, and if those are desirable. This extends to the case of the pick-and-place action, when the agent assesses where an object is currently located and decides to move it to a new location. Our agent will define locations as relatively large, qualitative regions, as many alternative representations for the location of an object would not be reasonable for our agent. The agent is not provided with the prior knowledge necessary to describe the object and its relationship with the table with geometric models. Without a model or more precise visual sensors, the agent is not capable of reliably finding the object's true six-dimensional pose in the workspace, and while this could be described in the agent's image space instead by using the object's binary mask and depth range, the agent's sensors are too prone to noise to work at this level of detail. More importantly, the agent's motors and motor skills would not be capable of recreating an exact position, even if it could be verified visually. The level of detail in our agent's representation is also not sufficient to determine an exact pose, and until the agent reaches a later stage where it is stacking, inserting, or building objects from smaller component objects, there is no additional utility to be gained by adding such details, so it will not be motivated to.

Instead of an exact pose, the agent will consider qualitative locations, which will better match the capabilities of the agent's sensors, motors, and knowledge. A qualitative region should be distinguishable and have utility based on properties of the environment. Some examples of locations could be to one side of an object, between objects, or inside a bin. In our experiments, we will define locations with colored patches on the table's surface, which simplify the agent's visual tasks, and are also flat on table so that the agent does not have to add obstacle avoidance like it would to place objects in relation to another object, or to avoid the walls of a bin while placing the object inside.

At a theoretical level, each location is a subset of the workspace surface $\mathcal{W}$, which is itself a specific subset of the agent's environment. $\mathcal{W}$ is a subset of the intersection of two spaces in the environment. The first is $\mathcal{R}$, the set of all reachable spaces, our agent's peripersonal space when reaching with its left arm. The second is $\mathcal{V}$, the set of all visible spaces from the agent's single, fixed perspective. All of the nodes of the PPS Graph exist at points in $\mathcal{R} \cap \mathcal{V}$. By construction, each PPS Graph node must be inside $\mathcal{R}$, since the node was recorded while that pose was held, and revisiting the stored configuration $\mathbf{q}$ will reach the same pose. All nodes are inside of $\mathcal{V}$ due to rejection sampling that prevented the agent from travelling to nodes outside of the field of view. In practice this was done to ensure the motion to the pose was safe, as it cannot be verified that the agent would not collide with an unseen object. There are theoretical explanations for this as well, as an unseen pose cannot have desirable image properties, so the agent would never choose to revisit the node. Storing the node would therefore have no utility, so the agent would be motivated to instead use its memory to store a node where the hand is visible. Whether by our rejection sampling approach or travelling to a node and then ruling it out for storage due to not being visible, the will be no nodes outside of $\mathcal{V}$. As a result, the peripersonal space graphs we construct will be sparse approximations of $\mathcal{R} \cap \mathcal{V}$, rather than the full reachable space $\mathcal{R}$. The workspace surface $\mathcal{W} \subset \mathcal{R} \cap \mathcal{V}$ is the subset of this space, given the specific setup of the environment, where objects may come to rest in a stable, quasi-static position, and where there are sufficiently nearby nodes so that an object at any point in $\mathcal{W}$ can still be reached and interacted with using the agent's end-effector. In our agent's

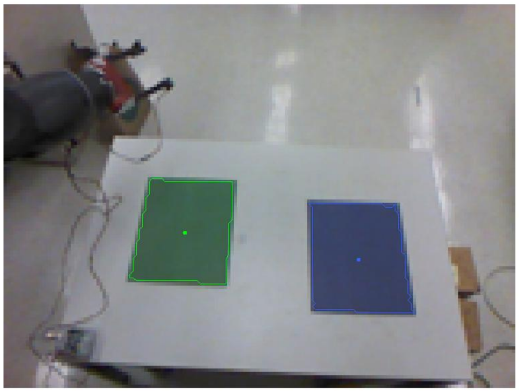


Figure 3.3: The location L_1 defined by the green patch on the table and the location L_2 defined by the blue patch on the table. Each location is shown with the boundary of its mask and center highlighted. The location L_3 corresponds to the rest of the table's surface, the grey region surrounding and between the colored patches.

environment, $\mathcal{W}$ includes the surface of the table in front of the robot and the space just above it that objects placed on the table will occupy. Note that it is not the case that nodes cover the full extent of this space, notably, by rejection sampling for safety there are no nodes where the hand would intersect with or touch the table. Also note that $\mathcal{W}$ will not include the floor of the environment where objects could also be in visible, quasi-static locations, as they would be too far from any PPS Graph nodes to remain reachable, and would hold these positions until an experimenter intervenes.

In practice, we define three qualitative locations within the tabletop workspace surface $\mathcal{W}$. The locations will be identified visually by the agent, and we place colored patches on the table to assist with this, as seen in Figure 3.3. $L_1 \subset \mathcal{W}$ will be defined by a binary mask for a distinctive green patch in the RGB image of I, and $L_2 \subset \mathcal{W}$ will be defined by a similar binary mask for a distinctive blue patch. Since the agent's perspective is fixed, these masks will also be constant up to occlusion by the agent's arm. When the location is not occluded, the agent will record an authoritative binary image for the location, and this will be be assumed to be the location's position in image space even if occlusions or environmental factors like lighting change the identified mask.

Objects will be considered to be in L_1 or L_2 if their binary image mask has a nonempty intersection with the location's. We also considered requiring the object's mask to be fully inside the location as a subset of its mask, but this would create false negatives when the top of an object that only touches the table within the colored patch could extend outside the mask for the patch. The agent would not be able to identify such cases with its typical representations of depth (as either a mean value or a range over the entire mask). We intended these locations to be relatively lenient and suitable for early placing actions, so we chose not to require the agent to add detail to its object representations and complexity to its reasoning. As a result, we chose this more inclusive definition for being "in" a location. This definition also does not require the object to be upright, or in any other particular orientation. Finally, to allow for more interesting pick and place actions, L_1 and L_2 will not be adjacent, and will be sufficiently far apart such that no foreground objects the agent works with could ever be in both locations.

The third location L_3 will correspond to the rest of the table's surface, $\mathcal{W} - (L_1 \cup L_2)$. This location will primarily be used to describe failures to place an object into L_1 or L_2, though it could also describe the initial position of an object to move. The agent could also learn a rule to place into L_3, but the rule would be more complex due to the "holes" in this location where L_1 and L_2 are, and the agent would have to avoid placing into them. This would also be a less natural task, of moving an object out of a location but to anywhere else, without a specific relationship to the old position. To avoid cases where an object could be in both L_3 and one of the other locations, placement in L_3 will not use the same inclusive definition. Instead, an object is in L_3 if and only if it is not in L_1 and it is not in L_2.

3.4 Features used for Action Learning

3.4.1 Feature Generation Assumptions

In this section, we will discuss the additional features and methods that will be used as the agent learns to move the arm, reach to a target object, grasp a target object, ungrasp a grasped object, and perform ungrasps so that an object is placed into a desired location. For each feature, we will state if it is assumed innate and provided to the agent, if the agent is provided with a measure and learns a threshold or desirable values, or if the feature is selected from a set of possible features as the best predictor of a reliable action.

In an ideal version of the model in this work, the agent would be fully unguided, and would generate all features independently and continue to use those that were useful in predicting and planning successful actions, rejecting others. The ideal agent would also be autonomous, and would introduce new features as it was intrinsically motivated to do so, without the need for learning phases where the experimenters signal to add a feature or to switch to learning a new action.

This ideal model is not achievable within the scope of this dissertation, as it would generate a larger number of features and with less guidance would take a longer amount of time to learn the necessary information for each action. To keep the learning methods feasible and guarantee results within the time frame of this work, it has been necessary to generate fewer features to choose from, and sometimes to partially or fully provide the measure the agent should use. One group of assumptions that could be retained with the ideal model is those that deal with processing the images and identifying objects and relevant properties. We consider this to be an interesting computer vision problem, but separate from the focus of our work, and also a problem where many existing solutions would be satisfactory for our needs.

The following chapters will describe the learning methods in more detail, and provide evidence that this work does contribute to the goal of working towards learning the actions from fully unguided experience. In future work, the agent could become closer to the theoretical ideal by generating larger sets of features to choose from for each phase, including identifying unusual events, learning to reliably repeat them with actions, and learning to verify that the action was successful.

3.4.2 Features for Moving the Arm

Learning to move the arm is primarily reliant on the assumed capabilities of the agent to use its sensors. The agent is assumed to be able to use proprioception to record the set of joint angles in a consistent order in the configuration vector $\mathbf{q}$. The agent is also assumed to be able to record the RGB and depth images of $I(\mathbf{q})$ concurrently with holding the configuration $\mathbf{q}$, and to perfectly recall these images for extracting features. Within these images, the agent is also assumed to be capable of identifying the regions corresponding to itself and important portions of itself, like the hand and palm. The agent is capable of creating binary image masks and depth ranges for these regions of interest, and can describe them with centers of mass (as derived from the pixels in the binary mask and the range of depths at the same positions - not a true center of the object, which along with its accurate geometry will remain unknown).

Before the agent can move the arm to a configuration $\mathbf{q}$ to motor babble and build the PPS Graph, it must be able to move all of the joints to set points. We demonstrate with experimental results that parameters for angular velocity-based control can be found that bring a single joint to a desired set point with either underdamped, overdamped, or critically damped motion. We then assume that the agent is capable of combining the controllers for the move joint actions together in parallel to form a reliable move arm action where all joints are brought to their set points in the same motion.

After the agent constructs the PPS Graph, the agent can calculate a local Jacobian estimate around each node n_i, using its neighborhood $N(n_i)$. This involves domain-general math to measure the change in configuration space and image space coordinates between the node and each neighbor, and then solving for the best fit. We assume the agent is capable of performing these operations, and is also capable of inverting the solution Matrix to determine the change in configuration needed to move to a position in the image that is nearby, but not represented by any node. Since the math to relate the seven dimensional configuration to the 3D image coordinates - even when done as a locally linear approximation - seems unreasonably complex for an agent similar to a human infant, it could be assumed to be performed by a subconscious, innate process of the brain. We also consider some simplifications to the relationship in Chapter 4, if the agent requires simpler math until a later stage.

3.4.3 Features for Reaching

Reaching also relies on the assumption that the agent can extract features of the hand and palm from the stored images, and adds the assumption that the agent can identify a target foreground object in the current visual percept to begin planning a reach. We currently assume that the knowledge objects exist and will be a useful abstraction from pixel-level data, but an experiment could show that it is possible for the agent to initially treat these as connected components of distinct pixels and define objects and their behavior upon seeing correlated changes to the pixels when the object is bumped.

The center of the hand in an image stored in a node will be given by $c_h = (u_h, v_h, d_h)$, where u_h and v_h are the coordinates of the center of mass of the binary mask of the hand based on the RGB image, and d_h is the mean disparity value over the pixels in the depth image specified by the binary mask. The same can be derived for the center of the palm $c_p = (u_p, v_p, d_p)$. It is important to remember that these are always derived from stored images, and will not use current percepts of the hand. A target object t also has a center $c_t = (u_t, v_t, d_t)$, but this will be taken from the current percept before planning a move in the presence of the object, or a reach for that object.

The agent is also assumed to be capable of taking the intersection and union of binary images, whether they are masks for the same object at different times or masks for different objects, such as the stored mask for the palm and the current mask of the target object. The agent is also innately capable of measuring the size of a binary image mask as the number of pixels in the region. The agent can similarly find the intersection, union, and size of depth ranges.

The agent can identify a short list of candidate final nodes to move to for a reach by looking for nonzero size intersections between the stored hand or palm mask and depth range and the current mask and depth range of the object before beginning the reach. The agent will learn that using the palm mask is more reliable than the larger hand mask.

The agent is provided with the ability to use the intersection over union ratio of two masks. When used on the same object's mask before and after completing a move trajectory, the agent will learn a threshold for this measure that identifies bumps as an unusual event. This threshold can later be used when attempting the reach action to check for success in causing a bump.

The agent's reaches will not be sufficiently reliable when choosing from the set of candidate final nodes randomly, so the agent will be provided with four features to attempt to rank the candidates and choose the most reliable option. These features will measure the difference between the center of the palm and the center of the target, either in one coordinate or in Euclidean distance. These features will be $f_u = |u_t - u_p|$, $f_v = |v_t - v_p|$, $f_d = |d_t - d_p|$, and $f_c = ||c_t - c_p||$. The agent will examine its past experience to learn that f_c is the best predictor, and to learn that minimizing this measure allows bumps to be produced most reliably.

3.4.4 Features for Grasping

In Chapter 6, the agent learns to grasp semi-reliably, and even reaching that 50% success rate involves a large number of features due to the high complexity of the action. A grasp approach is a specific reach trajectory, so all of the previously discussed features will still be relevant.

In addition to moving to the right position, the orientation of the hand is also important for grasping. The agent will be instructed to extract a set of five vectors from the stored images of the hand and current image of the target object. Two will describe the orientation of the hand in terms of the direction the grippers are pointing at the final and penultimate nodes of the trajectory ($\vec{g}_f$ and $\vec{g}_p$, respectively). The other three vectors will describe the direction of motion between the image-space centers of relevant regions. $\vec{m}_{pf}$ gives the direction from the penultimate node to the final node, $\vec{m}_{pt}$ gives the direction of motion from the penultimate node to the target, and $\vec{m}_{ft}$ gives the direction from the final node to the target. Formulas for the five vectors are given in equation 6.1, where they are used in Section 6.3. The current percept of the target object while planning also allows extraction of a vector parallel to the object's major axis, $\vec{o}$. The agent will be instructed to consider which cosine similarity between each pair of vectors has been observed in successful grasps. The agent will conclude that for the most reliable grasp trajectories, the relationship between $\vec{o}$ and any other vector should be approximately 0 so that the approach and orientation of the hand are near perpendicular to the object's major axis, and that all other cosine similarities should be 1, so that the other five vectors - two describing hand orientation and three describing direction of motion before local Jacobian adjustments - will be aligned. In practice, the agent will attempt to minimize the cosine similarity of any vector with $\vec{o}$ and use well positioned and oriented nodes and local Jacobian adjustments to maximize the alignment of the other vectors.

At this time, we also provide the rule for an agent to verify a grasp has occurred. Intuitively, when a grasp is successful, the grasped object moves along with the hand, and should be seen at a similar position in the image at each node visited in the trajectory after the grasp. The agent is instructed to verify grasps by checking if for each node in the trajectory, the stored hand mask has a nonzero intersection with the current object mask. This rule is simpler than those we have had the agent learn, but it is also possible to separate into grasp and nongrasp clusters using other metrics. A pair of metrics that was learned in another implementation was to compare the cosine similarity and distance travelled by the object and hand for each edge in the trajectory.

Grasping will also require the agent to correctly use the gripper aperture a, a degree of freedom that did not have a significant role for moving or reaching. The agent is assumed able to recognize that the Palmar reflex has been activated when a suddenly decreases without a conscious command. An experiment will be conducted where grasps are attempted with varied initial values of a, from which the agent will learn that $a = 100$ will be most reliable for grasping, and equally reliable for reaching.

Finally, the agent must also orient the grippers so that they will close around the target object. While this does not require a new feature, it does use special attention to the w2 joint that is best suited for rotating the hand at the wrist. In order to freely set w2, it must no longer be considered to be part of the definition of the nodes being travelled to, or important to setting the arm to the nearby configurations displaced from the node's according to its inverse local Jacobian estimate. When using the Explored PPS Graph, the agent can use the local Jacobian information and observe that the image space coordinates of the center of the hand do not change significantly as the angle of w2 changes. This justifies the use of w2 for other purposes, such as controlling the orientation of the hand and grippers. Alternatively, in the Improved Explored PPS Graph, the agent simply assumes that w2, like a, is not part of the definition of the nodes. With this earlier justification, the agent can add to the graph observations with multiple w2 settings at each node, and better understands how rotating it will affect the pose of the hand and grippers.

3.4.5 Features for Placing

The place action requires a different set of prerequisites - primarily that an object is currently grasped - and is performed with a specific move and then an ungrasp, but not a reach or grasp trajectory. Despite these differences, few new features need to be considered for placing.

In order to ungrasp, the agent explores changes to each degree of freedom and learns that increasing the gripper aperture a is the only change that reliably ends a grasp. While the agent is instructed to do so, it performs an exhaustive search of all possible pairs of degrees of freedom and directions of change, and would with sufficient time find the same solution in any manner that it was motivated to experiment with options, or if it did so randomly. The ungrasp is verified with the same test – whether the object moves with the hand – that verifies grasps, but with the opposite success condition (the object should no longer follow the hand).

As discussed in section 3.3, the locations where objects can be placed will be defined by binary image masks. The depth image values of these pixels can also be used to give the location a depth range, and the mean of these depths can be used to give the location an image-space center, $c_{L_i} = (u_{L_i}, v_{L_i}, d_{L_i})$. While checking for an intersection of the location's mask and an object's mask is enough to verify that a placement has occurred, the agent will be provided with several potential measures, and learns through experimentation that a short euclidean distance to the location's center best predicts that a node will be reliable for ungrasping an object and placing into that location.

3.5 Conclusion

Within this chapter, we have discussed peripersonal space and the representation for it used by our embodied robotic agent. In particular, we described the agent's visual and proprioceptive senses, and the agent's ability to record simultaneous observations in both modalities. We describe how the agent will create a set of peripersonal space graph nodes as it explores its space by motor babbling. In this case these motions are performed by adjusting the seven joints of the arm according to small angular deltas randomly sampled from a uniform distribution, and at each configuration visited, a node can be created to store the configuration held and the resulting visual percept. Using the paired data in all of the nodes together, the agent will possess a sparse, discrete mapping between configuration space and image space, which can be extended to a locally continuous mapping using each node's relationship to its neighbors.

In addition to representing the peripersonal space, a constructed PPS graph allows the agent to move the hand, with known affordances for safe motion between nodes denoted by edges. Using the mapping between joint angles and images stored in the nodes, the agent can learn to move the hand and predict its position and appearance, which will be examined further in Chapter 4. Extracting the additional features discussed in this chapter will allow the agent to learn to plan movement trajectories with well chosen final nodes (or motions) for successful reach and grasp actions in Chapters 5 and 6, respectively. We have also provided a definition for relatively large, qualitative locations as a subset of the workspace surface in the agent's environment. In Chapter 7, the PPS graph will also facilitate learning to move the hand to a suitable pose before ungrasping an object to place it in a desirable location. Chapter 7 will also show experimental results to support that the learned actions can be combined in sequence to allow the agent to pick and place objects.

Along with the demonstration that the PPS Graph and this set of actions could be learned from experience with minimal guidance, the representation described in this chapter is a contribution to three areas. First, the PPS Graph is suitable for suggesting and testing theories in developmental psychology. Second, the PPS Graph provides an efficient way to gather successful but unoptimal examples of the actions learned using it to plan motions, making it suitable as a starting point for reinforcement learning or scaffolding for other methods that are limited by sample complexity. Third, the PPS Graph allows actions up to pick and place to be learned with minimal assumptions, which is useful for symbolic action planners and motion planners that wish to justify the use of pick and place as an action primitive in the agent's prior knowledge. With the context of the learning processes from Chapters 4-7, each of these contributions will be discussed in further detail in the dissertation's conclusion in Chapter 8.

CHAPTER 4

Learning to Move the Arm

4.1 Introduction

In this chapter, we discuss the first stages of the agent's learning up to gaining the ability to move the all of the arm joints in parallel to bring the arm to a desired point in configuration space or the hand approximately to a desired point in image space. Beginning with a modest set of assumed prior knowledge and capabilities, the agent first learns to move individual joints in section 4.2. An experiment performing motions while using a range of parameters for a proportional controller for the angular velocity reveals a value that allows near critically-damped motion to a set point defined as an angular position. Once this has been demonstrated, we assume that the agent is capable of using these abilities in parallel to command all of the joints of the arm to set points at once, allowing it to bring each joint to a desired angle in a single motion.

The key result of this chapter is the construction of the Explored Peripersonal Space Graph, an instance of the theoretical PPS Graph representation described in Chapter 3. The Explored PPS Graph has all of the properties of the general PPS Graph model, and is built with a particular set of methods described in section 4.3. Once completed, the agent will have stored experiences of a large number of poses it has moved to while motor babbling, and the sum of this information acts as a sparse, discrete model of peripersonal space and a mapping between the configuration space of the arm and the appearance of the hand in image space at those recorded points. While the agent was able to command the arm to any set point without the graph, having the graph allows the agent to make motions with predictable results, as moving back to a previous configuration will produce a pose with a similar appearance and hand location in image space. The edges of the graph also provide an improvement over motion with the velocity controller to an arbitrary set point. In particular, the information stored in the edges can be used to perform motion planning of early - potentially inefficient and jerky, but safe - trajectories, while interpolating a straight path in configuration space between the from and to points does not rule out collisions with the environment or the self.

After the graph is completed, the agent can address one of the initial model's notable shortcomings - if only the nodes are considered as potential destinations, the sparseness of the graph makes some poses impossible to achieve. While motion to the nearest configuration stored in a node may be sufficient for some actions, such as semi-reliable reaching to bump objects, this method will lack the precision necessary for making reaching fully reliable, or for any significant level of success with the more difficult grasp action. In order to address this, the agent can extend the graph's discrete model to a locally continuous model. Observing the changes in configuration and image space between a node and each of its neighbors allows the agent to compute a local Jacobian estimate of the affects of small changes to the joint angles from those stored in that node on the palm's location in image space. While the local Jacobian allows the agent to predict the image space coordinates for any configuration near a node, the inverse of the local Jacobian will allow the agent to estimate the configuration necessary to place the palm at a desired position in the image. This will have applications that increase the reliability of the reach and grasp actions in Chapters 4 and 5, and allows for additional considerations while learning to move that we discuss at the end of section 4.4.

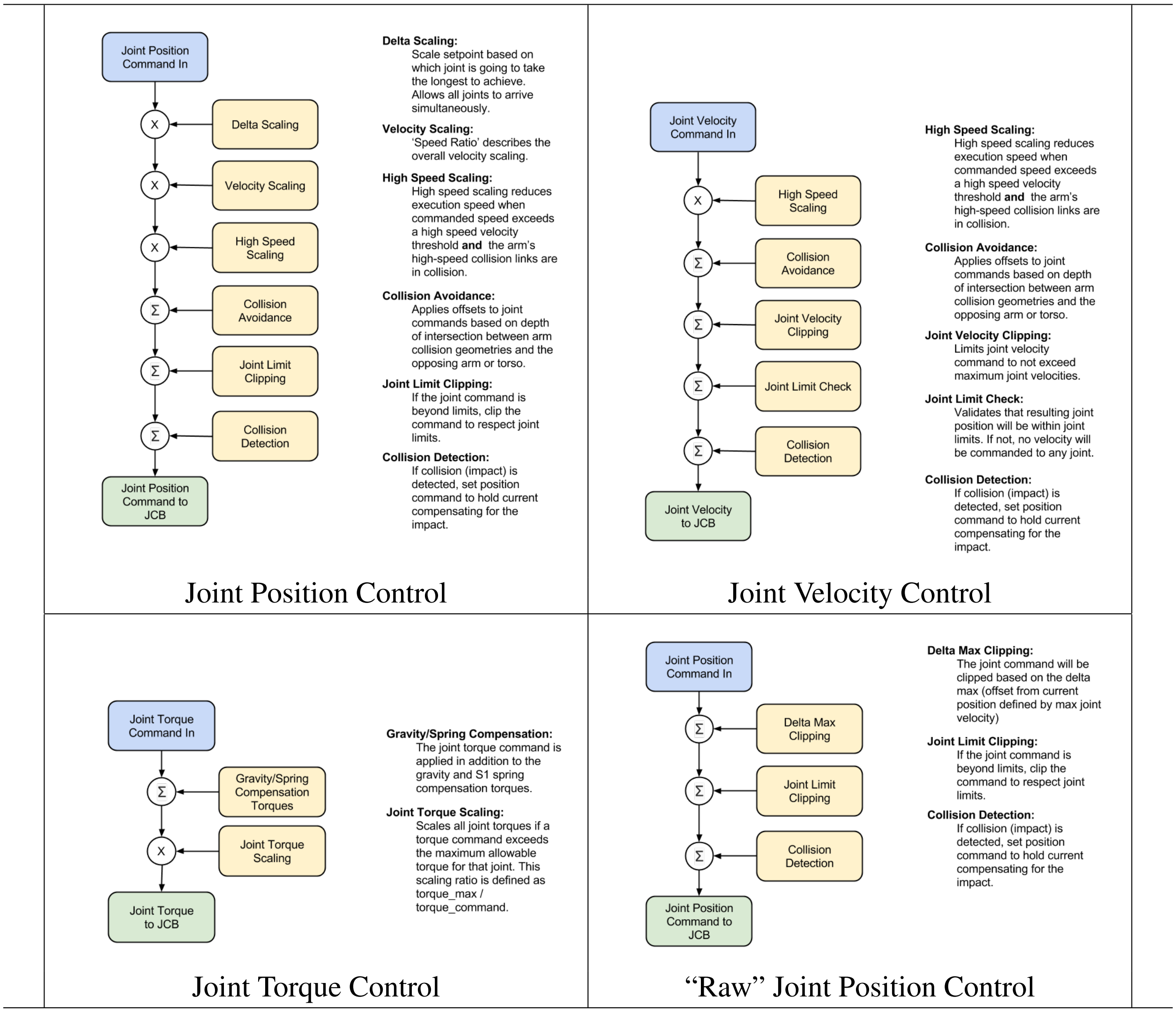
4.2 Learning to Move Joints to Set Points

4.2.1 Selecting a Control Mode

The Baxter Research Robot interface provides four built-in control modes, Joint Position Control, Joint Velocity Control, Joint Torque Control, and "Raw" Joint Position Control. It is important to note that all of these control modes give commands to the joints in terms of angular positions, velocities, or torques, and that no control mode allows specifying a desired 3D or 6D pose in the environment or makes use of inverse kinematics. Figure 4.1 shows flowcharts of the steps performed by each control mode to transform the issued motor command into the motor command executed by the joint control boards in the robot's hardware.

4.2.1.1 Justification and Limitations of Use of Joint Position Control Mode

We initially chose to use the Joint Position Control mode as it is the default and most widely used setting for the Baxter Research Robot. We continued to use this mode for the experiments in this work except for comparisons between modes in this chapter. As the primary control mode, it has the most support from the developers and the community, and has also had the most attention to ensure that the motions produced will be consistent and safe. In addition to this prevalence, Joint Position Control has advantages for efficiently performing the learning experiments we conducted. In particular, this control mode provides additional processing steps that act as layers of abstraction between the high level motion planning decisions that we demonstrate our agent has learned in this work (such as where to move to reliably reach a target object) and the low level control decisions



Delta Scaling: Scale setpoint based on which joint is going to take the longest to achieve. Allows all joints to arrive simultaneously.

Velocity Scaling: 'Speed Ratio' describes the overall velocity scaling.

High Speed Scaling: High speed scaling reduces execution speed when commanded speed exceeds a high speed velocity threshold **and** the arm's high-speed collision links are in collision.

Collision Avoidance: Applies offsets to joint commands based on depth of intersection between arm collision geometries and the opposing arm or torso.

Joint Limit Clipping: If the joint command is beyond limits, clip the command to respect joint limits.

Collision Detection: If collision (impact) is detected, set position command to hold current compensating for the impact.

Joint Position Control

High Speed Scaling: High speed scaling reduces execution speed when commanded speed exceeds a high speed velocity threshold **and** the arm's high-speed collision links are in collision.

Collision Avoidance: Applies offsets to joint commands based on depth of intersection between arm collision geometries and the opposing arm or torso.

Joint Velocity Clipping: Limits joint velocity command to not exceed maximum joint velocities.

Joint Limit Check: Validates that resulting joint position will be within joint limits. If not, no velocity will be commanded to any joint.

Collision Detection: If collision (impact) is detected, set position command to hold current compensating for the impact.

Joint Velocity Control

Gravity/Spring Compensation: The joint torque command is applied in addition to the gravity and S1 spring compensation torques.

Joint Torque Scaling: Scales all joint torques if a torque command exceeds the maximum allowable torque for that joint. This scaling ratio is defined as torque_max / torque_command.

Joint Torque Control

Delta Max Clipping: The joint command will be clipped based on the delta max (offset from current position defined by max joint velocity)

Joint Limit Clipping: If the joint command is beyond limits, clip the command to respect joint limits.

Collision Detection: If collision (impact) is detected, set position command to hold current compensating for the impact.

“Raw” Joint Position Control

Figure 4.1: These flowcharts describe the control modes of the Baxter Research Robot, and were posted by Kyle Maroney of Rethink Robotics, Inc. on May 20, 2014. As of the date of access (April 15, 2021), they can be found at https://sdk.rethinkrobotics.com/ wiki/Arm_Control_Modes, which provides additional information on each mode. In Section 4.2 we discuss the performance of each control mode at bringing a joint to a set point angle. We concluded that Joint Torque Control mode is unreliable for our particular robot. We consider the trade off between the simplicity of using Joint Position Control, which performs additional steps that allow the lowest level control to be abstracted away, and Joint Velocity Control which is flexible enough to facilitate custom control laws to compute the velocity commands and a process where the agent could autonomously learn to perform low-level control. We demonstrate how the flexibility of Joint Velocity Control allows additional experimentation in Section 4.2.1.3, and will continue hypothesis testing in future work. Outside of this section, we use Joint Position Control for all experiments in this work.

that are not a focus of this work (such as which motor commands and resulting torques to use to rotate the joints in a particular way while considering their current angle and compensating for gravity). Abstracting away the concerns for low level control by relying on the built-in processing of Joint Position Control mode allowed the agent to learn high level actions and demonstrate results before we had designed and completed experiments where the agent could be provided with a simple control law and learn the parameters necessary for the agent to reliably use a different control mode.

However, there are limitations to working with the Joint Position Control mode. One is that it constrains how low level of motions and component actions can be included in the autonomous learning process, as the steps built in to the control mode can be argued to be knowledge provided an expert designer, and these steps produce motor signals that include a high degree of fine-tuning that a natural agent would not have at this point. While we accepted an assumption that these capabilities would be innate for our learning agent while using this control mode - and this enabled us to focus sooner on the reaching, grasping, and placing actions learned in Chapters 5, 6, and 7 - cases could be made for different sets of assumptions to facilitate different start and end points for the agent's learning, or different learning methods. While no single assumption was particularly problematic, Joint Position Control can be limiting for the experiments because it forces a fairly large and specific set of assumptions and methods for the motion of the arm. The assumptions cannot be relaxed and some qualities of the motion cannot be changed because of the large number of built-in steps and the sometimes significant impact they can have on the final version of the commands sent to the joint control boards and resulting motions. While these assumptions were acceptable for our experiments, the lack of flexibility is a potential issue for other users. The case can also be made that the agent learns more autonomously in a different control mode, where less is built in and the agent must learn to compensate for these "missing" steps. We plan to change away from Joint Position Control mode for our experiments in future work. This decision is partially motivated by a desire to increase the flexibility of assumptions, and so that more qualities of the agent's behaviors even in low level motions are determined more by the agent's experiences and learning from them, and determined less by expert design.

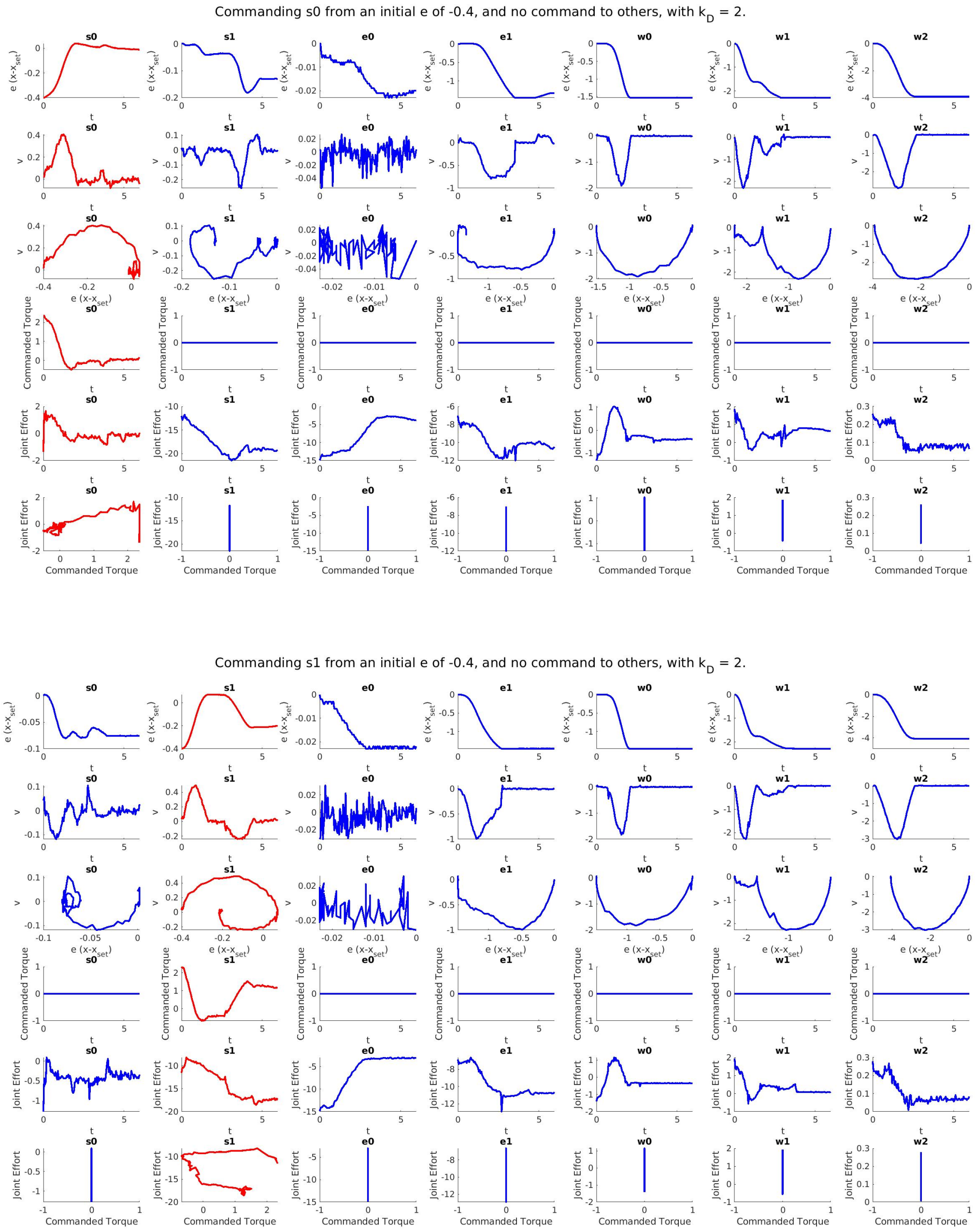
Our plan to transition away from Joint Position Control mode is also made in order to better answer specific questions we are interested in investigating in future work. A peculiar quality of infant motions during pre-reaching and early reaching is the use of a series of jerky submotions instead of a single smooth trajectory, as would be used by more skilled older children and adults. The PPS Graph model (discussed in chapter 3 and instantiated in section 4.3) provides one theoretical explanation for these submotions as the result of the agent moving along edges of the graph using the PPS Graph and Joint Position Control mode. Since the agent moves along graph edges to previously visited positions, the motion converges to the set point configuration of each node along the trajectory and then switches to move along the next edge, potentially changing the direction of each joint's motion, producing a jerk. However, it is unclear how what portion of this behavior is caused by using the PPS Graph trajectories and the limited set of nodes and edges. In future work, we would like to evaluate if other explanations could be theorized and determined to better fit the observed behavior or explain additional portions that are not well explained by properties of the PPS Graph model alone. In order to evaluate these alternative theories, we would need to implement them in the agent's control law or motion planner and observe the resulting behavior to determine its consistency with infant motion. However, adjusting these parameters, such as those that control the amount of damping for the motion, is not possible while using Joint Position Control mode because of the high number of built-in steps and required default settings. While less steps are taken to improve the expected quality of the motion with "Raw" Joint Position Control mode, it produces a similarly fixed response for each commanded set point and displacement from the current joint angle positions. Therefore, future work should use either Joint Velocity Control mode or Joint Torque Control mode to facilitate these experiments. The following sections justify Joint Velocity Control mode as the ideal choice for future work and demonstrate that it can be used to produce qualitatively different motions. We have also observed that it can produce a PPS Graph with the same of nodes as one built using Joint Position Control, supporting a claim that the same action learning could take place in a graph using either mode and that this change in future work will not require the actions to be relearned.

4.2.1.2 Demonstrating that our Robot's Torque Control is Unreliable

When using the built-in Joint Position Control and Joint Velocity Control modes for our robot, any joint can be commanded to a set point and reliably reach within a small threshold of that angle, regardless of the starting position of that joint and if the motion is concurrent with other joints. For Joint Position Control, the default error tolerance was 0.0087 radians, though observed errors were significantly smaller in practice, and this threshold could be set lower (0.005 for some implementations of the PPS Graph model) without altering the appearance of the robot's motion or the time required to complete the motion to an angle within this tolerated range around the set point. For Joint Velocity Control there was no default stopping condition for the motion, and tolerance was set to 0.01. When both the angular positional error from the set point and the velocity where less than this tolerance, the controller set the velocity to zero and the motion was considered complete. We found that the errors produced were similar in magnitude to those produced using Joint Position Control. We also determined that error does not propagate for either control mode over a large set of moves, so attempts to move to a sequence of set points will continue to converge to each set point in order. A specific example where the nodes of the Explored PPS Graph were revisited using Joint Position Control to retake the images and where the nodes were revisited using Joint Velocity Control to produce the Improved Explored PPS Graph (Section 3.2.2.3) showed that both modes could reliably visit the same set of configurations, within the tolerated error. In general, revisiting each node with Joint Velocity control did not produce significantly different deviations from the original node configurations than when the nodes were revisited with Joint Position Control, which the original set was gathered with. These observations support our claim that the control mode may be changed independent of the rest of the learning process, and without additional changes being required.

In contrast, for our robot the built-in Joint Torque Control was inconsistent and had behaviors that could not be accounted for, producing orders of magnitude larger errors. While many experiments were performed to investigate the issue and attempt to isolate the cause and correct for it, none were completely successful. Perhaps the best illustration of the agent's errant behavior is the one found in Figures 4.2-4.4. In these experiments, the agent is using a control law where the commanded torque is calculated with $k_p = 6$ and $k_D = 2$, that is, the torque was set to negative six times the current angular positional error minus a damping term of two times the current angular velocity. The initial error was $e = -0.4$ for this experiment. This experiment was part of a larger set that used a grid search over different k_p, k_D, and initial e values, and each combination of settings produced similar results to those shown. In this experiment, the agent calculated and commanded torques for a single joint to bring it within the tolerated 0.01 error of the set point angle and zero velocity at the same time. In this experiment, the agent only achieves this goal in trials where s0 is being commanded, and that joint is brought to the desired state. When commanding e0, the joint converges to an angle slightly outside the tolerated range, and trials are even less successful for all other joints. Furthermore, despite no commands being sent to the other joints while one is commanded, all other joints moved unpredictably, ending up quite displaced from the original angle that was intended be held constant (the error values shown for the joints that were not commanded treat the initial position as the set point). As can be seen from Figures 4.2-4.4, this is the case when any joint is the commanded joint, and motion of joints without commanded torques occurs whether those joints are more proximal or more distal than the commanded joint.

We hypothesized that there may be an undesirable default or time-out behavior when a joint is not commanded a torque. To test this, we repeated the experiment so that one joint in each trial still received a torque calculated by the control law and all other joints were constantly commanded with zero torque. These results were qualitatively the same as when the joints intended to be held steady were sent no commands at all. We also ensured that a zero command was not being treated as no command and leading to a time-out behavior by testing again with each joint that should be held steady being commanded with a small constant nonzero torque, such as 0.01. This torque should not be enough to overcome the innate friction and damping of the arm, so should be a possible way to command joints to hold a fixed angle if commands of zero are not allowed. Again, we saw the same type of errant behavior.


Commanding s0 from an initial e of -0.4, and no command to others, with k_D = 2.

Commanding s1 from an initial e of -0.4, and no command to others, with k_D = 2.

Figure 4.2: Data observed while the robot was commanded in Joint Torque Control Mode to move a single joint (shown with red plots) to its set point, which was 0.4 radians from the initial joint angle. All other joints (shown with blue plots) were sent no torque command. All plots in the same column display data for a single joint, and all plots in the same row are of the same type. The results from trials that command a shoulder joint are featured in this figure, and are unique because of their positional error over time plots in the first row. For s0, the error is successfully reduced below the tolerance threshold of 0.01, and after a small overshoot the joint angle converges near the set point. For s1, the error also approaches zero, but then converges to -0.2 instead. In addition to this failure to converge to the set point, for both shoulder joint trials note the problematic movement of the other joints that were not commanded (as seen by the constant 0 commanded torque over time), which is caused by joint efforts that are spontaneously applied without a known cause.

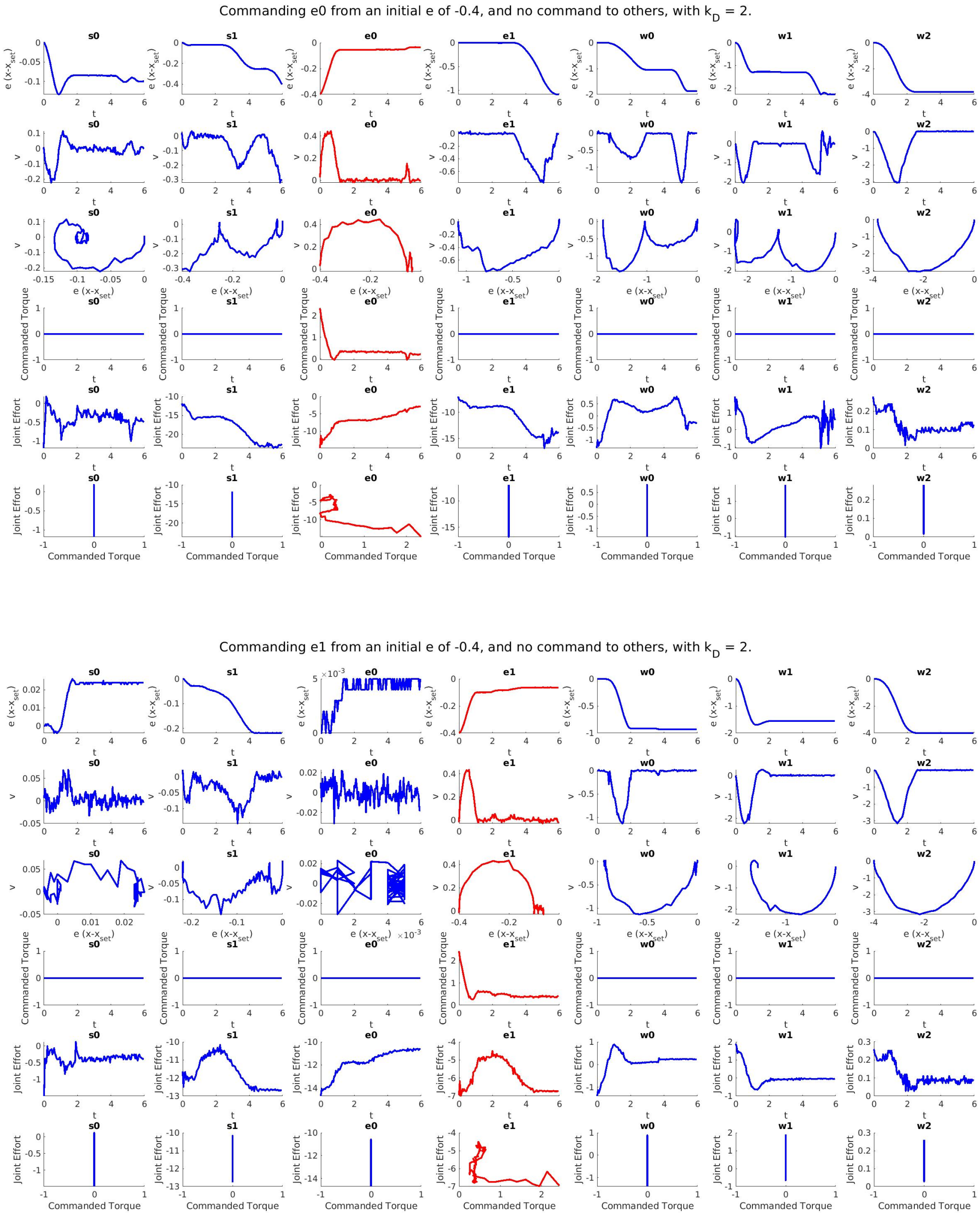


Figure 4.3: Analogous to Figure 4.2, except with a focus on the trials where the elbow joints e0 and e1 are commanded with torques intended to bring the joint to a set point angle. In this case both e0 and e1 are nearly moved to the set point, but fail to reduce the final error below the threshold of 0.01. Also note that these trials include the unexpected motion of all other joints, even though they were not commanded to apply any torque.

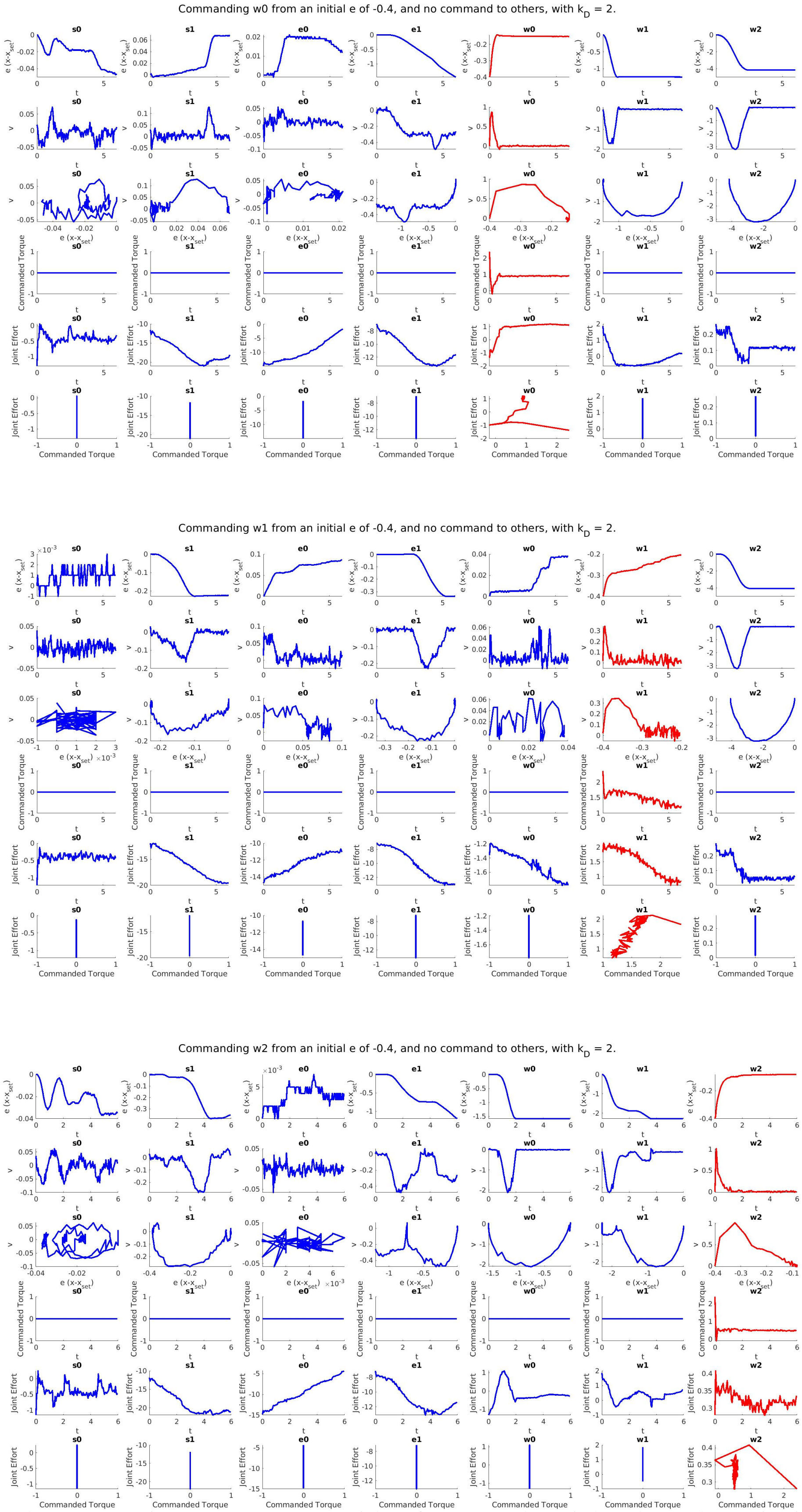


Figure 4.4: Analogous to Figure 4.2, except with a focus on the trials where the wrist joints w0, w1, and w2 are commanded. In each case the robot fails to bring the commanded joint to the desired set point angle, and the other joints exhibit undesired motions away from their positions that should have been held constant.

In further experiments, multiple joints were commanded at a time to resolve nonzero errors from set point angles. In these cases, some joints would control as desired and arrive at the tolerated range, while others would increase in error or fail to converge to the set point. Perhaps the most alarming result was observed in a set of experiments where every joint was commanded to apply zero torque and had a goal of maintaining the initial angle for the duration of the trial. Without being commanded to move any of the joints, the robot would apply significant efforts and cause large displacements, especially with the shoulder, and high velocities, especially with the smaller wrist joints. This drastic failure to maintain a position along with the issues that arise when moving one or more joints to a set point demonstrate that at least on our particular robot, the Joint Torque Control mode will be insufficient for learning and performing the manipulation actions in this work.

Before ruling out Joint Torque Control completely, we examined the relationship between values reported by functions in the ROS interface. The actual joint effort reported is often different from the sum of all reported efforts, which is itself different from the commanded torque. Another controller uses a gravity model with preset parameters to provide a gravity compensation effort, which by default is applied continuously to maintain the current position. There is also a hysteresis model that in the case of the s1 joint provides a nonzero effort to compensate for the large external spring. As seen in Figure 4.5, the actual effort does not always match up well with either the effort commanded according to our control law or the sum of this commanded effort and all other efforts being applied. This suggests either that there are problems in the reporting of these values, or that an unreported effort is the cause of the undesired motions that have been observed. In either case, further debugging of the issue was not feasible. We do not believe these problems to be universal for all torque-based controllers or for all Baxter Research Robots, but just for our particular robot, where there may be an issue with the calibration or hardware, or another unknown factor that prevents the Joint Torque Control mode from working as intended.

Given the problems our robot displays when Joint Torque Control mode is used, we choose the Joint Velocity Control mode as the more flexible option that will allow us to incorporate new parameters and test additional hypotheses about early actions. These tests are largely left for later work, but the ability to observe motions with different damping qualities is demonstrated in the following section. After this demonstration, we have opted to use the Joint Position Control mode for the remainder of this work due to its simplicity and better documentation from popular use.

4.2.1.3 Producing Qualitatively Different Motions with Joint Velocity Control Mode

When using Joint Position Control mode, the agent can command a desired position (an angle) for each joint and a built-in controller with fixed parameters moves the arm until each joint's set point is reached. This abstracts away the task of low-level control and leaves to the agent the decision of which joint angles are desired. This abstraction is not always desirable, since it requires

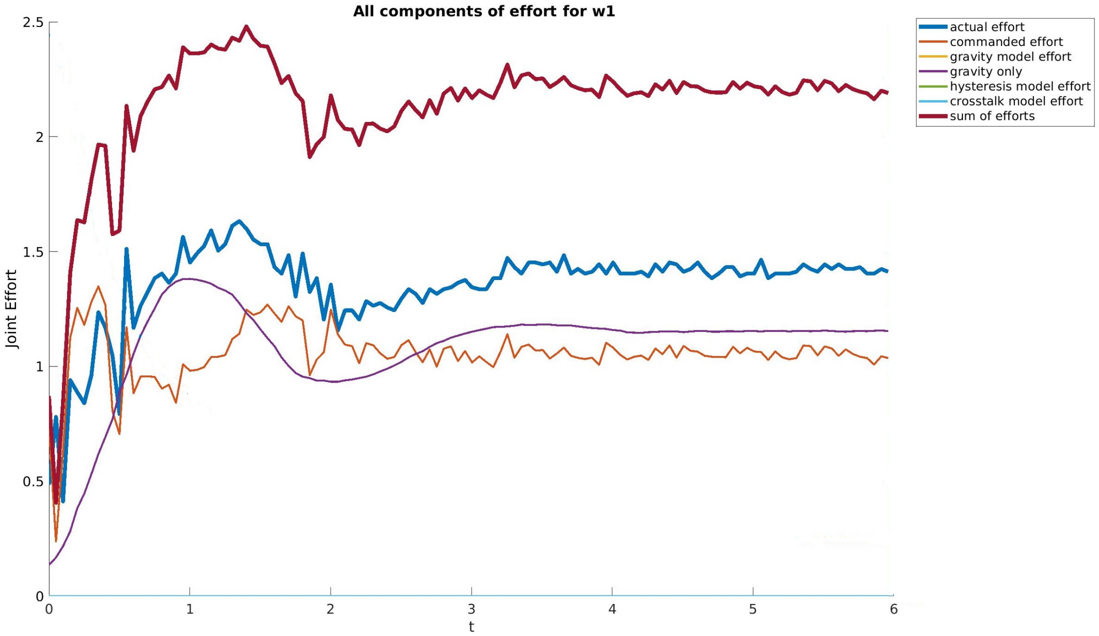


Figure 4.5: Effort values reported while the w1 joint was commanded torques over a period of six seconds. The error magnitude started at 0.1, but the joint overshot the set point and ended with an error magnitude of 0.35 on the opposite side. The angular position of w1 did not change after the first 2 seconds of the trial, even though some efforts continue to change after that point. This figure demonstrates the disparity between the sum of all efforts reported and the actual effort produced by the motors as broadcast in the robot's status. The actual effort does not appear to exclude the background processes such as gravity compensation or the hysteresis model, since it does not match the commanded effort alone either. We found that the difference between these values was not consistent between trials, and also varied depending on the joint or joints in use and their starting positions. The data available does not allow us to determine how the actual effort of each motor is produced, as there may be unreported components or these component values may be incorrect. The breakdown of efforts fails to provide insight on how the joints could move significantly without any torque commanded, and we conclude that Joint Torque Control mode cannot be used reliably with our robot for the experiments in this work.

assuming the low-level control is innate and functions in a way that was programmed in with expert knowledge, rather than a way which the agent autonomously learns. This assumption is minor compared to other common assumptions that may involve kinematics or geometric models, and influence both the low-level control and high-level decision making tasks.

While we accept the assumptions of Joint Position Control for this work, we demonstrate an additional advantage of Joint Velocity Control to be explored further in future work, which is the flexibility of a custom control law. In our case, the velocity to command is calculated as $-k_p \cdot e - k_D \cdot v$. In our testing, we found that the innate damping of the arm was strong enough that adding additional damping with a positive value of the coefficient k_D always produced divergent motion. We then set $k_D = 0$ for all remaining trials, where k_p could still be altered to determine the proportionality between the error from the set point and the velocity to command.

In Figure 4.6, we display an example motion between two random joint angle vectors $\mathbf{q}$ and $\mathbf{q}'$ while using four different values of k_p in the control law. These four settings were found as part of a larger search over k_p values, and were selected for presentation due to their exhibition of four qualitatively different motions. $k_p = 0.5$ produces overdamped motion, $k_p = 1.25$ produces near critically damped motion, $k_p = 3.0$ produces underdamped motion, and $k_p = 6.0$ produces divergent motion. In future work, the most significant of these are likely to be the efficient near critically damped motion that resembles a skilled agent's movement, and the underdamped motion with overshooting that resembles the movement of an unskilled agent, such as an infant learning early move and reach actions. The motion of s0 provides the clearest example of the four qualities achieved with these settings, and the plots for s0 are presented in larger size in Figure 4.7 for more detailed viewing. These can also be compared to the plots in Figure 4.8 where Joint Position Control is used to perform the same motion with s0. The added flexibility of Joint Velocity Control that makes it desirable for future work is on display in this comparison, since the fixed settings of Joint Position Control can only produce a single behavior during this move. However, this behavior is sufficient for the learning performed in this work, allowing us to continue at this time with the simpler mode, Joint Position Control.

4.2.1.4 Note on Relation to Autonomous Learning

It should be noted that the process of selecting between the available control modes is experimenter-driven and that the results evaluated may vary for different robots. However, we do not consider this to interfere with the independence of our learning agent or its autonomy as it learns to move the arm or to reach, grasp, and place objects. Our justification is that a natural agent does not appear to have multiple control modes available, so it is acceptable for the agent's autonomous learning to begin after the experimenters have already gathered the data necessary and selected a control mode that will be used in the remaining experiments as the agent moves and gathers experience.

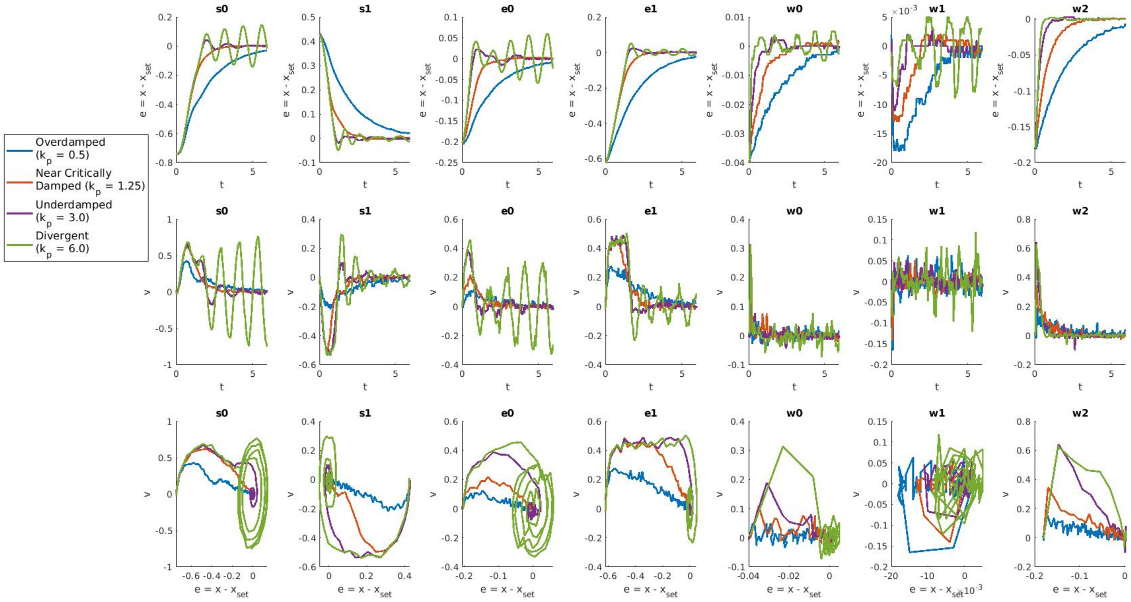


Figure 4.6: Plots displaying the positional error over time, velocity over time, and phase diagram for moves of each joint performed with Joint Velocity Control and varied parameters for the control law. The robot's built-in Joint Velocity Control mode is more flexible than its Joint Position Control mode, since it can be used with a custom control law with variable parameters instead of a single default. We found that the innate friction and damping of the robot's arm was sufficient, and that performance was best when the damping term k_D was set to zero. Then varying the coefficient proportional to the error k_p allowed the robot to produce qualitatively different motion behaviors. When $k_p = 1.25$, the motion is nearly critically damped, and converges to the set point relatively quickly without overshooting. When $k_p = 3.0$, underdamped motion can be observed, where the joint angle arrives at the set point angle with too much velocity remaining and overshoots one or more times, but eventually slows and converges. This is an important behavior to be able to observe, as it resembles the motion of infants performing pre-reaching motions and early reaches. Underdamping has been proposed as one possible cause of infants' jerky submotions, and the ability to adjust a parameter to introduce or remove underdamping and compare the qualities of the resulting motion to those of infants can be valuable for showing consistency or inconsistency with that theory. Our PPS Graph model also suggests another plausible explanation, that the agent may exhibit jerky submotions as it moves between familiar poses. The motions produced by executing trajectories in this manner can also be compared to infant motions to determine if either theory or a combination is best supported by consistency with our implementation. It is also possible with $k_p = 0.5$ to observe overdamped motion where the joint converges too slowly to the set point angle. Finally, divergent motion can be achieved with $k_p = 6.0$ or greater, where the oscillations from overshooting the set point continue to grow over time. If this result is replicated, take care to have a clear space and a short timeout period to ensure safety as the motion will grow larger until it is stopped. While all joints exhibit the four qualitatively different behaviors, the example with s0 shows the characteristics most clearly, and these are presented in larger size in Figure 4.7 and compared with the corresponding figures for Joint Position Control in Figure 4.8.

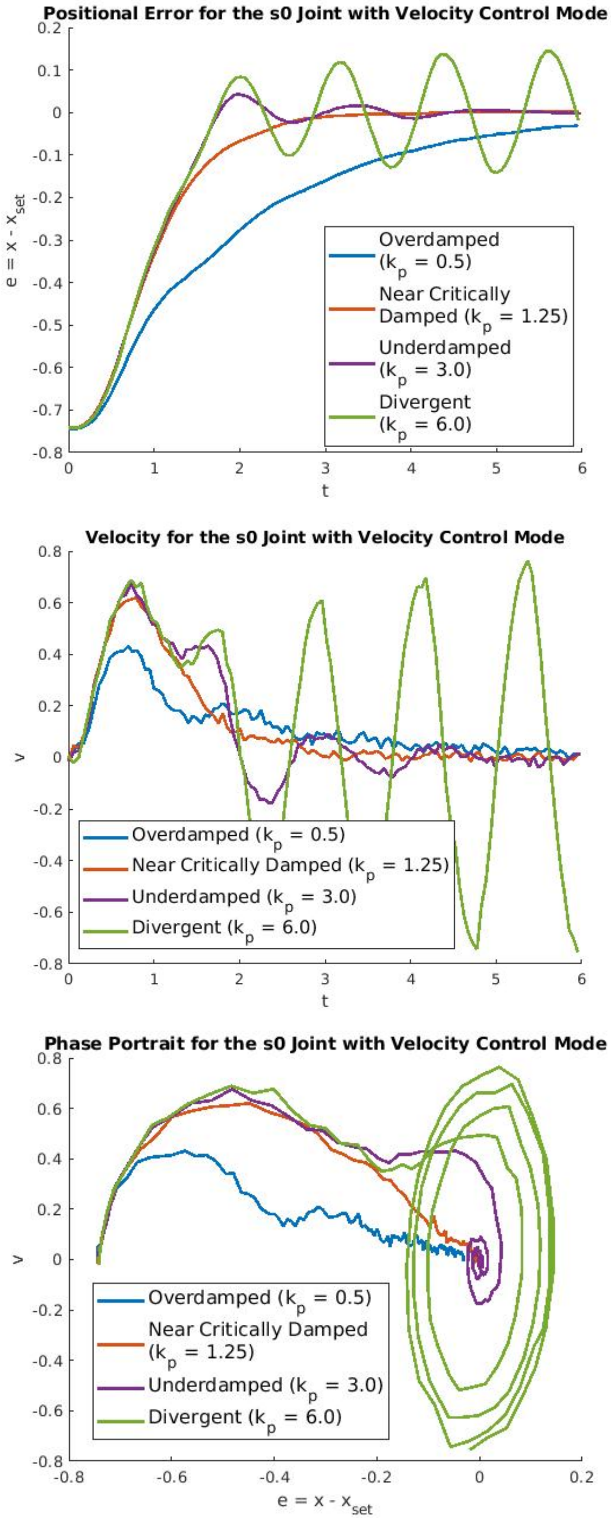


Figure 4.7: Larger views of the error, velocity, and phase portrait for the s0 joint that appeared alongside the other joints in Figure 4.6. By using Joint Velocity Control adjusting the control law parameter k_p and relying on the innate damping of the arm, overdamped, near-critically damped, underdamped, and divergent motion can all be produced with our robot. The ability to produce these qualitatively different behaviors is important for checking the consistency of our model with observed infant behaviors and theories for the causes of those behaviors. This is not possible to the same extent with the less flexible Joint Position Control mode, which has results shown for the same motion in Figure 4.8.

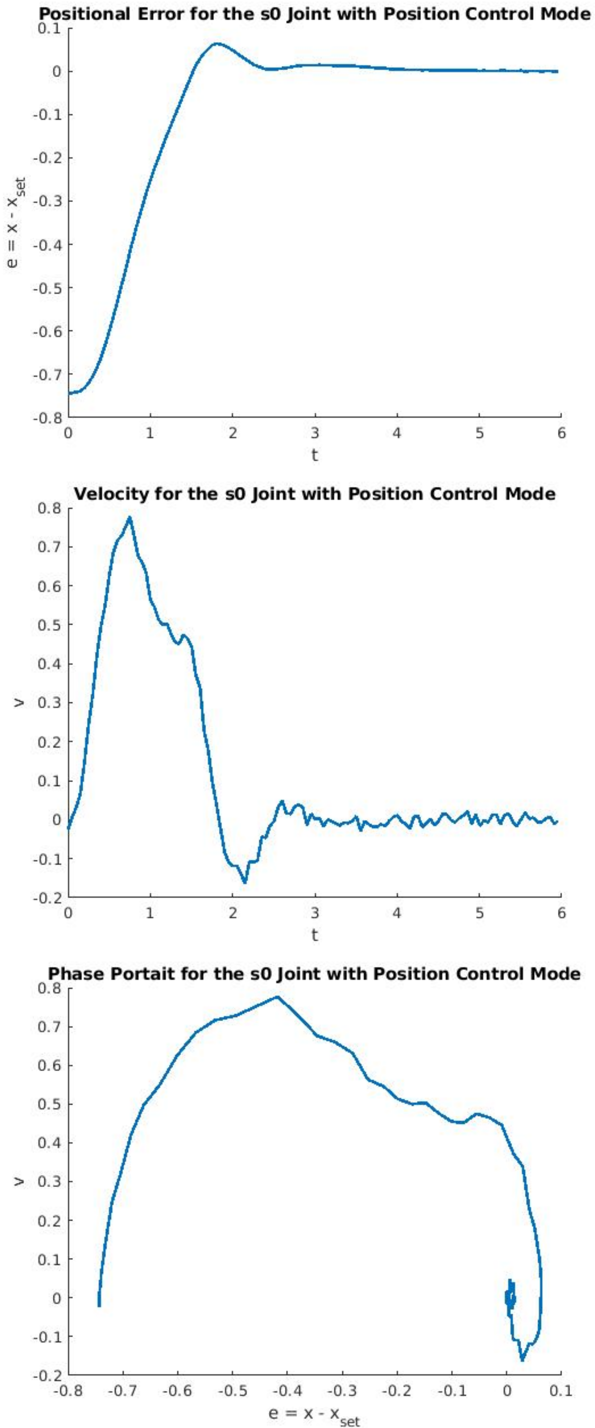


Figure 4.8: Plots of the error, velocity, and phase portrait for the same motion with s0 as performed in Figure 4.7, but using Joint Position Control. Joint Position Control only allows customization of the error tolerance and timeout length for a motion, and requires only an input of the set point angle for each move. All other parameters are set internally to defaults, and the motion is processed to have standard desirable qualities with the steps in Figure 4.1. With the limited options, only a single qualitative motion behavior can be observed. Typically, the controller for Joint Position Control is near-critically damped, but in the case of this large initial displacement there is a slight overshot before converging, and the motion appears slightly underdamped. While Joint Position Control lacks flexibility that would allow for additional hypothesis testing, we use it for all remaining experiments in this work due to its straightforward usage and high reliability.

In this way, the agent is presented with a single control mode to use, but is not given additional information on how to use it, therefore maintaining autonomy in the learning stages. Further, it is also reasonable to assume the experimenter's selection of the control mode because the remaining steps are not dependent on any specific mode being used. In particular, we have determined that the same graph could be constructed with either Joint Position Control or Joint Velocity Control. Since the experience gained would be taken from events taking place at the same set of graph nodes, the learning should also be similar and analogous reach, grasp, and place actions could be found if a different control mode was chosen.

4.3 Building the Peripersonal Space Graph

4.3.1 A Move Arm Action using Move Joint Actions in Parallel

By the end of the previous section, the agent has two usable methods for moving the joints of the arm. One is the Baxter Research Robot's built-in angular position control mode, called "Joint Position Control" in the robot's documentation. This method can be assumed to be capable of taking a joint to a set point (a specified angle) without any additional learning. The second is the Baxter Research Robot's built-in angular velocity control mode, Joint Velocity Control, which required the learning of suitable parameters for a proportional derivative (PD) controller to move any of its joints to a set angle.

The experiments completed in the previous section narrowed the options for methods from the four control modes available for the Baxter Research Robot to either Joint Position Control or Joint Velocity Control. As we discussed in the previous section, Joint Velocity Control gives the highest number of areas where the agent can learn autonomously. It also allows further evaluation of theoretical explanations for motion qualities observed in infants, as it gives the ability to control the properties of the motor commands, such as damping and other properties of the motor commands and the resulting motion. In contrast, in Joint Position Control mode these properties all have assumed default values. However, giving the agent the responsibility for setting these parameters requires additional learning before the steps where learning from unusual events is demonstrated. Further, the learned parameters allow the agent to reach the same end positions, but the motion between can be less refined than the motion guided by the built-in controller for Joint Position Control. This means using Joint Velocity Control introduces the possibility that unreliability in the learned actions could come from imperfections in the action policy or from imperfections in the underlying control laws, which would require additional steps to determine the quality of only the learned action policies independently. For these reasons, we have only used Joint Velocity Control mode in this chapter to evaluate its performance, and we leave the completion of all following experiments using Joint Velocity Control for future work.

At this time, we have only completed the full set of experiments with Joint Position Control mode. This control method requires the simplest inputs and handles the most underlying mechanics for the user, and was used to simplify the control portion of this work, especially beneficial in the early experiments. When using Joint Position Control, the agent performs linear interpolation in configuration space to move from one position to another, and adjusts the torques applied so that the motion is critically damped and all joints arrive at the same time. Again note that similar motions can be achieved with Joint Velocity Control mode and enough tuning of the parameters, and we assume that when the action learning experiments are repeated with velocity control in future works the results will agree with the results of these first experiments and the claims they support. To begin supporting the validity of this assumption, in the upcoming analysis of the Peripersonal Space (PPS) Graph built with Joint Position Control, we show that the set of configurations stored in the nodes of a PPS Graph built with Joint Velocity Control is the same within a small margin of error.

As the underlying control for the robot handles gravity compensation and other factors that depend on the motion of other joints, the motor signals necessary to move each joint can be considered independent. We also assume that since motion of one joint does not affect the prerequisites of the motion of another, the agent can combine the move joint actions in parallel, bringing all the joints to their set point angles in the same motion instead of a sequence of one-joint motions. The act of performing the move joint actions, where each joint is brought to a desired set point angle, in parallel will define the earliest version of the move arm action, sufficient for bringing the arm to a nearby configuration and performing motor babbling.

4.3.2 Motivation for Additional Learning to Move

At first, the capability to move all of the joints to set points in unison may seem like all that is required for a satisfactory move arm action. However, there are two key shortcomings that the agent will need to address by gaining additional experience that will be recorded in the PPS Graph. First, the agent has no guarantee of the safety and feasibility of the motion. Especially for motions where the initial configuration and set points are distant from each other, the agent cannot guarantee no portion of the arm will collide with the environment without assuming significantly higher information and capabilities, such as a 3D point cloud of the obstacles and a motion planner to adjust the trajectory around them. One should note that while the path will be linear in configuration space, it will very seldom be linear in image space or standard 3D axes for the workspace, and the area swept through for a long trajectory is difficult to predict as the arm rotates at multiple degrees of freedom. The risk is lessened for the Baxter Research Robot since the position control mode places caps on the velocity and torque applied to each joint, so a command for a large change will not result in potential damage to the hardware or danger to those nearby. Further, Baxter has compliant actuation, and does not attempt to force its way past obstructions. Still, given the relative fragility of any robot when compared to natural agents that can respond to minimize pain and heal afterwards, moves should only be executed if there is a demonstration or justified assumption that they will be collision free.

The second reason that the ability to move the joints to set points needs to be combined with additional knowledge is that the agent has only used proprioception up to this point to monitor the angular position and velocity of each joint. While this gives the agent the capability of moving to a desired configuration by learning to use the velocity controller (or if the capability is assumed due to use of the position controller), the agent would not have a model for how the hand and arm move in the environment and could interact with other objects. While the mechanics of moving the arm will not change by creating the PPS Graph, the data contained by the graph will allow the agent to select its destinations to perform actions that influence the environment in desirable ways. In particular, because the PPS Graph provides a discrete mapping between points in configuration space and image space, the agent can now plan and make moves to desirable positions in the image by moving to the configuration that best corresponds to the desired position in the image, such as a position matching the center of an object that is the target of a reach or grasp.

4.3.3 Experiment 4.1: Creating the Peripersonal Space Graph

Keeping with the general goal of being consistent with natural agents, the method by which our agent builds the PPS Graph is inspired by the early motions of human infants. A baby begins to explore its environment and the range of motion of its arms with seemingly random movements and no clear external goal.

Our embodied robot agent will perform motor babbling to observe similar exploratory motions. Recall that we use a Baxter Research Robot, and in this work focus exclusively on motions made with its left arm and grippers. The state of this arm can be given by eight degrees of freedom: a vector of seven joint angles, $\mathbf{q} = \langle q^1, \ldots, q^7 \rangle$ corresponding to the vector of joints $\langle s_0, s_1, e_0, e_1, w_0, w_1, w_2 \rangle$, and the aperture a between the gripper fingers, described as a fraction of its maximum width.

The robot learning agent creates a PPS graph $\mathcal{P}$ of N nodes by sampling the configuration space of its arm. From an initial pose $\mathbf{q}_0$ in an empty environment, the robot samples a sequence of perturbations $\Delta\mathbf{q}$ from a distribution $\mathcal{D}$ to generate a sequence of poses:

$$\mathbf{q}_{i+1} = \mathbf{q}_i + \Delta\mathbf{q}_i \text{ while } i \in [0, N-1] \tag{4.1}$$

For our experiment, the agent begins with an empty PPS Graph $\mathcal{P}$. We place the robot's left arm so that the hand is at an arbitrary position near the middle of field of view above the table, and the agent records the configuration that holds the arm at that position in the first node, as $\mathbf{q}_0 = \mathbf{q}(n_0)$. After the special case of $\mathbf{q}_0$, each configuration $\mathbf{q}_{i+1}$ is found using the random motor babbling search

Joint Name	Index in **q**	Hardware (Radians)				Software (Radians)		
		Min	Max	Range	σ_k	Min	Max	Range
s0	1	-1.7016	1.7016	3.4033	0.34033	-0.5	1.5	2.0
s1	2	-2.147	1.047	3.194	0.3194	-0.65	0.5	1.15
e0	3	-3.0541	3.0541	6.1083	0.61083	-1.5	1.5	3.0
e1	4	-0.05	2.618	2.67	0.267	0.5	2.0	1.5
w0	5	-3.059	3.059	6.117	0.6117	-1.5	1.5	3.0
w1	6	-1.5707	2.094	3.6647	0.36647	- 1	1.5	2.5
w2	7	-3.059	3.059	6.117	0.6117	-2.8	2.8	5.6

Table 4.1: The joints of the Baxter Research Robot in proximal to distal order, which matches the order of their appearance in the configuration vector **q**. This table gives the allowed angles of each joint in terms of the physical limitations of the hardware and the more restrictive bounds imposed in the software for conducting the motor babbling to build the PPS Graph. The value listed for σ is 10% of the joint's hardware range, and was used as the standard deviation when sampling deltas for that joint from a normal distribution.

described in equation (4.1) from the previous configuration $\mathbf{q}_i$ and a straightforward instantiation for $\mathcal{D}$ that contains appropriate $\Delta\mathbf{q}_i$ for our robot. For each joint angle k, the displacement to add is sampled from a normal distribution with a standard deviation σ_k equal to a tenth of the full range of that joint. That is, $\mathcal{D} = N(0, \sigma_k)$, and the proposed next angle for each joint k can be calculated as

$$q_{i+1}^k = q_i^k + \Delta q^k \text{ where } \Delta q^k \sim N(0, \sigma_k) \text{ and } \sigma_k = 0.1 \cdot range(q^k). \tag{4.2}$$

Additional information on the the ranges of angles that each joint can hold given the constraints of the robot's hardware and the value of each σ_k can be found in Table 4.1. Note that the table also contains a smaller "software" range of allowed angles for each joint defined by the code for motor babbling written for this experiment. Every joint has a software range that trims to extremes of the hardware ranges to avoid additional strain on the robot from motions at or near the rotational limits of the joints. Some joints, especially around the robot's elbow, have additional limitations put in place, which narrows the range of motion of the agent's robot arm to more closely resemble the range of motion in a typical human arm. If any q_{i+1}^k falls outside the allowed range, a new Δq^k is sampled until the resulting $q_{i+1}^k = q_i^k + \Delta q^k$ is permitted, without recording the rejected angles or subtracting from the number of remaining nodes to generate.

We also use a form of *rejection sampling* to impose biases similar to those observed in infant behaviors. While the motor babbling of human infants may appear random, it does exhibit biases toward moving objects and toward keeping the hand visible [54, 55, 53, 52]. Therefore, we require that the resulting end-effector pose must fall within the field of view. Human infants are also soft and robust, so they can detect and avoid collisions with minimal damage. Since this is not the case for the robot, each new configuration proposed by adding the sampled deltas must be evaluated for safety before executing the motion. To prevent damage to our Baxter Research Robot, the configuration must not cause the hand to collide either with the table or with the robot's own body, and must result in a hand position that will have a measurable depth in order to be recorded in the graph. We implement these checks using a manufacturer-provided forward kinematics model that is below the level of detail of our model, and is used nowhere else in its implementation. The exact criteria used are given in Table 4.2. If any of these conditions are violated, the proposed configuration is rejected and a new $\mathbf{q}_{i+1}$ is sampled (in this case a new Δq^k is sampled for each joint k, rather than for a single joint when a proposed angle q^k_{i+1} is rejected for falling outside of the allowed range). In future work, we will considering additionally biasing the sampling to resemble human infants' pre-reaching motions toward objects, or to move in a cyclic fashion, often returning to the center of the field of view.

After all checks and any necessary resamplings are completed, the arm is physically moved from its current configuration $\mathbf{q}_i$ to the new configuration $\mathbf{q}_{i+1}$. After each new pose has been safely reached by physical motion of the arm, the robot pauses and collects a perceptual image $I_{\mathbf{q}_{i+1}}$ that corresponds to $\mathbf{q}_{i+1}$. Each visual percept $I_{\mathbf{q}_{i+1}}$ is taken by a fixed-viewpoint RGB-D camera mounted above the robot's head. This camera provides an RGB image I_{RGB} and a depth-registered image I_D as shown in (Figure 4.9). Section 3.2.1.2 provides additional information on the stored raw camera images and the processed image manifolds that allow salient portions to be represented more concisely, such as the binary image mask for the robot's palm, p_{i+1}. After the image processing is complete (or after the raw images are saved and prepared for processing if the processing will be run in a batch at the end of motor babbling, as in our current implementation), the node $n_{i+1} = (\mathbf{q}_{i+1}, I_{\mathbf{q}_{i+1}})$ is added to the set of all nodes $\mathcal{N}$ and the undirected edge $e_{i,i+1} = (n_i, n_{i+1})$ is added to the set of all edges $\mathcal{E}$, building onto the PPS Graph $\mathcal{P} = \langle \mathcal{N}, \mathcal{E} \rangle$.

This experiment continues until the total number of nodes added to $\mathcal{P}$ is $|\mathcal{N}| = 3000$. In this work the agent is instructed to stop adding nodes at this set stopping point, but future work may evaluate the effects of different values of $|\mathcal{N}|$ or methods the agent could use to determine the best final size of $\mathcal{N}$. When the agent reaches $|\mathcal{N}| = 3000$ and stops motor babbling, the graph that has been built so far is a linear chain. As a result there is only a single path between any two nodes, and this path is typically very long since it must visit each consecutively numbered node between the starting and ending nodes. In addition to inefficiency, having a single path through the graph does not provide options for avoiding obstacles or selecting the most reliable approach for a learned action. The graph needs much higher connectivity, which can be obtained by adding new edges linking existing nodes in $\mathcal{P}$.

It is not feasible to test every pair of unconnected nodes, so we apply a heuristic. Recall from Section 3.2.1.3, which provides additional details on PPS Graph edges, that the length of an edge be

Allowed Range	Rejection Criterion	Reason for Rejection
$0.46 < x < 0.92$	$x \leq 0.46$	The hand must be at least a minimum distance in front of the robot's body to avoid triggering measures for avoiding self collisions.
	$x \geq 0.92$	Positions further forward than this will collide with a barrier used to provide a simpler background in some example photos.
$-0.4 < y < 0.4$	$y \leq -0.4$	Hand positions further past the left edge of the table will have lower utility and may be out of view.
	$y \geq 0.4$	Hand positions further past the right edge of the table will have lower utility and may be out of view.
$0 < z < 0.3$	$z \leq 0$	The hand must not be below the table's surface.
	$z \geq 0.3$	The hand must not be so far above the table that it is above the field of view.
	$\|\|(x, y, z) - (0.38, 0.12, 0.81)\|\| < 0.36$	Positions too close to the Kinect's location, $(0.38, 0.12, 0.81)$, will have an invalid depth reading.

Table 4.2: For safety, the agent performed rejection sampling while motor babbling, using a built-in forward kinematics function to check each proposed node configuration before moving to it. The rejection criteria used are explained above. These were used in addition to enforcing that each joint angle was within the ranges shown in Table 4.1. It is important to note that the agent is not aware of these safety checks and that neither forward or inverse kinematics are used in any other experiment in this work.

the Euclidean distance between the configurations at its endpoint nodes, considered in joint space.

$$||e_{ij}|| = d(n_i, n_j) = ||\mathbf{q}_i - \mathbf{q}_j||_2 \tag{4.3}$$

Also let $\widetilde{||e||}$ denote the median length of all the edges in the current (linear) graph. This is the average[1] length of edges known from exploration to be safe. The heuristic is that when two nodes n_i and n_j could be connected with an edge shorter than $\widetilde{||e||}$, that is, when $d(n_i, n_j) < \widetilde{||e||}$, then the edge e_{ij} is predicted to also be safe and can be added to $\mathcal{P}$, if it is not already present. Once an edge e_{ij} has been added, the agent can safely move along it from n_i to n_j. In practice, the heuristic has performed well, and no significant collisions have been caused by moving along these added edges. In all motions performed for this work, some minor glancing contacts with the table did occur along a select few edges, but not often enough or severe enough to suggest the heuristic failed to identify affordances for safe motion between nodes. If desired, the problematic edges can be removed by the experimenter as they are found as an added precaution, to ensure it will not be considered for use again. Since the current implementation for this motion relies on the built-in Joint Position Control mode, the agent uses linear interpolation to bring each joint angle q^k from its value in $\mathbf{q}_i$ to its value in $\mathbf{q}_j$.

With the inclusion of all edges that satisfy the safe motion heuristic condition, we expect that $\mathcal{P}$ will support planning of multiple trajectories between any given pair of nodes. Because $\mathcal{P}$ is still a sparse approximation to the configuration space, trajectories across the environment will tend to be jerky. As a final step of building the Explored PPS Graph, the the agent designates a *home node*, n_h, where the arm rests naturally and that allows relatively unoccluded observation of the environment. The concept of a home node has also been observed in some infants in [49], as some of their infant subjects appeared to move the hand to an out of the way position and better observe the environment before reaching, and some infants displayed cyclic motions, returning back to familiar positions before extending further. For a consistent behavior, we set the convention for our agent that all actions will be planned while at n_h, and all trajectories begin at n_h, and eventually return there, too. We will also define the terms n_f for the final node of a trajectory, and n_p for the penultimate node.

4.3.4 Experiment 4.1 Results: Analysis of the Explored PPS Graph

The agent successfully followed the motor babbling algorithm to make 2999 moves and record 3000 nodes. To see a demonstration of the positions visited and the space covered by the Explored

[1] We chose to use the median $\widetilde{||e||}$ rather than the mean μ_e for the heuristic as it is slightly lower and provides a more conservative prediction of the set of safe edges to add. $\mu_e > \widetilde{||e||}$ because of the sampling of deltas from a normal distribution, which produced a small number of long outlier length edges that bias the mean upwards. Another parameter that could be examined in future work is the effect on the connectivity of the graph and resulting motions of the agent if a different percentile threshold was used instead of the median's 50th percentile.

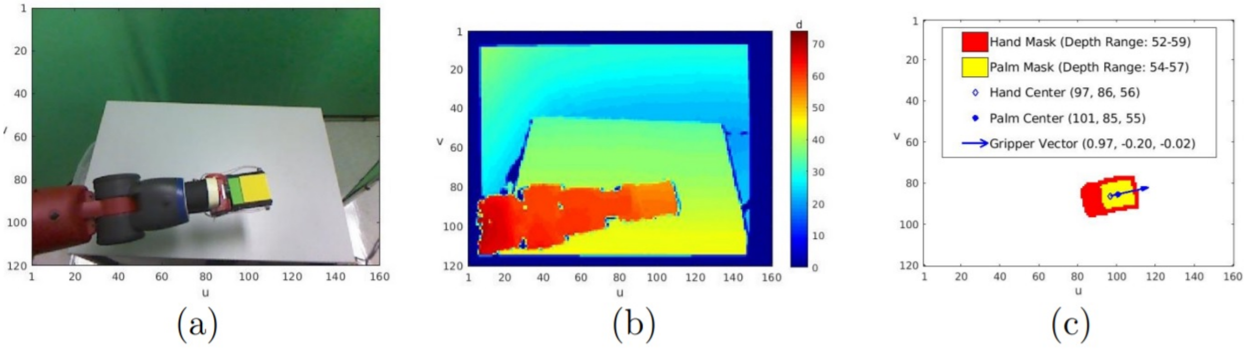
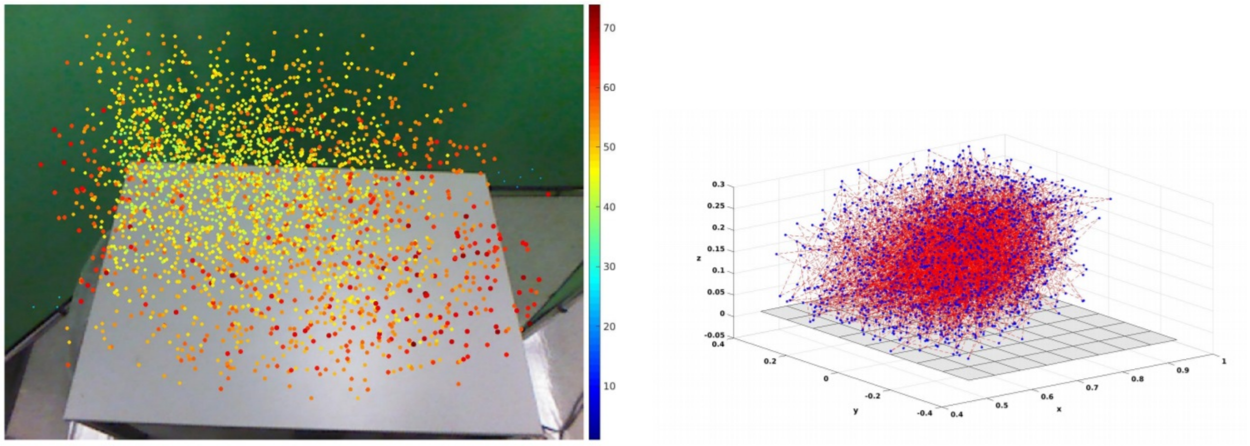


Figure 4.9: An example of the agent's visual percept and stored representation for a node n_i. **(a)** A single RGB image I_{RGB}, scaled down to 120×160 resolution, taken while the arm configuration is set to $\mathbf{q}_i = \mathbf{q}(n_i)$. **(b)** The registered depth image I_D taken at the same time. Note that the depth values are a measure of disparity, so smaller values are further from the camera. **(c)** The full representation the agent stores for the node n_i. Aided by the yellow block held between the gripper fingers, the agent segments the palm mask, corresponding to the grasping region of the hand. The larger hand mask includes the palm mask (shown in yellow) and parts of the robot image segment near the block, typically the gripper fingers and lower wrist (shown in red). The range of depth image values within each mask is also stored, as are the center of mass and mean depth value for each mask. Finally, to estimate the direction the grippers are pointing, a vector is drawn from the hand mask center through the palm mask center.

PPS Graph, the reader may wish to view one or both 5-minute video summaries of the motor babbling process that we have prepared. The video at https://www.youtube.com/watch?v=EsGCTS2vx1s displays the nodes and edges added to the graph in sequence, showing each in terms of the standard (x,y,z) coordinates of the end-effector in the 3D environment around the robot. The same representation of the graph being built edge by edge (not including the heuristic edges that are added in a step after motor babbling) is shown in the video at https://www.youtube.com/watch?v=7rAPpjFpNZw, but it also displays the corresponding I_{RGB} image recorded for each node as it was visited. When the process is complete, the Peripersonal Space Graph $\mathcal{P}$ is a sparse approximation of the configuration space of the robot arm (Figure 4.10). It is evident that random sampling through unguided exploration has distributed $|\mathcal{N}| = 3000$ nodes reasonably well throughout the workspace, with some localized sparse patches and a region in the far right corner that is generally out of reach of the robot's left hand.

With most workstation computer setups, this experiment should produce an instance of the Explored PPS Graph in approximately 5-10 hours. In our case, the initial motor babbling to record the 3000 nodes took place over 5 hours. This runtime included built-in pauses between motions to capture images while the robot was at a complete stop, which may need to be adjusted in length depending on the delay of the camera and the time required to write the image to disk. We chose to record only the raw RGB and disparity images during motor babbling and perform



(a)

(b)

Figure 4.10: Two visualizations of the Peripersonal Space (PPS) Graph $\mathcal{P}$, with size $|\mathcal{N}| = 3000$. Each visualization shows wide coverage that facilitates movement throughout the environment, with a few sparse patches. **(a)** An example RGB percept of the empty environment, overlayed with the (u, v) center of mass locations for all $|\mathcal{N}|$ nodes. The dot for each node is colored and sized according to its mean disparity value d (see key along right edge). Nodes with higher disparity (closer to the camera) appear larger and more red, while nodes with lower disparity (farther from the camera) appear smaller and closer to blue. The information in this display is stored in the individual node representations and is available to the agent. **(b)** The nodes of $\mathcal{P}$ displayed in the true world (x, y, z) coordinates of the Baxter Research Robot's default frame of reference. The gray plane represents the surface of the table. Nodes are plotted as blue points. The 2999 edges in the original chain from motor babbling are shown as dotted red lines. Not shown are the edges added according to the safe motion heuristic. The information in this display, or anything involving (x, y, z) coordinates, is not available to the agent.

the image processing to produce the binary image masks and other manifolds for all nodes in a batch afterwards. The distance-based safe motion heuristic must also be evaluated for each pair of nodes in the graph to determine which edges to add to finish building the graph. On our system, all additional processing was completed in MATLAB and representations were saved to disk for future use in about 3 hours.

Due to the arbitrary choice of $\mathbf{q}_0$ and the randomness introduced when sampling each Δq^k, each time this experiment is repeated it will produce a different Explored PPS Graph. However, this should not affect the remaining experiments, as none of our results suggest that the performance of the agent is dependent on specifics of our graph instance that would differ in another random graph instance. If desired, it is possible to recreate a graph that is effectively identical to our graph by sending the arm to each $\mathbf{q}_i$ and recording $I_{\mathbf{q}_i}$ for your robot and environment. We have performed this successfully to account for changes to the camera's position relative to the robot and to update the appearance of the background in the stored images to be consistent with the current environment of the robot.

In the remainder of this section, we examine the numerical statistics of the Explored PPS Graph and discuss the significance of the graph for the agent's ability to move the arm and to represent its peripersonal space.

4.3.4.1 Quantitative Analysis

Random exploration of the configuration space with a stopping point target of $|\mathcal{N}| = 3000$ creates 3000 nodes, connected in a chain with 2999 edges. The distribution of the lengths of the explored edges is given in Figure 4.11. As expected, the distribution is skewed to the right, with a few long edges that skew the mean length, $\mu_e = 0.96$, to be greater than the median length, $\widetilde{||e||} = 0.92$. Since these lengths are calculated in terms of configuration-space distance, the Euclidean distance between joint angle vectors as in Equation 4.3, these numbers can be unintuitive on their own. However, these numbers, and the maximum explored edge length of 2.3 can be compared with the maximum possible move length to show that they are relatively small moves. This maximum possible move length can be found as the distance between extremes of the configuration space, such as from a position with every joint at its minimum angle to a position with every joint at its maximum angle. This length is 12.4 using the ranges with respect to the hardware limits, and this length is 7.9 with respect to the more restricted joint angle ranges in our software.

Of the original 2999 edges, 1499 are shorter than the median edge length $\widetilde{||e||}$ and 1614 are shorter than the mean edge length μ_e. Using the assumption that moves of about the same distance as the lengths of the already-traversed edges will be equally safe allowed the application of a heuristic test for safe motion. In particular, for any two nodes $n_i, n_j \in \mathcal{N}$, if $d(n_i, n_j) < \widetilde{||e||}$, then e_{ij} is added to $\mathcal{E}$. 108,718 edges were added by this method, bringing the total to $|\mathcal{E}| = 111,717$. Since

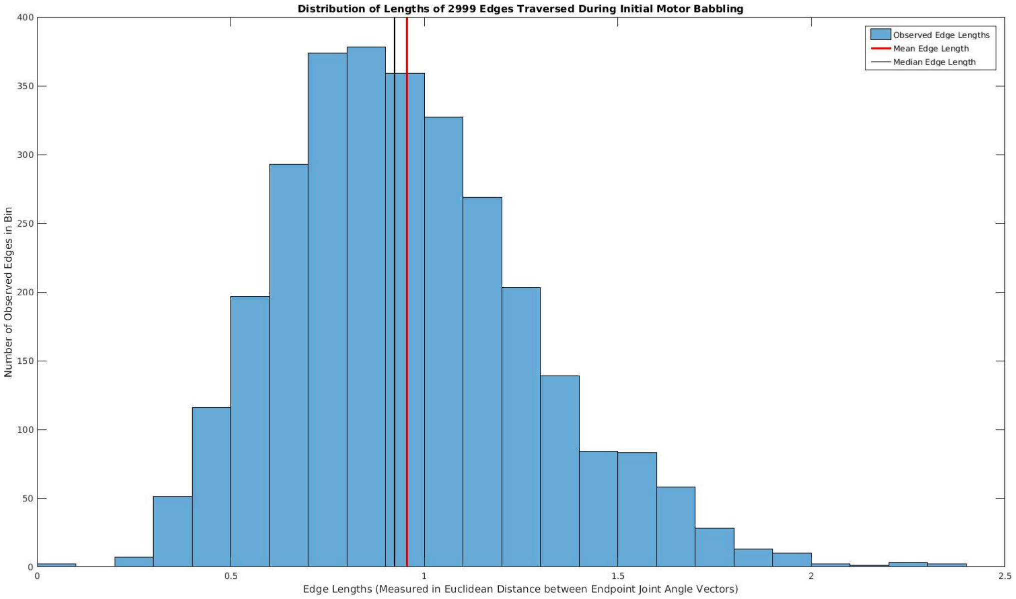


Figure 4.11: A histogram of the distribution of distances traversed with motor babbling motions that defined the first 2999 edges of the graph. Each edge length was calculated according to Equation 4.3, using the distance between the endpoint nodes in configuration space rather than in image space or standard 3D coordinates. The mean and median edge lengths are superimposed at the correct value on the plot's horizontal scale. The median edge length is used in a heuristic to evaluate all pairs of nodes to determine if an edge should be added between them. This heuristic allows 108,718 new edges with lengths less than this median to be added, producing the final state of the Explored PPS Graph.

the vast majority of edges are added due to the heuristic rather than proven to be safe by traversal, a majority of motions will be expected to be performed along these new edges too. Therefore, it is important that the heuristic test is sufficiently strict to keep these edges safe. This helps to justify the use of the median rather than the larger mean or maximum length of the explored edges. In practice, this has been enough to ensure that the edges truly do represent affordances for feasible and safe motions between the stored node positions.

We were initially surprised by the scale of 117,717 edges, but a helpful perspective comes from comparison to the complete graph of 3000 nodes, where each node n_i is connected to every other node n_j. This complete graph has ${}_{3000}C_2 = 4,448,500$ edges, meaning that the Explored PPS Graph $\mathcal{P}$ has the same number of nodes and only about 2% as many edges as the complete, fully-connected graph.

We can also provide a similar perspective in support of the choice of $|\mathcal{N}| = 3000$, an apparently large number of nodes, in a graph that serves as a sparse representation of the agent's peripersonal space. Since the configuration space is seven dimensional (without counting the gripper aperture a, which wasn't changed during the construction of the graph), if the graph had a set of nodes storing configurations $\mathbf{q}$ in an equally-spaced grid instead, each joint would have only $\sqrt[7]{3000} \approx 3.14$ different angle settings available.

There are several additional statistics describing the connectivity of $\mathcal{P}$ that can be examined at this point where $\mathcal{N}$ and $\mathcal{E}$ have been finalized. Specifically, we can create histograms that show the distributions of: the degrees of the nodes (Figure 4.12), the shortest path length between any pair of nodes (Figure 4.13), and the shortest path length between the home node n_h and any other node (Figure 4.14). Each figure's caption gives additional statistics and discussion on the distribution shown.

4.3.4.2 Affordances and Significance

The significance of the PPS Graph can be summarized in two points. First, the graph serves as a general representation of peripersonal space. The characteristics of the graph representation, including the types of information stored in its component nodes and edges, provide one example of a knowledge base sufficient for learning and acting in the agent's nearby environment. The PPS Graph representation can be constructed autonomously by the agent by recording its percepts during motor babbling. Each node of the graph stores an observed pair of a configuration and its associated visual features, and the set of nodes as a whole serves as a discrete mapping between two modalities that describe the agent's peripersonal space (or in this work, a region of the peripersonal space that has been prioritized because it is in view and near or above the workspace surface where objects may be interacted with). While not explored in this work, he ability to visit familiar positions and observe the consistent position of the arm may help to identify it as the self, and by revisiting the

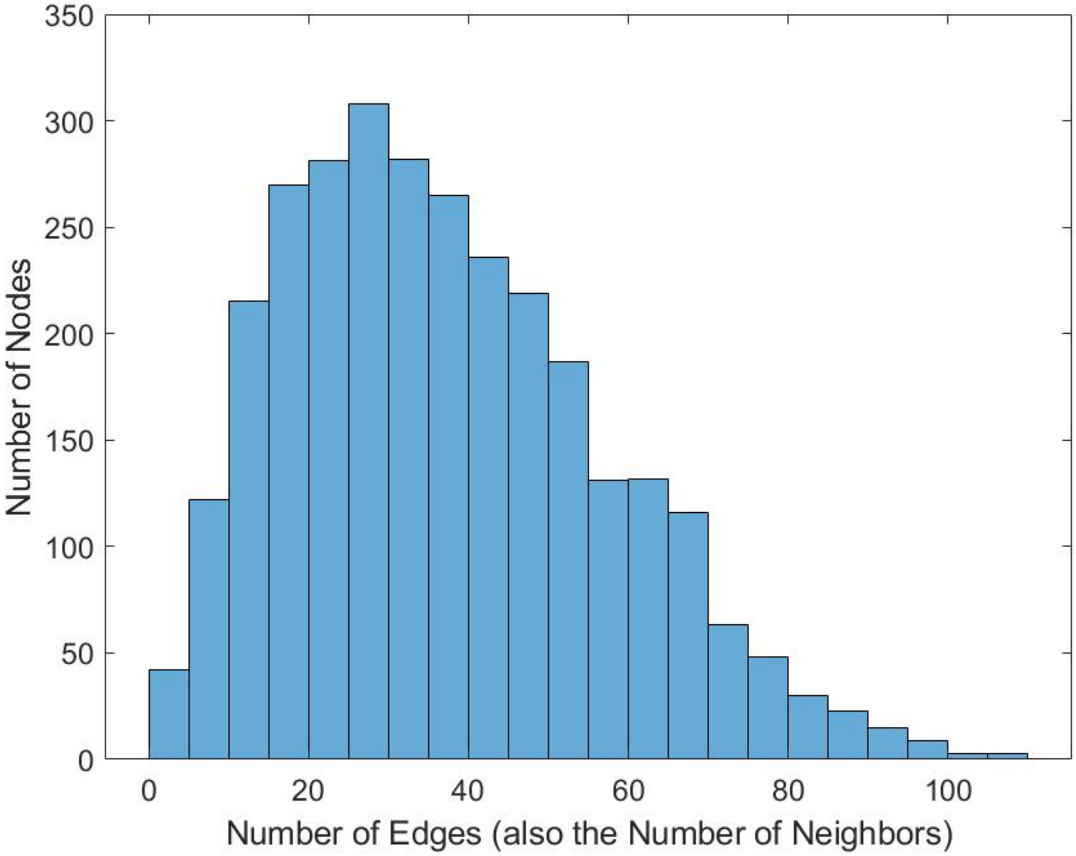


Figure 4.12: A histogram of the distribution of the degrees of each node in the Explored PPS Graph $\mathcal{P}$. The degree of a node is defined by the number of edges connected to the node, and each edge connects a distinct pair of nodes, the degree also indicates the number of neighbors a graph node has. The minimum degree is 1 and the maximum degree is 110. The average degree can be represented with the mean of 36.8, median of 34, and mode of 25. The distribution is skewed to the right, with most node degrees between 10 and 55. Nodes with especially low degrees tend to be near the edges of the graph and only have neighbors on one side, while nodes with especially high degrees tend to be near the center where they can be surrounded with neighbors and where the random walk may have passed nearby many times. As nodes generally have more than one neighbor, the agent has multiple options for moving to almost every node and these can help with obstacle avoidance and alignment of the hand for actions like grasps. Additionally, the maximum degree of 110 is still low enough that it is feasible in time complexity to evaluate properties of every neighbor or incoming edge, which is also important for grasping and any other action that does not always use the shortest graph path trajectory.

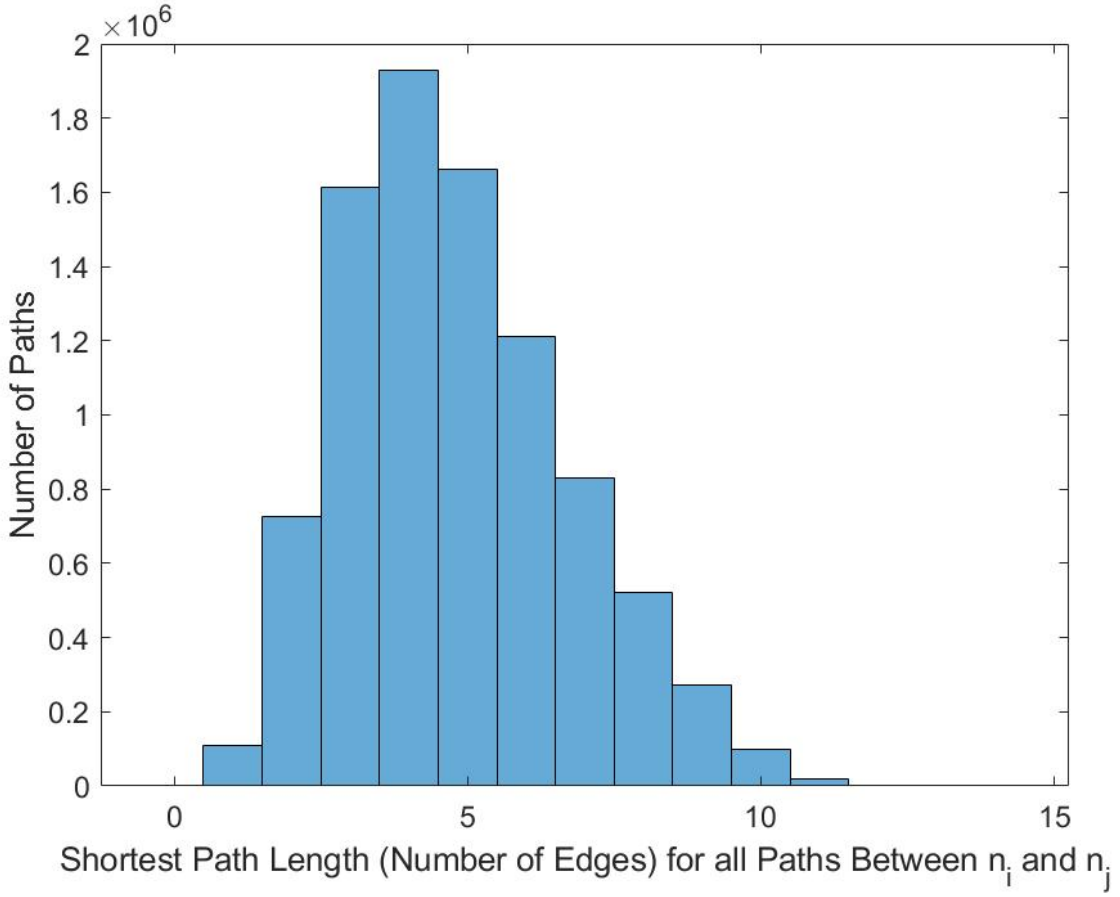


Figure 4.13: A histogram showing the distribution of shortest path lengths, in terms of number of edges between all pairs of distinct nodes n_i and n_j. Note that this is not necessarily the lowest sum of configuration-space edge lengths along the path, and often does not minimize the total movement during the trajectory in image-space or the 3D environment. The mean, median, and mode length for shortest paths are 4.8, 5, and 4, respectively. As expected, the number of length one paths matches the number of edges, 117,717. This distribution is also skewed to the right, with very few exceptionally long paths. The maximum path length is 14, so the diameter of this Explored PPS Graph is also 14.

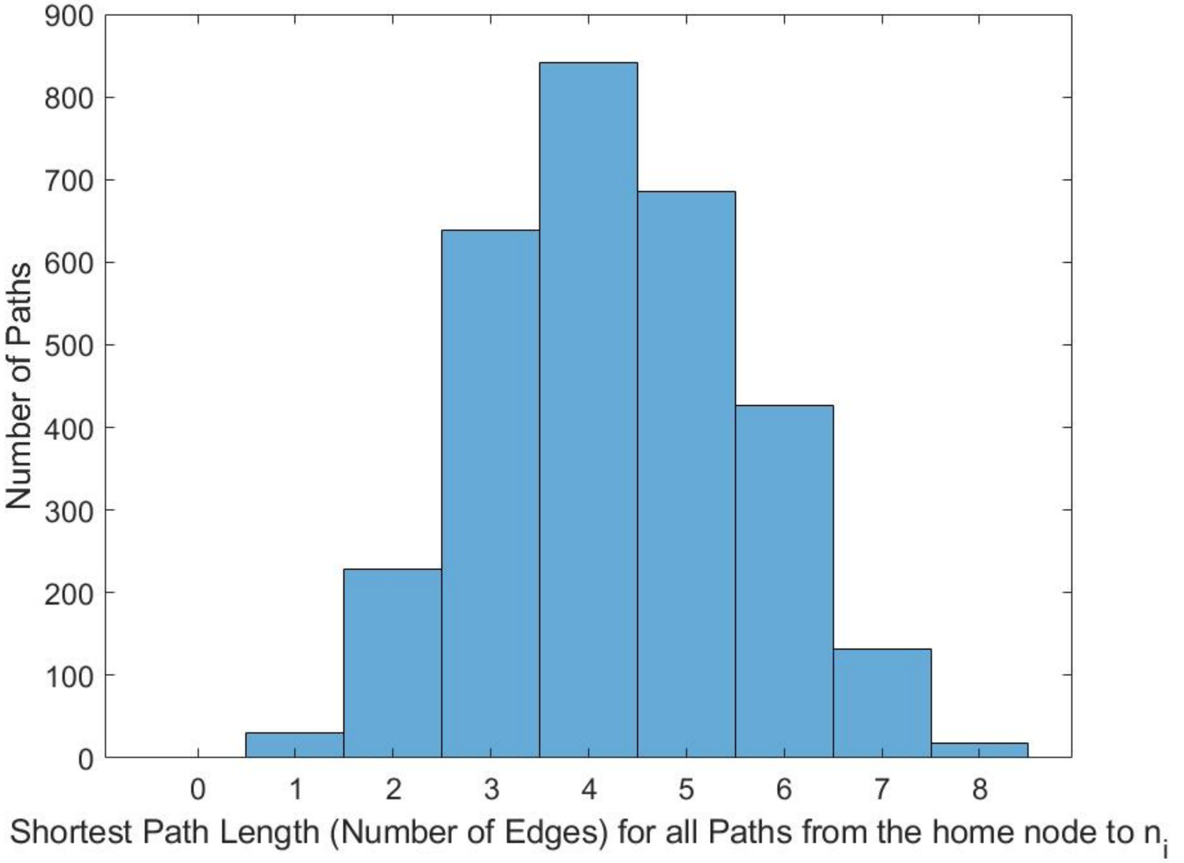


Figure 4.14: A histogram showing the distribution of shortest path lengths between the home node n_h and any other node n_i. The lengths of these paths are generally more influential than the shortest path lengths between pairs of arbitrary nodes shown in Figure 4.13, as the agent begins and ends each pre-reaching move, reach, and grasp trajectory at the home node. Short paths from the home node are beneficial so that the arm can be moved along the path with less pauses for visual observations at each node traveled through, and so that a higher share of the observations that must be processed to check for unusual events like bumps will be the observation at the final node of the trajectory, where the event is more likely to occur once the action is more reliable at producing the event than random motion. The distribution of shortest path lengths from the home node n_h to any other node n_i can be described by the range of 1 to 8 edges in the trajectory, as well as the mean of 4.3, median of 4, and mode of 4.

same familiar position the agent may identify changes in the background that can best be modeled by introducing the concept of nonself objects. Given the ability to recall the visual state for each node, the agent may identify nodes with a desirable feature, such as an occupying a region of space or an intersection with an object to reach or grasp. Because each node may be used to map its desirable visual state to the configuration recorded when the node was first observed, the agent may generate a set of configurations that will reliably produce the desired outcome. When this set includes the configurations from multiple nodes, additional visual features may be used to devise a function to select the one that will be most reliable. The PPS Graph may then be used as "scaffolding" to learn increasingly expert ways to perform actions. In future work, an agent may may be initialized efficiently with the early action policies that can accomplish the goal based on the information in the PPS Graph and learned criteria, and then modify the policy by any method to produce more precise, smooth, and skillful late actions.

The second significance of the completed PPS Graph is the affordance for safe and feasible motions along the edges. While the mechanics of the move arm action remain the same and use the same Joint Position Control mode, the agent can predict the results of moving the joints to a vector of angles that has already been visited, and can verify that this can be performed without the risks of a single large linearly interpolated motion from the starting configuration to the desired set point. Moves that are too large or involve too large of rotation of one or more joints are at risk of being disrupted by the robot's built-in measures to avoid self-collisions, or of causing a collision with the table or other nonself objects in the environment, which would typically require a more sophisticated motion planner and a point cloud or similar model of these obstacles. However, without those requirements, any path $\langle n_1, \ldots, n_m \rangle$ in a PPS graph $\mathcal{P}$ corresponds with a safe trajectory $\langle \mathbf{q}_1, \ldots, \mathbf{q}_m \rangle$ of the arm. In addition to the feasibility of these trajectories, they also have the desirable quality of resembling early infant behavior with the arm. In particular, as the agent moves along each edge and changes direction upon reaching each node in the trajectory, it produces an infant-like sequence of jerky submotions, examples of which can be seen in Figure 4.15. While the set of individual nodes in the graph only provides the information necessary to return to previously visited configurations with predictable results, in the following section we describe how local Jacobian estimates in the neighborhood of each node may be used to extend the graph model from a discrete mapping to a locally linear and continuous mapping. Using these local Jacobian estimates, the agent is able to model the effects of changes in configuration away from any stored configuration, and with the inverse of the local Jacobian estimates the agent may predict the configuration change needed to produce a visual state that was similar to the state for some node, but not observed at any node. These predictions tend to be highly accurate as long as the intended change in configuration and visual state is relatively small, similar in magnitude to the change between a node and one of its neighbors.

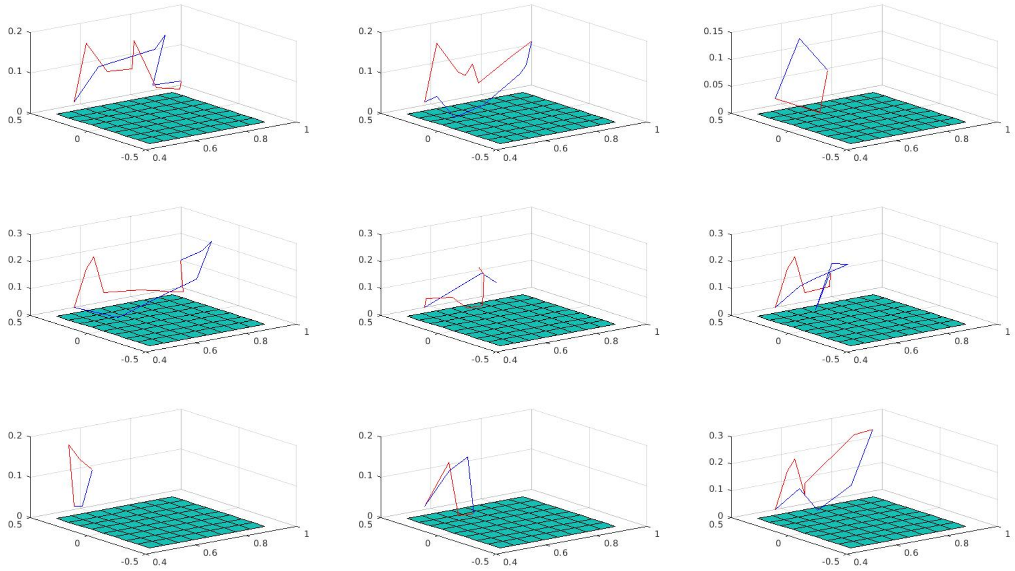


Figure 4.15: An illustration of trajectories along PPS graph edges which feature jerky submotions, and where the direction may change significantly at each node. The trajectories are plotted using the position of each node visited in standard 3D Euclidean coordinates for the environment, calculated from the nodes' stored configurations and the robot's default forward kinematics software that the agent does not have access to. The green plane matches the location of the table's surface and is provided for context. In each plot, the trajectories begin at the agent's home node and end at a randomly selected other node. The red trajectory is the shortest graph path from the home node to the final node when measured in configuration space (which the agent uses) and the blue trajectory is the shortest graph path between the same nodes in terms of distance in image space. Neither of these is necessarily the shortest path in standard coordinates for the environment, which the agent cannot measure or use. As a result, the paths tend to be inefficient, and can be improved to produce smoother and more skilled late action forms of the early actions learned in this work.

4.4 Extending the Graph to a Locally Continuous Model

The PPS graph $\mathcal{P}$ is a discrete, sampled approximation to a continuous mapping between the continuous configuration space of the arm, and a continuous space of perceptual images. The full Jacobian model $J(\mathbf{q})$ relating joint angle changes $\Delta\mathbf{q}$ to changes in hand center image-space coordinates Δc is a nonlinear mapping, dependent on the current state of the arm $\mathbf{q}$, a seven-dimensional vector. The full Jacobian is therefore prohibitively difficult for the agent to learn and use. However, $\mathcal{P}$ does contain sufficient data for making linear approximations of the relationship between $\Delta\mathbf{q}$ and Δc local to a particular $\mathbf{q}_i = Q(n_i)$. This estimate is most accurate near the configuration $\mathbf{q}_i$, with increasing error as the distance from $\mathbf{q}_i$ increases.

The linear approximation at a node n_i is derived using the neighborhood $N(n_i) \equiv \{n_{i'} | \exists e_{i,i'}\}$, the set of all nodes $n_{i'}$ connected to n_i by an edge for feasible motion. The local Jacobian estimate $\hat{J}(n_i)$ considers all edges $e_{i,i'}$ such that $n_{i'} \in N(n_i)$. Each edge provides an example pair of changes $\Delta\mathbf{q} = \mathbf{q}_{i'} - \mathbf{q}_i$ and $\Delta c = c^p_{i'} - c^p_i$. If there are m neighbors, and thus m edges, these can be combined as an $m \times 7$ matrix ΔQ and an $m \times 3$ matrix ΔC, respectively. $\hat{J}(n_i)$ is the least squares solution of

$$\Delta Q \; \hat{J}(n_i) = \Delta C. \tag{4.4}$$

For a given change $\Delta\mathbf{q}$ in arm configuration, $\Delta\mathbf{q} \; \hat{J}(n_i) = \Delta c$ gives a local linear estimate of the resulting change Δc in the appearance of the hand. Conversely, given a desired change Δc in the appearance of the hand, the pseudo-inverse $\hat{J}^+(n_i)$ provides a straightforward computation for the change $\Delta\mathbf{q}$ in arm configuration that is predicted to produce that result.

Figure 4.16 shows an example graph neighborhood and a visualization of the information contained in each edge. The resulting $\hat{J}(n_i)$ is a 7×3 matrix where the element at $[row, col]$ gives the rate of change for c^{col} (either the u, v, or d coordinate of the palm's center of mass) for each unit change to q^{row}. A possible adjustment $\Delta\mathbf{q}$ to $\mathbf{q}_i$ may be evaluated by determining if the predicted new palm center, calculated as

$$\hat{c'}^p_i \equiv c^p_i + \Delta\mathbf{q}\hat{J}(n_i) \tag{4.5}$$

and the palm mask p_i translated by $\Delta\mathbf{q}\hat{J}(n_i)$ have desirable features. Rotations and shape changes of p_i that will occur during this motion are not modeled, but are typically small.

The remainder of this section evaluates the accuracy of predictions made with these estimates. We evaluate the accuracy when calculated as described thus far in Experiment 4.2. In Experiment 4.3, we vary the size of the neighborhood (both for where examples are drawn from and for how distant the points to predict can be) and determine the effect on prediction accuracy. These results will support the use of radius one neighborhoods. Experiment 4.4 evaluates the accuracy of local Jacobian estimates that are formed with only subsets of the joints, and these results suggest a

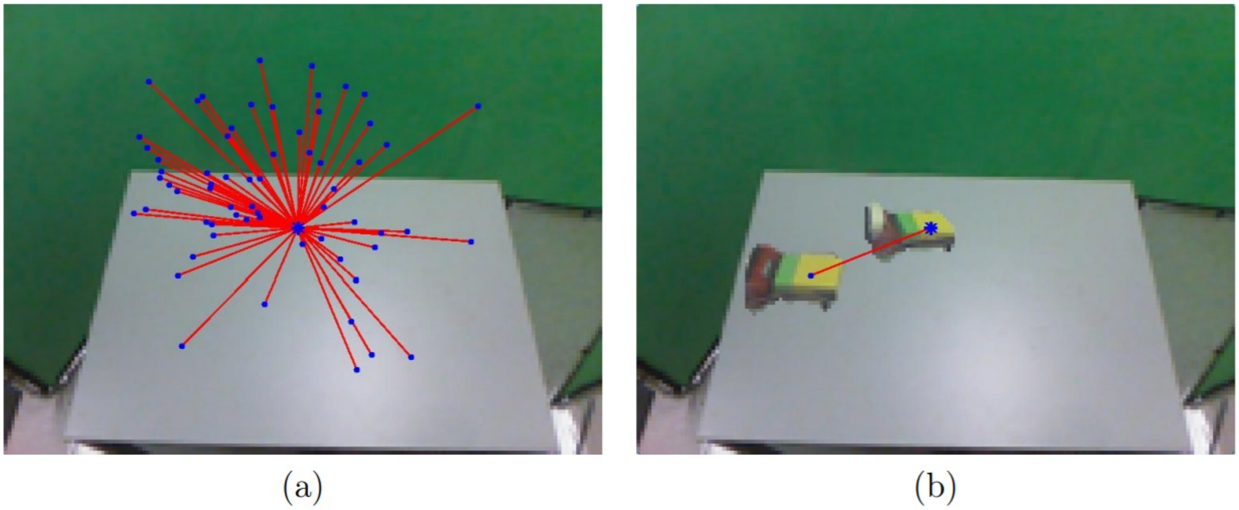


Figure 4.16: **(a)** The agent considers the graph neighborhood around a node n_i to estimate the change in appearance for small changes in configuration near n_i. The predictions will be made by a local Jacobian estimate $\hat{J}(n_i)$ (see equation (4.4)). n_i is near the center of $\mathcal{P}$ and has a large number of neighbors. Each edge is relatively short in configuration space, where edge feasibility is measured, even though some neighbors appear distant in image space. The furthest neighbors tend to be those where most of the edge length comes from a difference in proximal joint angles that have a larger effect on workspace position. **(b)** The images of the node n_i and one of its neighbors are superimposed with a representation of the edge, drawn between their centers of mass. This example illustrates a change in configuration $\Delta\mathbf{q}$ and the resulting change in center locations Δc along one edge.

possible explanation for the proximal to distal bias in joint usage by infants.

4.4.1 Experiment 4.2: Evaluating the Accuracy of the Local Jacobian Estimates

Each $\hat{c}'^p_i$ predicted with a Local Jacobian estimate $\hat{J}(n_i)$ according to Equation 4.5 can be checked for accuracy by finding the error between this prediction and the true c'^p_i reached when the agent moves from the configuration $\mathbf{q}_i$ by $\Delta\mathbf{q}$. This error can be calculated in a straightforward manner by

$$||\hat{c}'^p_i - c'^p_i||_2. \tag{4.6}$$

Since the errors are distances between predicted and actual image-space centers, they are measured in pixels by pixels by disparity units, though the agent is not aware of these units. The agent treats the error as a 3D vector with unspecified units, and finds the magnitude without consideration for the different units and scales.

In order to evaluate the accuracy of the Local Jacobian estimates computed by the agent, it is necessary to determine the error between the image-space coordinates predicted using an estimate and the actual coordinates reached after performing a move. One option is if for each node n_i, several perturbations $\Delta\mathbf{q}$ are generated, and the resulting $\hat{c}'^p_i$ is predicted. Then the agent could apply the perturbation $\Delta\mathbf{q}$ to the joints and capture a new image, determining the actual new center position c'^p_i so that the error can be calculated as in Equation 4.6. However, because it will be necessary to test multiple perturbed configurations for all 3000 nodes $n_i \in \mathcal{N}$ in order to evaluate the general accuracy of these predictions, the additional motions would require an undesirable amount of time.

Fortunately, the information stored in the PPS Graph provides an alternative source of perturbed configurations. Instead of sampling small $\Delta\mathbf{q}$ to produce $\mathbf{q}' \in N_{\mathbf{q}}(n_i)$, the continuous configuration space near $\mathbf{q}_i$ defined as

$$N_{\mathbf{q}}(n_i) \equiv \{\mathbf{q} \mid \{\exists n_j \in N(n_i) \mid ||\mathbf{q} - \mathbf{q}_i||_2 \leq ||\mathbf{q}_j - \mathbf{q}_i||_2\}\} \tag{4.7}$$

the agent can use $\mathbf{q}_j$ from each $n_j \in N(n_i)$, the alternative neighborhood definition that contains only the configurations stored in nodes connected to n_i with edges,

$$N(n_i) \equiv \{n_j \mid \exists e_{i,j}\}. \tag{4.8}$$

Since $n_j \in N(n_i) \implies \mathbf{q}_j = Q(n_j) \in N_{\mathbf{q}}(n_i)$, each $\mathbf{q}_j$ can be used instead of a randomly sampled $\mathbf{q}'$. Since the predictions will be made without considering the stored c^p_j, and simply by substituting $\Delta\mathbf{q} = \mathbf{q}_j - \mathbf{q}_i$ into Experiment 4.5, the use of a previously visited configuration instead of a random new configuration does not affect the evaluation of the mean prediction accuracy. Using the neighboring nodes' configurations also has the key benefit that the error can be calculated with Equation 4.6 without additional motion, since c_j^p has already been observed.

4.4.2 Experiment 4.2 Results

Local Jacobian estimates $\hat{J}(n_i)$ were calculated for all nodes n_i that have defined palm centers c_i^p and that have at least seven neighbors, which allows the least squares solution to be computed. In this implementation of the Explored PPS Graph, there are 2946 such nodes.

Individual predictions $\hat{c}_j^p$ have an error given by $||\hat{c}_j^p - c_j^p||_2$, a version of Equation 4.6 that has been customized for the context of using stored nearby configurations instead of random ones. To express the quality of a specific local Jacobian estimate $\hat{J}(n_i)$, we take the mean of these errors for all n_j with defined palm centers (poses where the palm was visible and has at least one valid depth reading),

$$\frac{1}{|N(n_i)|} \sum_{j|n_j \in N(n_i)} ||\hat{c}_j^p - c_j^p||_2. \tag{4.9}$$

In addition to reporting the mean for the predictions made in the neighborhood of individual nodes, we can aggregate these mean errors to determine the distribution of prediction qualities overall when local Jacobian estimates calculated in this way are used. In this case, the range of mean errors was $[1.27 \times 10^{-14}, 14.49]$. The minimum mean error comes from a node that was extremely close to predicting the correct palm center position for every node in its neighborhood, while the maximum mean error was almost three times larger than average. The full distribution of mean errors had a mean of 5.306 and a standard deviation of 1.50. While the d component of the error is a difference in disparity and less intuitive, thinking of this error in terms of a distance in pixels, as it is in the u and v dimensions, demonstrates that the predictions are quite close to the actual positions. When the agent is attempting to move the hand to desired positions or regions, including for use in reaching or grasping actions where the desired region is determined by a target object, this error tends to be small compared to the size of the desired region. It is important to remember though that each $\hat{J}(n_i)$ is a linear approximation, and only intended to be used to predict coordinates close to n_i, and will become more inaccurate quickly with extrapolation beyond the local region where the tangent is close to the true Jacobian surface near n_i.

4.4.3 Experiment 4.3: Estimating the Local Jacobian with Different Neighborhood Sizes

The definition of the neighborhood $N(n_i)$ above assumes that neighborhoods have a radius of one, meaning that neighbors of n_i are only those nodes n_j that can be reached within a shortest path length of at most one, requiring an edge $e_{i,j}$ to exist. However, it is also possible to determine if this is the optimal neighborhood size, or if a different size produces local Jacobian estimates that allow more accurate predictions $\hat{c}_j^p$.

In order to evaluate different neighborhood sizes, we use a more general definition of the set of nodes included in the neighborhood of n_i to allow the radius to change. Assuming $s(n_i, n_j)$ evaluates to the length of the shortest graph path from n_i to n_j (measured in the number of edges, not in total configuration space or image space distance traveled), let the neighborhood of radius R for n_i be determined by

$$N_R(n_i) \equiv \{n_j \neq n_i \mid s(n_i, n_j) \leq R\}. \tag{4.10}$$

Note that $N_1(n_i)$ is equivalent to the existing definition of $N(n_i)$ from Equation 4.8.

Neighborhoods $N_R(n_i)$ have two roles in this experiment. First, all nodes $n_j \in N_R(n_i)$ provide an observed example of a perturbation $\Delta\mathbf{q}$ from the node's configuration $\mathbf{q}_i$ and the corresponding Δc, the change in image-space coordinates of the hand's center. In this way $N_R(n_i)$ serves as the training neighborhood that allows the local Jacobian estimate $\hat{J}(n_i)$ to be computed. Second, the agent makes a prediction $\hat{c}_j^p$ for the center of each node $n_j \in N_R(n_i)$ based on the difference $\mathbf{q}_j - \mathbf{q}_i$ and $\hat{J}(n_i)$, and determines the mean error of all predictions made for each node n_i. In this way $N_R(n_i)$ serves as the evaluation neighborhood, and as its size changes the range of distances at which the accuracy of the locally linear estimate $\hat{J}(n_i)$ is evaluated.

When experimenting with different neighborhood sizes, the size of the training neighborhood and evaluation neighborhood can be modified separately. That is, the training neighborhood may be defined as $N_{R_{train}}(n_i)$ and the evaluation neighborhood as $N_{R_{eval}}(n_i)$, with R_{train} not necessarily equal to R_{eval}. We evaluate the mean accuracy with all combinations of (R_{train}, R_{eval}) for $R_{train}, R_{eval} \in [1, 14]$. This range contains all meaningful radius values, as the minimum radius that allows nodes to have any neighbors is $R = 1$, and once the radius is equal to the diameter of the graph at $R = 14$, all 2999 other nodes are already part of each node's neighborhood, and further increases in R will have no effect. This evaluation was also carried out entirely without further motion, as all necessary information on configurations $\mathbf{q}$, image-space palm centers c^p, and adjacency is stored in the nodes and edges of the PPS Graph.

4.4.4 Experiment 4.3 Results

The mean errors produced by each local Jacobian estimate $\hat{J}(n_i)$, which are themselves means of the error of each prediction $\hat{c}_j^p$ made using the estimate, are shown in Figure 4.17. These values are presented in a heatmap to better visualize the effects of changing R_{train} and R_{eval}. We observe the minimum average error when $R_{train} = R_{eval} = 1$, supporting our definition of $N(n_i)$ and initial method of calculating $\hat{J}(n_i)$ as they provide the most accurate model of the relationship between configuration space and image space around the observed $(\mathbf{q}_i, c_i^p)$ pair of a node n_i.

We can draw additional conclusions from Figure 4.17 by comparing the values when different radii are used. Increasing only the size of the training neighborhood slowly increases the error magnitudes. All training data is still valid and centered around the same node, so it provides useful, but slightly less specific information. Surprisingly, even global training data with $R_{train} = 14$ so that all other nodes are used as examples can be used to produce relatively accurate linear approximations for local use. While less than triple the error of the best case with $R_{train} = 1$, this will be less desirable as an error of this magnitude in image space may cause the agent to fail to reach a desired region or object.

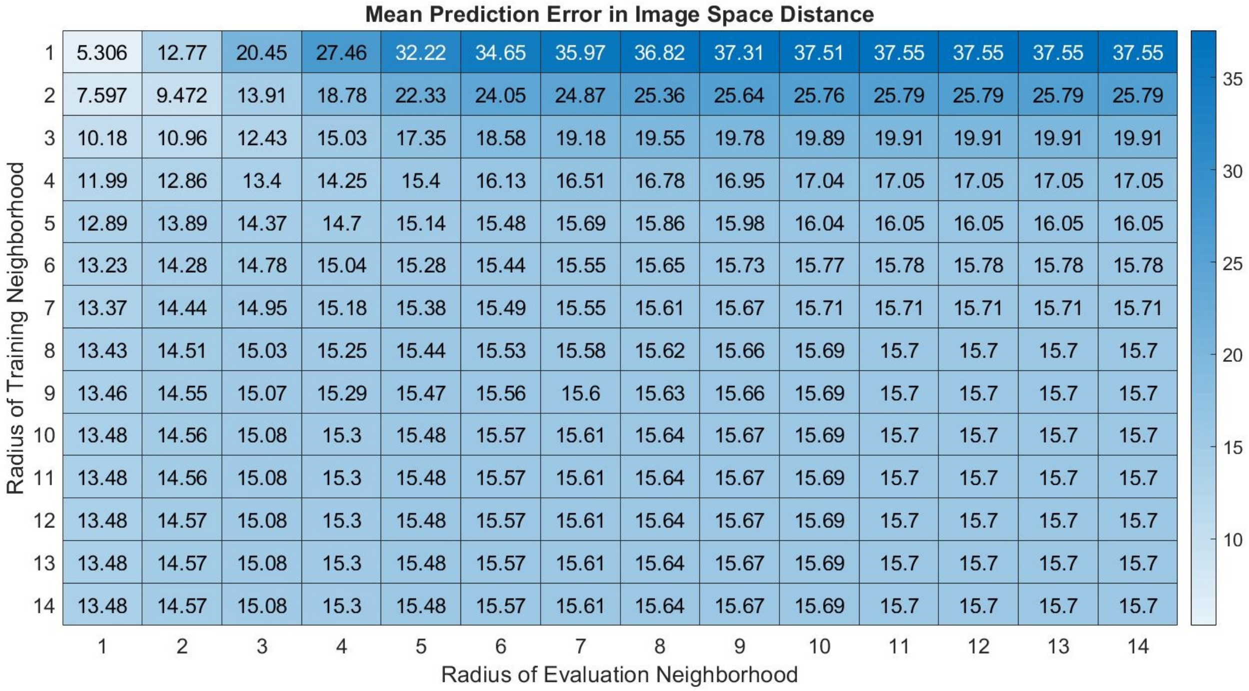
Increasing only the size of the evaluation neighborhood increases the error magnitudes quickly, as the agent is asked to extrapolate further and further past the distance where the examples it was trained on can be found. The highest mean error is associated with $R_{train} = 1$ and $R_{eval} = 14$. The extrapolation error is more pronounced for the worst case node than the average, with several nodes with more than an order of magnitude higher error than the mean. Given the 120×160 size of the stored images, these worst predictions are of center positions that would fall well out of view of the camera.

When both the training and evaluation neighborhood sizes are increased together, the mean error has a moderate increase. If the predictions will need to be over a larger region of space, it is best to take examples from that larger region as well. Still, the lesser fit of a linear approximation at longer distances makes the errors higher than they are for the $(1, 1)$ neighborhoods.

A final note should be made about the bottom right corner of the heatmap, where the same error values are observed in multiple cells. While the diameter of the graph is 14, most nodes will have all or very nearly all nodes in their neighborhood with a smaller radius. As seen in Figure 4.13, it is very uncommon for the shortest path length $s(n_i, n_j)$ to be greater than 10, so almost all nodes have the same neighborhoods for training or evaluation once $R > 10$, and larger radii yield the same predictions and errors.

4.4.5 Experiment 4.4: Investigation of Proximal to Distal Bias

One of the strengths of the PPS Graph model is its capability to facilitate the learning and execution of actions that are consistent with observed infant behaviors and the hypothesized causes of these behaviors. Notably, PPS Graph trajectories are composed of jerky submotions [49], and two potential explanations - the visiting of stored configurations as set way points on the way to the goal pose and underdamping of the arm - can be investigated using their implementation in the graph model and the control law for Joint Velocity Control. We also have the example of the agent not relying on current vision of the hand. This has been observed by developmental psychologists since infants can reliably reach for lit objects in a dark room [11], and our agent's trajectory planning



Mean Prediction Error in Image Space Distance

Radius of Training Neighborhood \ Radius of Evaluation Neighborhood	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	5.306	12.77	20.45	27.46	32.22	34.65	35.97	36.82	37.31	37.51	37.55	37.55	37.55	37.55
2	7.597	9.472	13.91	18.78	22.33	24.05	24.87	25.36	25.64	25.76	25.79	25.79	25.79	25.79
3	10.18	10.96	12.43	15.03	17.35	18.58	19.18	19.55	19.78	19.89	19.91	19.91	19.91	19.91
4	11.99	12.86	13.4	14.25	15.4	16.13	16.51	16.78	16.95	17.04	17.05	17.05	17.05	17.05
5	12.89	13.89	14.37	14.7	15.14	15.48	15.69	15.86	15.98	16.04	16.05	16.05	16.05	16.05
6	13.23	14.28	14.78	15.04	15.28	15.44	15.55	15.65	15.73	15.77	15.78	15.78	15.78	15.78
7	13.37	14.44	14.95	15.18	15.38	15.49	15.55	15.61	15.67	15.71	15.71	15.71	15.71	15.71
8	13.43	14.51	15.03	15.25	15.44	15.53	15.58	15.62	15.66	15.69	15.7	15.7	15.7	15.7
9	13.46	14.55	15.07	15.29	15.47	15.56	15.6	15.63	15.66	15.69	15.7	15.7	15.7	15.7
10	13.48	14.56	15.08	15.3	15.48	15.57	15.61	15.64	15.67	15.69	15.7	15.7	15.7	15.7
11	13.48	14.56	15.08	15.3	15.48	15.57	15.61	15.64	15.67	15.69	15.7	15.7	15.7	15.7
12	13.48	14.57	15.08	15.3	15.48	15.57	15.61	15.64	15.67	15.69	15.7	15.7	15.7	15.7
13	13.48	14.57	15.08	15.3	15.48	15.57	15.61	15.64	15.67	15.69	15.7	15.7	15.7	15.7
14	13.48	14.57	15.08	15.3	15.48	15.57	15.61	15.64	15.67	15.69	15.7	15.7	15.7	15.7

Radius of Training Neighborhood
Radius of Evaluation Neighborhood
35
30
25
20
15
10

Figure 4.17: Each value shown in this heatmap is the mean over the local Jacobian estimates $\hat{J}(n_i)$ of all nodes n_i of the mean error values calculated by Equation 4.9 in the predicted image-space coordinates $\hat{c}_j^p$. Each mean of mean errors shown is calculated using a particular combination of training and evaluation neighborhood radii. The radius R of a neighborhood $N_R(n_i)$ is the maximum shortest path length (in number of edges) from n_i to a node n_j for n_j to be included in the neighborhood. The minimal mean of mean errors value observed with the (1,1) neighborhoods justifies the use of radius one neighborhoods for all future calculation of local Jacobian estimates and as a guideline for the region of space where the estimates can be used to produce reliable predictions.

has the same property because only stored visuals of the hand are needed to plan where it will be moved.

The ability to compute local Jacobian estimates from neighborhoods in the PPS Graph allows us to perform a preliminary investigation of a potential cause for the proximal to distal joint bias observed for most typically developing infants[41]. Infants near reach onset tend to move the arm primarily with the shoulder joint (or by leaning at the waist when this even more proximal degree of freedom is unrestricted), with little utilization of the elbow or wrist to adjust the hand until later stages. Our current model does not include this bias, but an analysis of the accuracy of modifiedd local Jacobian estimates suggests a theoretical explanation for why it may be present.

While the linear approximation $\hat{J}(n_i)$ is much less complex than the function for the true globally applicable $J(\mathbf{q})$, calculating it involves solving for the least squares solution in Equation 4.4. While this calculation is domain-general, and could possibly be carried out by a neural process specially suited to estimating trends from observed examples, it is still a large operation to be carried out by an infant's mind. If we hypothesize that computing the full 7×3 matrix of values for $\hat{J}(n_i)$ is prohibitively difficult, then an alternative would be for the agent to only consider the changes in a subset of joints for the estimate it will construct and use for predictions. If only k joints are considered and m is the number of neighboring nodes in $N(n_i)$, then ΔQ will be an $m \times k$ matrix and ΔC will remain an $m \times 3$ matrix. As long as $k < 7$, the computation of the $k \times 3$ matrix $\hat{J}(n_i)$ will be less complex.

In this experiment, we evaluate the mean of all mean prediction error values from $Equation\ 4.9$ using the standard neighborhoods $N(n_i) = N_1(n_i)$ and all possible nonempty subsets of joints considered. In particular, we will want to compare the magnitude of prediction errors made by local Jacobian estimates that considered more proximal joints to the magnitude of prediction errors made by local Jacobian estimates that considered the same number of joints or more, but more distal joints. If predictions are more accurate when proximal joints are considered than when a same size or larger set of distal joints are considered, this will support a hypothesis that a bias to carry out motions with the proximal joints allows the motion to be more accurately modeled and performed.

4.4.6 Experiment 4.4 Results

The mean errors for predictions made with $\hat{J}(n_i)$ calculated with all possible nonempty subsets of joints considered are shown in Figure 4.18. We observe that for all numbers of joints k, the most accurate local Jacobians all consider a set of joints including s0, the robot's most proximal arm joint. As k increases, the joints are added to the optimal considered set in roughly proximal to distal order, with elbow joints being included before wrist joints. We also note from these results that the order in which joints should be included is also consistent across k values. Once a joint is added to consideration by the optimal set of size k_1, it remains in all sets of size $k_2 \geq k_1$. The order is not in perfect proximal to distal order, and we offer two explanations for the disparity. In particular, after $s0$, we see that $e1$ and then $e0$ are added before the second shoulder joint $s1$, which should come after $s0$ in proximal to distal order. The first explanation is that because the agent is predicting the displacement of the center of the hand when all joint angles changed (not just the joint angles considered by the particular local Jacobian estimate), the benefit of choosing a second shoulder joint because it is more proximal and larger is outweighed by the benefit of selecting elbow joints that provide more additional unique information about the motion made with the full arm. Of the two elbow joints, $e1$ offers the most unique information as the second joint because in addition to being an elbow joint instead of a shoulder joint, it is also a bend joint instead of a twist joint. While this appears to be an important factor, the benefit of choosing more proximal joints with influence over a larger portion of the arm is still considerable enough that all four shoulder and elbow joints are added to the optimal set before any of the three wrist joints are added. The second explanation is that for the particular joint angle ranges allowed for the motor babbling to build the graph and for the camera perspective used, $e1$ may have had an unusually large influence on the position of the hand, allowing it to be more valuable to consider than $s1$ and $e0$.

While we have observed a justification for a roughly proximal to distal ordering of joint usage for the robot's arm, we can also evaluate the results for sets of joints grouped in a manner that resembles a human arm, with a shoulder, elbow, and wrist. For these experiments, rather than selecting individual joints for consideration, a subset of the groups {shoulder, elbow, wrist} is considered. When the shoulder group is included, the estimate will be calculated with a set of joints including $s0$ and $s1$, and similarly the elbow group will have $e0$ and $e1$ and wrist group will have $w0$, $w1$, and $w2$. Full results and analysis are given in Figure 4.19. With these joint groupings, the optimal order for inclusion is perfectly proximal to distal, and supports our hypothesis about the potential role of the local Jacobian estimate in producing motions carried out with a proximal to distal joint bias. In future work, additional evaluations can be performed to continue the investigation of this theory.

4.5 Conclusion

The focus of this chapter has been to construct an instance of the PPS Graph model that was formalized in Section 3.2.1, and then to use the knowledge stored in the completed graph as a mapping between proprioceptive and visual states and to facilitate safe motion of the arm. We present the procedure and statistics for the Explored PPS Graph rather than another variation as it was the implementation used in Chapter 5 for reaching, Chapter 6 for grasping, and Chapter 7 for placing.

As a precursor to the motion of the arm that allows the graph to be constructed, we consider the Baxter Research Robot's built in control modes in Section 4.2. We found that Joint Velocity

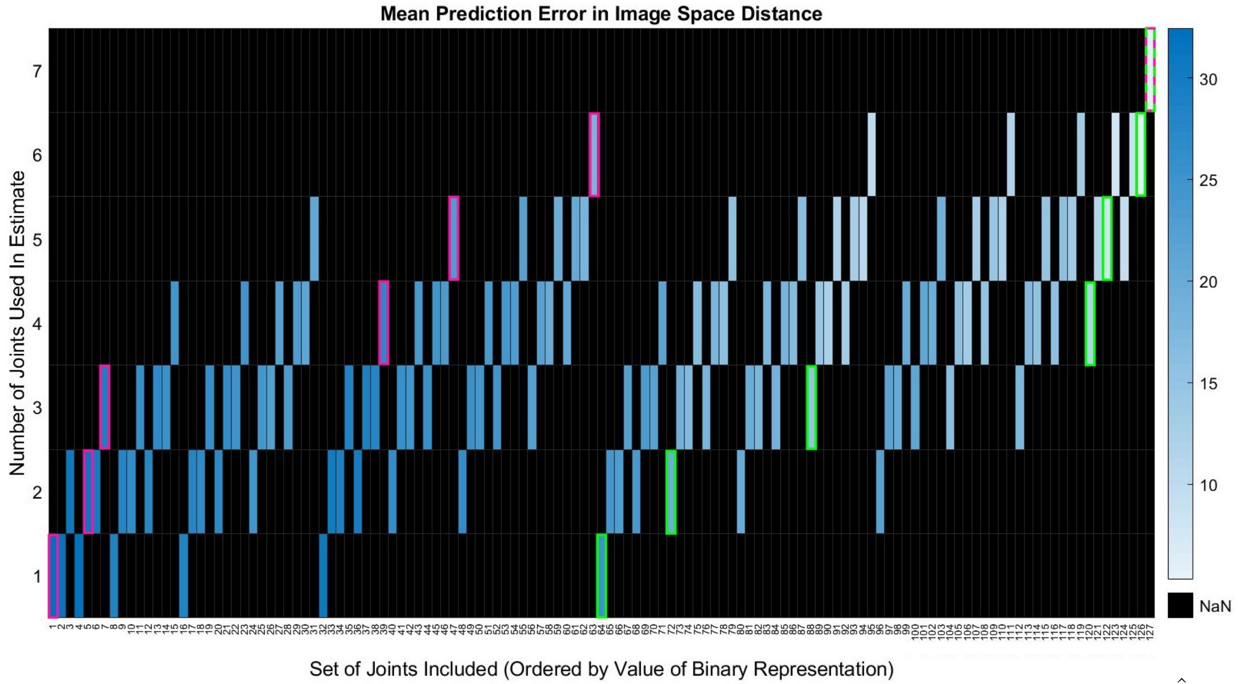


Figure 4.18: A heatmap of the image-space prediction errors for all local Jacobian estimates $\hat{J}(n_i)$, when those estimates are calculated considering only a subset of joint values. Each error shown is the mean of the mean prediction errors in the neighborhood of individual nodes (the mean of Equation 4.9 over all nodes n_i). To read this figure, the row of the heatmap indicates the number of nodes considered when calculating the local Jacobian estimates and center position predictions, and the column number can be converted to a binary representation of the specific joints included. For example, column 74 corresponds to 1001010_2, which indicates s0 is included, s1 is not, e0 is not, e1 is, w0 is not, w1 is, and w2 is not, or the set of joints $\{$s0,e1,w1$\}$. This set contains three joints, so column 74 has the mean error displayed in row 3, and NaN is displayed in all other rows because the number of joints does not match the set. The joint sets that produce the local Jacobian estimates with the most accurate predictions (lowest error) in each row are highlighted with a green border. In increasing size and decreasing error, these optimal sets are $\{$s0$\}$, $\{$s0, e1$\}$, $\{$s0, e0, e1$\}$, $\{$s0, s1, e0, e1$\}$, $\{$s0, s1, e0, e1, w1$\}$, $\{$s0, s1, e0, e1, w0, w1$\}$, and the complete set of joints $\{$s0, s1, e0, e1, w0, w1, w2$\}$. The order in which the joints should be included to have the least error with the simplest model is roughly proximal to distal, with shoulders tending to come before elbows and both of those before wrists. At the extremes, the most proximal s0 should always be included and the most distal w2 should only be included if all joints are being used. The joint sets that produce the highest errors, and should be avoided if possible, are highlighted with a pink border. These sets have joints added in nearly the opposite order, and are $\{$w2$\}$, $\{$w0, w2$\}$, $\{$w0, w1, w2$\}$, $\{$s1, w0, w1, w2$\}$, $\{$s1, e1, w0, w1, w2$\}$, $\{$s1, e0, e1, w0, w1, w2$\}$, and the full set of joints $\{$s0, s1, e0, e1, w0, w1, w2$\}$. If the agent is not capable of modelling the relationship between all joint angle changes and the change in image space position, the most accurate alternative is to use as many joints as it can perform the calculation for, and for these joints to tend to be proximal. These results are consistent with a hypothesis that proximal to distal bias in joint use may be an effort to perform motions with only the subset of joints included in a simplified mapping between configuration space and image space, such as, but not necessarily equal to, these local Jacobian estimates calculated with only a subset of joints.

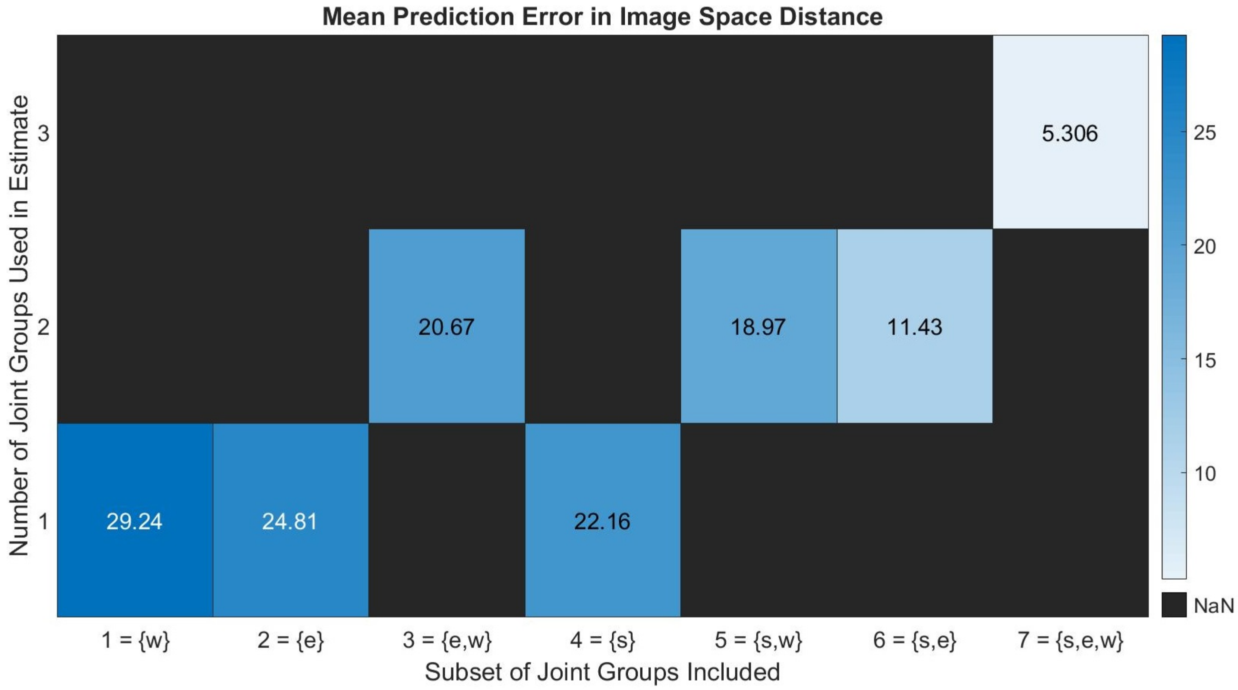


Figure 4.19: The mean of the mean prediction errors of local Jacobian estimates that are computed using only a subset of joints. In this case, joints are included or excluded as a set that resembles the role (but not structure) of the shoulder, elbow, or wrist of the human arm. The optimal order for inclusion of the groups is perfectly proximal to distal. When all groups are included, the estimate matches the one from Experiment 4.2, and the mean error value is the same. This is the most accurate model, but when only a subset of groups can be included, the shoulder should always be chosen. It is also valuable for the agent to consider as many groups as it can perform the computation with, as increasing the number of groups from one to two always improves the prediction accuracy, even if it is {e,w} instead of {s}. Even though the wrist group includes information from three joints instead of the two joints of the other groups, w should always be included last, and the estimate based on only w is the least accurate.

Control will be important for future work due to its flexibility to customize parameters that allow additional theories to be modeled and evaluated. In particular, we have already demonstrated that motion qualities with different levels of damping may be observed by changing the coefficients for the control law that calculates velocities to command. However, the higher level of abstraction, more detailed documentation, and simplicity of use led us to carry out all experiments in this work after the control mode comparison with Joint Position Control, which only requires joint angle set points to be specified. It is possible to construct graphs with sets of nodes that store nearly identical configurations and visual states with either control mode, and we claim that consistent results will be obtained if there is a switch to Joint Velocity Control after the experiments in this work or if all experiments are repeated with Joint Velocity Control. Both of these control modes reliably bring each joint angle to a goal position, and when the joints are moved in parallel the full arm can be moved, as a first iteration of the move arm action.

In Section 4.3, we discuss how the capabilities of the built-in Joint Position Control mode alone are insufficient for the move arm action. While the built-in controller handles low-level control precisely and can linearly interpolate from one configuration to another, it cannot verify if this motion will be safe, and it cannot predict the result of this motion and if it will be desirable for the agent. This motivates the construction of the Explored PPS Graph, which is built by recording the joint angle vectors sensed by proprioception and the image data from a camera at 3000 poses the robot visits while motor babbling with the arm. In addition to providing the agent a guarantee of safe, albeit infant-like motion, by moving along graph trajectories, the stored pairs of configuration and visual features in each node will allow the agent to plan reliable moves for manipulation actions in the following chapters.

In Section 4.4, we demonstrate a method by which the agent can extract information from a node and its neighbors to estimate how changes in configuration space near the node will change a property in image space, in this case the coordinates of the center of the palm region of the hand. These local Jacobian estimates are linear approximations that are valid near each node, and allow the mapping between configuration $\mathbf{q}$ and palm center c^p near the node locally continuous. Compared to the discrete and in some places sparse mapping provided by the stored pairs in the nodes, this is a much fuller representation of the peripersonal space. The ability to model and use motions near but not directly to the nodes of the PPS Graph will also grant the agent necessary precision for actions that strongly require it, such as grasping in Chapter 6.

In Chapter 5, the agent will use the ability to move the arm to observe the typical result of its appearance changing in a static background, and then an unusual result where a part of the background also changes - because it is a foreground object that has been bumped. While these bump events are rare, the graph supports repeated motions throughout the space, and enough will eventually be observed to discover conditions shared between each case. The agent will also utilize the knowledge gained from the building of the Explored PPS Graph to identify nodes with positions in image space that will be reliable for reaching to objects to repeat these bump events. The nodes will also provide the mapping to the configuration for this pose, and the edges that connect the PPS Graph will provide a trajectory that can be used to move to that configuration. The ability to move the arm and to use the information in the PPS Graph will continue to be used in Chapter 6 when reaching is refined into a more specific grasping action, and in Chapter 7, when the hand must be moved along with a grasped object to place it into a desired location with an ungrasp.

CHAPTER 5

Learning a Reliable Reach Action

In our model, learning the reach action takes place in three stages. First, the agent must learn to detect the unusual event of bumping a block, causing a quasi-static change in the environment, against the background of typical arm motions that leave the environment unchanged. Second, the agent learns criteria for selecting nodes from the PPS graph, such that moving to one of those nodes increases the likelihood of bumping a specified block. Third, the agent learns how to interpolate in continuous space between the nodes of the PPS graph to further increase the likelihood of bumping a target block.

Since these three learning stages have different character, depend on different knowledge, and apply different methods, we describe our research on each of them with its own Methods-Experiments-Results description.

5.1 Observing the Unusual Event of a Bump

5.1.1 Methods

During the construction of the PPS Graph, the agent's perceptual input can be easily factored into a static background, and a highly variable foreground corresponding to the robot's hand and arm. This allows the nodes of the PPS Graph to be characterized by the perceptual image of the robot's hand. By detecting a correlation between "random" motor output and perceived hand motion, the agent diagnoses that the hand is part of the agent's "self".

Once the PPS Graph has been completed, additional objects are placed into the workspace. The objects used for this work are rectangular prism blocks with a single long dimension. The blocks are placed upright at randomly generated coordinates on the table in front of the robot, with the requirement that each placement leaves all blocks unoccluded and fully within the field of vision. The objects have distinctive colors not present in the background, making it easy to create a binary image mask for each object in the RGB image. This image mask can be applied to the depth image to determine the range of depth values associated with the object.

The agent creates binary image masks as more efficient representations of its own hand and of foreground objects that may be targets of actions. Recall from Section 3.2.1.2, for each $n_i \in \mathcal{P}$, the agent finds the end effector in $I_{RGB}(n_i)$ and records two binary masks that describe its location in the image. The *palm mask* p_i is defined to be the region between the gripper fingers, which will be most relevant for grasping.[1] The *hand mask* h_i includes this region as well as the gripper fingers and the wrist near the base of the hand. h_i reflects the full space occupied by the hand, which is most useful to identify and avoid nodes with hand positions that may collide with obstacles. The state representation for a node also includes the range of depths the end effector is observed to occupy. This range is found by indexing into $I_D(n_i)$ with either mask, and determining the minimum and maximum depth values over these pixels. That is, the depth range of the palm $D(p_i) \equiv [\min(I_D(n_i)[p_i]), \max(I_D(n_i)[p_i])]$, and the depth range of the full hand $D(h_i)$ is defined analogously. Edges can also be associated with a binary mask for the area swept through during motion along it, $s_{i,i'}$, approximated by a convex hull of the hand masks of the endpoint nodes, h_i and $h_{i'}$. While similar to the true path of the hand through the image, $s_{i,i'}$ will often be imperfect due to the tendency of the rotational joints to create paths that appear curved in image space and workspace, rather than straight as the convex hull suggests. The depth range of motion along an edge is the full range between the minimum and maximum depths seen at either endpoint, $D(s_{i,i'}) \equiv [\min(D(h_i), D(h_{i'})), \max(D(h_i), D(h_{i'})]$. It is possible for either the minimum or maximum disparity value to occur in the middle of the middle of the edge rather than at the endpoint nodes, so $D(s_{i,i'})$ may underestimate the full range of depths passed through during the motion.

Many (but not all) motions of the arm leave the other objects unaffected, so the new objects typically behave as part of the static background model. However, occasionally the hand bumps into one of the objects and knocks it over or shifts its position. This is defined as a *bump* event, and is detected by the agent as a quasi-static change to the perceptual image of the object.

When an image of an object is characterized by a binary mask, the difference between two images A and B can be measured by the *Intersection Over Union* measure:

$$IOU(A, B) = |A \cap B| / |A \cup B|. \tag{5.1}$$

Comparing the images A of an object at times t_1 and t_2, when $IOU(A(t_1), A(t_2)) \approx 1$ the object has remained static. In case we observe $IOU(A(t_1), A(t_2)) \ll 1$, the object may have moved, but the object can only be definitively shown to have moved if care is taken to exclude the case of a temporary occlusion of an object by the hand or arm. In this work, this is done by choosing t_1 and t_2 at the beginning and end of the motion, which are times when the hand is at the home node position and will not occlude any position on the tabletop.

[1] We use the word "palm" for this region because of its functional (though not anatomical) similarity to the human palm, especially as the site of the Palmar reflex [15].

We define a *reach* as the action of following a trajectory resulting in a bump event with a target object. While the agent does not yet know how to make a *reach* action reliable, the IOU criterion will allow it to distinguish between successful and unsuccessful *reach* actions. In subsequent stages, the agent will exploit features with different values for the success and failure cases to learn how to *reach* reliably.

5.1.2 Experiments

The agent continues to practice its new capability to perform motions allowed by the PPS Graph and observe the results of these motions. Note that the experiments in this chapter (and Chapters 6 and 7) were conducted using Joint Position Control, the robot's built-in angular position-based control mode, and the Explored PPS Graph model, though it would also be possible to use Joint Velocity Control mode for the advantages considered in Section 4.2 and the improvements to the PPS Graph suggested in Section 3.2.2.3. We consider the motion capabilities and resulting PPS Graph with either control mode to be similar, and expect that these experiments would not have qualitatively different results if repeated with angular velocity-based control. We leave this repetition as a possibility for future work, as it should not affect the agent's learning or performance.

Experiment 5.1: Exploration. The agent follows this procedure:

1. Observe the environment while at the home node n_h, and find the initial mask for each of three objects newly placed in the foreground.

2. Select a random final node n_f in the PPS Graph.

3. Perform a graph search to determine the shortest path trajectory from the home node n_h to n_f.

4. Execute the trajectory, checking the visual percept at each node along the path for any significant change to an object mask.

5. If a difference is detected between the current percept recorded at the completion of each motion and the initial percept observed for an object, or the current node is n_f, immediately return to the home node along the shortest path.

6. Calculate the IOU values between the initial and final masks for each object.

    - If an apparent change at intermediate node n_i that triggered an immediate return is not confirmed (i.e., $IOU \approx 1$), then repeat the trajectory, continuing past n_i, to search for a subsequent bump event.

7. Cluster all IOU values seen so far into two clusters.

8. Repeat until the smaller cluster contains at least 20 examples.

By clustering the results of the IOU criterion, the agent learns to discriminate between the typical outcome of a trajectory (no change) and an unusual outcome (a bump event). These outcomes are defined as the unsuccessful and successful results of a *reach* action, respectively. Subsequent stages will identify features to allow increasingly reliable *reach* actions.

Experiment 5.2: Reach Reliability. To quantify the improvement in reliability in each stage, we first establish a baseline level of performance for the policy of selecting a random final node n_f and then following the shortest path in the PPS Graph to n_f. This experiment consists of 40 trials with a single, randomly-placed target block.

5.1.3 Results

Following this procedure, with three new objects added to the environment, the agent moved along 102 trajectories and gathered 306 IOU values between initial and final object masks. Figure 5.1 provides a histogram of all observed IOU values. Where t is the target object mask prior to the motion, and t' is the target object mask following the motion, the IOU values fell into two well-separated clusters.

$IOU(t, t') \approx 1$	285	typical: “no change”
$IOU(t, t') \ll 1$	21	unusual: “bump event”

Intuitively, a trajectory to a random final node is unlikely to interact with an object on the table. However, in a rare event the hand bumps the object, knocking it over or sliding it along the table, and sometimes off the table (the resulting absence of a final mask leads to an IOU of 0, so no special case is necessary).

The strategy of returning to the home node to observe the final mask allows the agent to rule out occlusion by the hand as the source of the perceptual change.[2] This has not been observed to make false positive bump classifications. This is important so that the agent will not learn incorrect conditions for a bump. There are a small number of false negatives where the hand and object do collide, but without lowering the IOU enough to fall into the smaller cluster. The agent is still able to learn the conditions from the reduced number of observed bumps, and may even favor actions that cause larger, more reliable bumps as a result.

[2] We have also experimented with alternatives to the IOU measure that can differentiate between bumps and occlusions without the return motion. One option is to identify a bump by the presence of non-empty set of pixels in the set difference $t' - t$. While an occlusion can hide pixels of the object, only an actual move of the object, such as a bump, can cause the object to appear in pixels where it did not appear before. However, this method tended to be more sensitive to noise due to the smaller size of occluded masks considered and required more sophisticated image processing, so at this time we have elected to require the return motion and use the IOU measure.

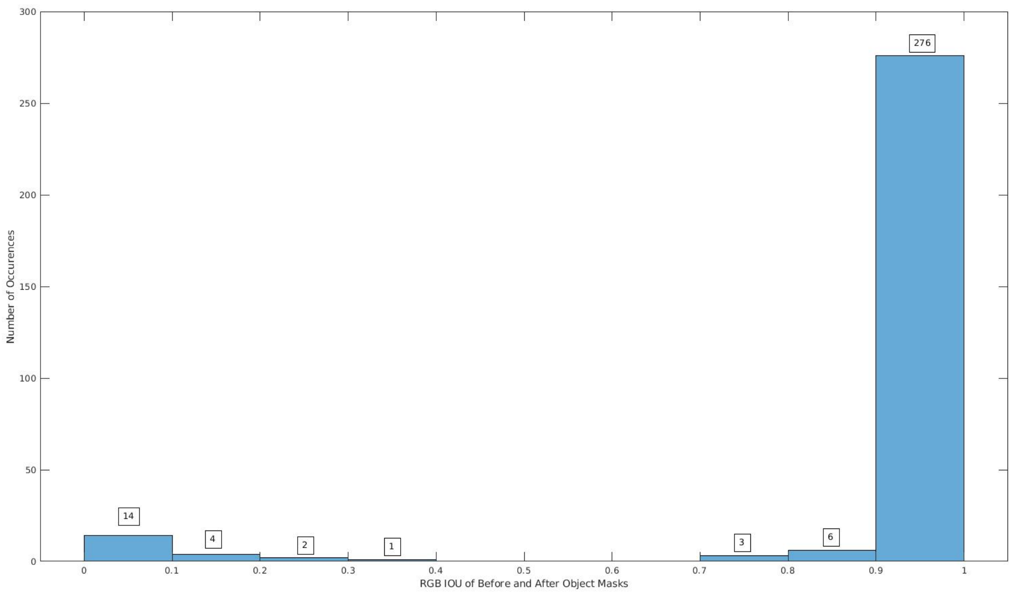


Figure 5.1: All observed values of $IOU(t, t')$ where t is a target object mask prior to a motion and t' is a mask for the same target object after the motion. Three objects were observed during 102 motions to produce the 306 values. There is a clear separation between a large cluster where the IOU is approximately one and a smaller cluster where the IOU is significantly less than one. The large cluster contains examples of the typical "no change" event and the small cluster contains examples of the unusual "bump" event.

The agent can classify all future motions in the presence of an object by associating the resulting observed IOU with one of the two clusters. While we human observers can describe the smaller cluster as a *bump* event, the robot learning agent knows only that the smaller cluster represents an unusual but recognizable event, worth further exploration. The agent has no knowledge of what makes a reach succeed. The following stages will help fill that gap.

The quantitative baseline experiment gives a reliability of 20% for the reach action to a random final node, which will be compared to other methods in Figure 5.4.

Reach reliability given selection method for n_f	
Select random target node n_f from PPS graph (baseline)	20.0%

5.2 Identifying Candidate Final Nodes

5.2.1 Methods

The agent has identified the rare event of a *bump*, and has defined *reach* as the action that can cause this event. Choosing a target node n_f randomly from the PPS graph gives a baseline reliability of 20%. The agent is now intrinsically motivated to search for ways to improve the reliability of the *reach* action. This can be done by identifying one or more features that discriminate between the cases that result in a bump, and those that do not.

The PPS graph stores a visual percept of the hand on each node, and the agent has a current visual percept of the target object. Comparing these percepts is straightforward, since they have the same frame of reference given our assumption of a static camera, and the agent has the RGB masks and the depth ranges from each image. Any nonempty intersection predicts that the hand and the target object will occupy the same region of the RGB image, or the same depth, or both.

The stored visual percepts also allow the agent to derive the image-space center of mass of the end effector at a given node. Centers and directions will have three components, two for the (u, v)-coordinates in the RGB image, and one (d) for depth values in the Depth image. For a node n_i, the center of the palm c_i^p is composed of the center of mass of p_i and the average depth, mean$(P_D(n_i)[p_i])$, and the center of the hand c_i^h is derived from h_i and $P_D(n_i)[p_i]$ in the same manner. Center c^t for a target object with mask t and depth range $D(t)$ in the current percept is also found analogously.

Using the PPS graph, the agent improves *reach* reliability in three steps.

1. Determine which binary image masks and which intersection property best predict the occurrence of a *bump* event.

2. Identify a set of candidate final nodes from the PPS graph with this intersection property. Select an arbitrary node in this set as the target node n_f.

3. Determine the best measure of closeness between centers of palm and target object, and select the closest node n_f from the candidate final node set.

5.2.2 Experiments

Experiment 5.3: Which intersection property is best? By further analysis of the data reported in Section 5.1.3 from 102 reaching trajectories, the agent can determine which binary image mask, and which intersection property, best predict whether a trajectory will produce a *bump* event.

The agent compares binary masks b representing the palm (p_f) or the hand (h_f) at its final pose or throughout its final motion $(s_{p,f})$. For each binary mask b and the mask t representing the target object, the trajectories are placed in four groups according to whether $b \cap t$ and/or $D(b) \cap D(t)$ are empty or nonempty. Counts of observed bumps and the total number of trajectories within each group allow the conditional probabilities of a bump to be computed.

The set of PPS graph nodes that satisfy the selected mask intersection property, with the best choice of mask, will define the set of *candidate final nodes* for a *reach* trajectory.

Experiment 5.4: Using the Candidate Final Nodes. An improved reach action policy can be created by selecting the target node n_f as a random member of the candidate final node set, rather than a random node from the entire PPS graph. The shortest graph path is found from the home node n_h to this final node n_f. This policy is evaluated using the same method as Experiment 5.1 in Section 5.1.2: reaching for 40 blocks, presented individually at randomly assigned locations on the table.

Experiment 5.5: Selecting the Best Candidate Node. In spite of every candidate node having non-empty intersections between the hand and target in terms of both the mask and depth range, the reliability of this reach action is still only 52.5%. One reason is that when the agent considers the binary mask derived from the RGB image and the depth range from the depth image together it over-estimates the space occupied by the hand or an object. In particular, the agent makes a simplifying assumption that the hand or object occupies the entire depth range across the entire mask. As a result, the intersection may take place in space that one or both objects will not actually occupy. Another reason is that some predicted intersections and corresponding collisions when the trajectories are executed may be very small, resulting in imperceptible bump events.

To address this issue, we provide four candidate distance measures between hand and target object and corresponding Boolean features that indicate if each measure is less than or equal to some threshold k. In particular, using the image-space (u, v, d) coordinates of the target object denoted by the superscript t and of the candidate final node denoted by the subscript f (and the superscript p as it is the coordinates of the center of the palm rather than the larger hand mask), we define three features for one-dimensional differences ($f_u = |u^t - u^p_f| \leq k$, $f_v = |v^t - v^p_f| \leq k$, and $f_d = |d^t - d^p_f| \leq k$) and one feature for the Euclidean distance between centers ($f_c = ||c^t - c^p_f|| \leq k$). Recalling when each feature was true in all previous reach trajectories and which reaches successfully caused a bump, the agent will determine which distance measure feature f and threshold k allows the conditional probability $P(Bump|f)$ to be maximized most reliably. The agent then selects from the set of candidate nodes the node that minimizes that distance measure. Once this node is chosen, the rest of the path is planned as before. This improved policy is evaluated the same way as Experiment 5.3.

5.2.3 Results

Experiment 5.3 results. The conditional probabilities and the results they are estimated by for each type of silhouette and each intersection feature combination are shown in Table 5.1. The set of intersection feature groups where $b = p_f$ contains the group with the highest conditional probability. A bump is most likely (64%) to occur at a final node n_f where the palm percept has a nonempty

Hand silhouette

	$D(h_f) \cap D(t) \neq \emptyset$		$D(h_f) \cap D(t) = \emptyset$	
$h_f \cap t \neq \emptyset$	64%	(7/11)	9.8%	(5/51)
$h_f \cap t = \emptyset$	7.6%	(6/79)	1.8%	(3/165)

Palm silhouette

	$D(p_f) \cap D(t) \neq \emptyset$		$D(p_f) \cap D(t) = \emptyset$	
$p_f \cap t \neq \emptyset$	64%	(7/11)	0%	(0/28)
$p_f \cap t = \emptyset$	14%	(12/84)	1.1%	(2/183)

Edge (hand) silhouette

	$D(s_{p,f}) \cap D(t) \neq \emptyset$		$D(s_{p,f}) \cap D(t) = \emptyset$	
$s_{p,f} \cap t \neq \emptyset$	60%	(12/20)	11%	(8/75)
$s_{p,f} \cap t = \emptyset$	1.2%	(1/87)	0%	(0/124)

Table 5.1: Each array represents the four possible intersection conditions, and each entry holds the conditional probability of a *bump* event in a trajectory satisfying that intersection condition, estimated as the ratio of observed *bump* events to possible bumps during trajectories. Recall that three objects were present for each of the 102 trajectories, so the total number of observations reflected in the denominators is 306. When the self is represented with either p_f and $D(p_f)$ or h_f and $D(h_f)$, having both intersections nonempty provides the highest conditional probability of a bump (64%). Using the second best intersection condition available with each silhouette (14% for the palm and 9.8% for the hand) as a tiebreaker, the palm silhouette is chosen as the best indicator of reliability for reaching. The edge silhouette is attractive for potential obstacle avoidance applications, since no bumps were observed over a large sample size of 124 possible bumps when both intersection features were empty.

intersection in both mask and depth range with the target percept, that is, where

$$p_f \cap t \neq \emptyset \quad \wedge \quad D(p_f) \cap D(t) \neq \emptyset. \tag{5.2}$$

The process of identifying a node as a candidate is demonstrated in Figure 5.2. If no nodes are found that satisfy the condition of Equation 5.2, the criteria are relaxed so that the candidate set may include nodes with the next most reliable intersection set, which was when $D(p_f) \cap D(t) \neq \emptyset$ but $p_f \cap t = \emptyset$, as 14% of past motions to such nodes caused bumps. To ensure the candidate set never remains empty, if there are still no candidates, the criteria can be further relaxed for two additional iterations, first to include nodes with $p_f \cap t \neq \emptyset$ but $D(p_f) \cap D(t) = \emptyset$ and finally to include nodes where both $p_f \cap t = \emptyset$ and $D(p_f) \cap D(t) = \emptyset$. Since these sets span all possible combinations of intersection features, with the inclusion of this final set, the set of candidate nodes is guaranteed to be nonempty will in fact contain every node in the PPS Graph. This guarantee allows the agent to plan reaches by its ordinary method when nodes with desirable intersection features are absent. While a random choice among such an inclusive set is unreliable (equivalent to the baseline), Experiment 5.6 will show that an informed choice among this set can be sufficient.

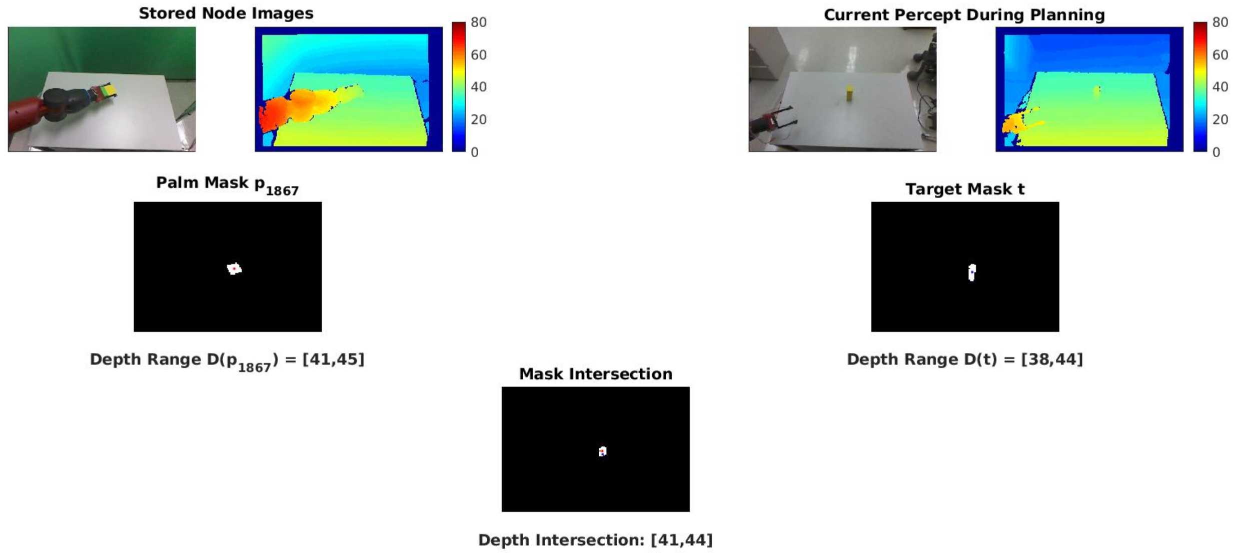

Stored Node Images
80
60
40
20
0
Current Percept During Planning
80
60
40
20
0
Palm Mask p_{1867}
Target Mask t
Depth Range $D(p_{1867})$ = [41,45]
Depth Range D(t) = [38,44]
Mask Intersection
Depth Intersection: [41,44]

Figure 5.2: Candidate final nodes are identified from their intersection features. [**top row**] RGB-D percepts taken from the node definition (left two) and from the current percept of the environment (right two). [**middle row**] The palm masks and depth ranges from the stored percept (left) and current percept (right). [**bottom image**] The intersections of the palm masks and the depth ranges are both non-empty, so the current node is identified as a candidate for reaching the observed block. The palm masks and depth ranges for each node can be computed in advance. The intersections of the mask and range from a target block can be quickly evaluated for all 3000 PPS graph nodes to generate the set of candidate final nodes.

Experiment 5.4 results. Of the same 40 placements as the baseline (Experiment 5.1), 39 have at least one node with both mask and depth range intersections with the target, and the policy of moving to one of these nodes bumps the target 21 times. Only one placement required a relaxation of intersection criteria to have a non-empty candidate final node set because no node had both an RGB and Depth intersection with the target. Attempting a reach to that placement with this method was not successful. Overall, the reach action is now 52.5% reliable. The comparison in Figure 5.4 shows reaching to an arbitrary candidate node is more than twice as reliable as the baseline action of moving to a random final node.

Experiment 5.5 results. Figure 5.3 shows the results of comparing several different distance measures between the center positions of the hand and of the target object. This result supports the use of the final node candidate with the smallest center to center distance with the target $||c^t - c^p_f||$. The choice of the Euclidean distance measure agrees with our intuition, as it best conveys when the hand and object are actually close, whereas the hand and object could have similar values for a single coordinate but not be in close proximity, due to differences in the other coordinates. Attempting the 40 reaches again with final nodes chosen using this feature, the agent now considers the reach action to be 77.5% reliable, with 31 successes, 7 false negatives, and 2 actual failures to bump the object. This result is also included in the comparison in Figure 5.4.

Tabulated results from Experiments 5.2, 5.4, and 5.5:

Reach reliability given selection method for n_f	
Select random target node n_f from PPS graph (baseline)	20.0%
Select arbitrary candidate node n_f	52.5%
Select candidate node n_f with hand center closest to target center	77.5%

This method, for identifying candidate target nodes that increase the probability of bumping a specified block, can be extended to avoid bumping specified blocks.

5.3 Reaching to Positions between PPS Graph Nodes

5.3.1 Methods

Recall that the first improvement to the reach action was to learn to identify a set of candidate final nodes, which for a given target is the set containing all nodes where the stored hand representation and the current percept of the target intersect in both the RGB and depth images. Moving to an arbitrary candidate final node instead of a random node from the PPS graph more than doubles the rate at which bumps are successfully caused. However, the line for f_c in Figure 5.3 demonstrates that the success rate for reaches increased as $||c^t - c^p_f||$ decreased. Choosing the candidate node nearest to the target object improved the reliability of the reach to 77.5%, but this method is limited by the density of the PPS Graph near the target. Especially in relatively sparse regions of the graph, even the nearest node may not be close enough for a reliable reach. The agent must learn to make small moves off the graph to reach closer to the object than the nearest node.

After the agent has planned a reach trajectory that would use a final node n_f, it can use the local Jacobian $\hat{J}(n_f)$ and its pseudo-inverse $\hat{J}^+(n_f)$ to improve the accuracy of its final motion, and the likelihood of causing a bump event. Recall from Section 4.4 that finding the difference between n_f and each of its neighbors in configuration space produces a matrix ΔQ, and concatenating each corresponding difference in image space produces a matrix ΔC. Repeating Equation 4.7 for convenience, $\hat{J}(n_i)$ is the least squares solution of

$$\Delta Q \ \hat{J}(n_i) = \Delta C. \tag{5.3}$$

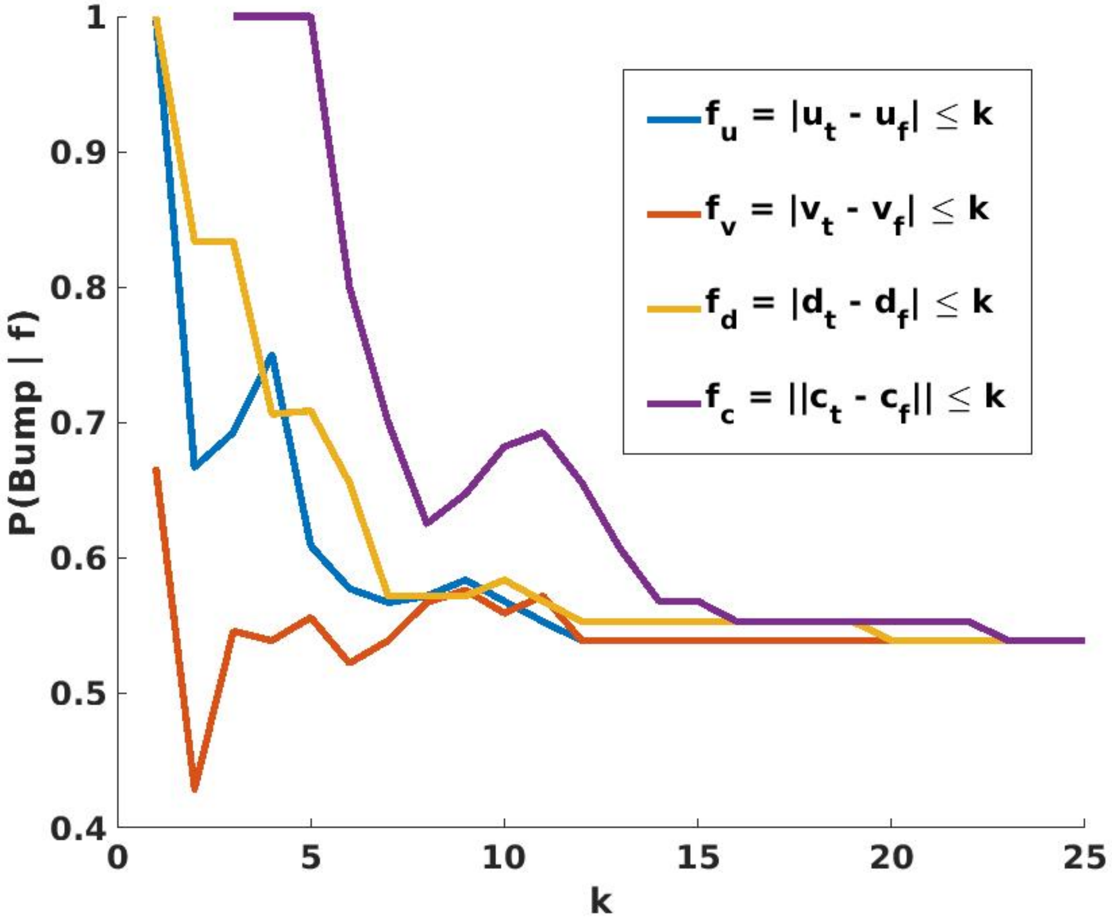


Figure 5.3: Given percepts for hand and target object, the agent searches for the feature f that will maximize the conditional probability $P(Bump \mid f)$. Each feature considers the centers of the palm and target in (u, v, d) image-space. f_u, f_v, and f_d evaluate to true if the absolute difference in one coordinate is less than a variable threshold k, and f_c is true if the distance between centers is less than k. The probabilities shown in this graph are based on the 102 trajectories used previously, and their outcomes. For all values of k, $P(Bump \mid f)$ is maximized when $f = f_c$. The agent therefore selects as n_f the candidate node where the hand is closest to the target object, thereby minimizing k and maximizing $P(Bump \mid f)$.

Since $\hat{J}(n_i)$ is a non-square 7×3 Matrix, it has no true inverse, but domain-general linear algebra allows the pseudo-inverse $\hat{J}^+(n_f)$ to be calculated. While the local Jacobian can be used as a mapping from a proposed configuration change to an estimated resulting change in image space, its pseudo-inverse can be used as a mapping from a desired change in image space to the estimated change in configuration that would be necessary to produce it.

With the center of the palm in the stored percept of the chosen n_f denoted c^p_f, and the center of the target object in the current percept during planning denoted c^t, the desired change in the palm percept is $\Delta c = c^t - c^p_f$. The agent can estimate the change in configuration needed to produce this desired visual change, and add it to the stored configuration q_f to produce an updated final configuration for the planned trajectory,

$$q^*_f = q_f + (c^t - c^p_f)\hat{J}^+(n_f) \tag{5.4}$$

When the agent moves to the configuration q^*_f, the palm center should be approximately aligned with the target's center. A motion that aligns the centers should increase the size of the intersection, making the action robust to noise, and increasing the likelihood of the resulting bump event.

While the ability to make a small move off of the graph to q^*_f increases the robustness of the reach, it does not eliminate the need for a set of candidate final nodes, or for the decision to use the nearest node to the target as n_f. As $\hat{J}^+(n_f)$ is a local estimate, if $||c^t - c^p_f||$ is large, the error in the recommended Δq will also tend to be large. Choosing the nearest candidate n_f minimizes the factor by which natural errors in $\hat{J}^+(n_f)$ will be multiplied, giving the best accuracy for the final position of the reach. Adding the use of the inverse local Jacobian gives the final reaching procedure below.

5.3.2 Experiment 5.6: Reaching to target adjusted by local Jacobian

The final improvement in the *reach* action starts with the trajectory planned to the closest candidate node n_f to the target object. The configuration q_f in that node is then adjusted according to the local Jacobian for the neighborhood of n_f. The final motion in the trajectory then goes to q^*_f, rather than q_f. In effect, the PPS graph supports a local linear approximation to the full Jacobian over the continuous configuration space, based in the neighborhood of each node.

This improved policy is evaluated the same way as Experiments 5.2, 5.4, and 5.5.

5.3.3 Experiment 5.6 Results

Using this procedure on the training set of target placements, the agent perceives bumps at the final node of all 40 trajectories. This 100% result demonstrates that the reach action has become reliable, and is a significant improvement from the previous methods shown in Figure 5.4.

Reach reliability given selection method for n_f	
Select random target node n_f from PPS graph	20.0%
Select arbitrary candidate node n_f	52.5%
Select candidate node n_f with hand center closest to target center	77.5%
Adjust target away from n_f using local Jacobian	100.0%

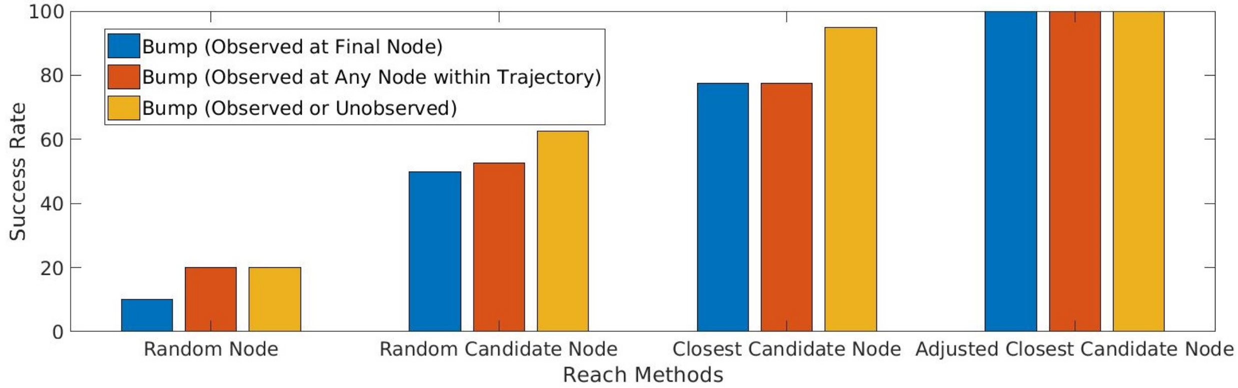


Figure 5.4: Reliability of the agent's action to reach and bump a single target object by following a trajectory to a selected target node. The four groups represent (1) randomly selected target node; (2) random selection from among candidate nodes with non-empty image intersections; (3) select closest among candidate nodes; (4) adjust node with local Jacobian to best match target object. Within each group, the bars represent different criteria for success: (l) observed bump at final node, which measures the agent's ability to cause bumps intentionally and efficiently; (m) observed bump anywhere in the trajectory, which identifies bumps that can be learned from; (r) any bump, observed or unobserved, which measures ground truth of bump occurrence.

5.4 Transfer of Learning for Obstacle Avoidance

Intuitively, reaching to a target and causing a bump requires causing a collision between the hand and the target so that the hand moves the target to a new position. So far, the agent has learned intersection features and other visual properties that identify nodes that will be reliable for reaching. This expected reliability is based on the high conditional probability that moving to a node with those properties will cause a bump. At the same time, it learns features and values that identify nodes that will not be reliable for reaching because of the low conditional probability that moving to them will cause a bump. While such nodes are removed from consideration for the candidate final node set for reaching, they are desirable in the case of an obstacle avoidance task, where bumps with any foreground object treated as obstacles are undesirable. Because both tasks rely on predicting the likelihood of a bump when moving to a node or along an edge, the same features can be used and there is potential for a high degree of transfer learning. Except for adding additional terms to the formula for calculating the best path, adding obstacle avoidance requires no additional learning - the agent can simply select the nodes and edges with the lowest likelihood of a bump when avoiding obstacles instead of selecting those with the highest likelihood when reaching a target.

The simplest trajectory for avoiding obstacles is to not move at all. The agent could also plan moves around the periphery of the environment, far away from the tabletop and anywhere with potential for collisions with the foreground objects. Obstacle avoidance is a more interesting task in a context where the agent must avoid obstacles while reaching to a target, modifying the planned trajectory as necessary. While we have not conducted this experiment using our current model and experiment set, we have addressed this task in past work [22].

In Sections 3.2.2.1-3.2.2.2, we discuss alternative PPS Graphs that were built using a different motor babbling technique and different visual sensors. Through the experiments described in [22], the agent used these tools to learn how to reach with a 90% success rate - a demonstration that the learning described in this chapter is not dependent on the specific sensors used or the details of the PPS Graph. While these were quite different in [22] (three low-resolution webcams that gathered two-dimensional images and a graph with 1001 nodes with randomly sampled configurations, respectively), the agent was able to learn analogous features in that setting, and those features also produced highly reliable reaching. During Experiments 5 and 6 of that work [22], the agent performed reaches in pairs of trials with two foreground objects placed randomly on the tabletop. In first trial of the pair, one object served as the target and the other served as the obstacle, and in the other trial these roles were swapped. In order to discourage obstacle avoidance strategies such as not moving or not approaching the target object, the reaching task was weighted more heavily. This experiment yielded the 90% success rate for reaching - a number that may have been higher if the agent was not required to consider the obstacle when selecting a trajectory. Despite treating obstacle avoidance as the secondary goal and using the same trajectory, the agent succeeded at avoiding the obstacle in 74% of trials, and in 68% of trials both the reaching and avoidance goals succeeded.

In [22], we relied on edge silhouettes, a representation from the perspective of each camera of the space that would swept through by the hand as it moved along an edge. This was particularly important due to the use of the Learning PPS Graph (Section 3.2.2.2), which was especially sparse with only nine manually placed nodes and relatively long edges. Due to the structure of this graph, a high number of bumps occurred during the motion, and would not have been predicted if only considering the palm or hand masks at the endpoint nodes themselves. Checking only for intersection with the node hand masks would only predict bumps at the very beginning or end of the motion. In addition to allowing consideration of bumps that may occur during the motion, edge silhouettes provided a conservative over-estimate of the space swept through because they ignored the timing of the hand's motion through the larger mask, instead treating the hand as occupying the full mask for the full duration. This allowed for some false positive bumps along some edges to be predicted, so the agent required additional features to rule out those edges when choosing the final edge of a reach trajectory. But when avoiding obstacles, the over-estimating nature of the edge silhouettes was helpful, as there were very few false negative predictions, and an edge with no perceived intersections was highly reliable.

To repeat this experiment in the context of this work, the edge silhouette $s_{p,f}$ and its depth range $D(s_{p,f})$ are most similar to the three edge silhouettes used previously. Additional features such as intersections using the smaller p_f and $D(p_f)$ could still be used to refine the selection of candidate final nodes for reaching. While it is possible that the edge silhouette will again be naturally useful for obstacle avoidance, we hypothesize additional features may be needed as the space swept through is even more so overestimated when using $s_{p,f}$ and $D(s_{p,f})$ that are available from the RGB-D camera. For moves along long edges, treating the hand as occupying the entire depth across the entire mask makes the implied space so large that it becomes uninformative. It may also not be possible to avoid an overlap between the overly large hull formed and a tabletop obstacle in any successful reach trajectory. In future work, two or more foreground objects can be placed and the agent's success rate for avoiding obstacles while reaching can be evaluated with only the edge silhouettes and with new features until a satisfactory reliability is achieved.

5.5 Conclusion

By the end of Chapter 4, the agent had learned to move all the joints of the arm to set points individually or in parallel. After the agent had built the PPS Graph by motor babbling, it was able to use the information stored in the nodes and edges to move throughout its environment. The agent had also learned to make small adjustments to the stored configurations so that it could move with suitable precision to positions within its continuous peripersonal space as long as they were in the neighborhood of a node's configuration. In Chapter 5, the agent identified the unusual event of a bump and applied these capabilities for moving the arm to learning a reach action to recreate that event with increasing reliability. Incorporating intersection features and a technique with the pseudo-inverse of the local Jacobian to reduce expected distance from the target produced a 100% reliable reach.

In section 5.1, the agent conducted a large sequence of moves to randomly selected nodes along PPS Graph paths and discovered that moving to some nodes caused the position at which a foreground object appeared in the visual percepts to change significantly. The agent computes the Intersection over Union (IOU) between the before and after binary mask (t and t') of each foreground object present for each node moved to. Using clustering, the agent found many examples of a typical result where $IOU(t, t') \approx 1$ and a small cluster of unusual results where $IOU(t, t') \ll 1$.

We call this particular type of unusual event a bump. The agent calculated an IOU threshold using these clusters so that it could classify future observations as typical results of moving (no bump) or a bump. The section concludes with Experiment 5.2 that established a baseline that 20% of trajectories from the home node to a random node with a single target object cause a bump. The remaining experiments in this chapter are the agent's process of learning a reach action that causes bumps more reliably.

In section 5.2, the agent improved the reach action by learning better criteria for selecting the final node of the trajectory. In Experiment 5.3, the agent reviews its past experiences to determine that the conditional probability of a bump is highest when the stored binary mask and depth range for the final node both intersect with their counterparts observed for the target object while planning. The agent applied this finding in Experiment 5.4 and by randomly selecting from the set of nodes with the best intersection features available instead of from the set of all nodes was able to improve the reliability of the reach to 52.5%. In Experiment 5.5, the agent chose to move to the closest of these candidate final nodes to the target (according to the most informative distance measure, found to be $||c^t - c^p_f||$) instead of a random candidate final node, which made the reach action 77.5% reliable.

In section 5.3, the agent uses its ability to move to positions interpolated between nodes. While this capability had been present since the agent finished building the PPS Graph and could analyze its neighborhoods to calculate the local Jacobian and its pseudo-inverse, it is better for the agent not to use this capability in the earlier stages of learning to reach since it does not have definitive visual information for these interpolated positions. The known visual percepts stored in the nodes facilitated learning the intersection features and importance of closeness without relying on estimated masks and depth ranges. However, with these techniques known, the agent can then calculate the arm configuration in the neighborhood of the nearest candidate final node that it estimates would give the hand the same image-space center as the object, for $||c^t - c^p_f|| = 0$. When the agent replaced the final node in the reach trajectory with this modified position the action became 100% reliable over the set of positions tested in Experiment 5.6.

Section 5.4 discussed the previous work [22] that demonstrated learning a 90% reliable reach action with a different PPS Graph and different visual system, requiring different features to be learned. The existence of two alternative methods that produce successful reaches support the generality of the type of learning from unusual events performed, and shows that the specific choices of sensors and assumptions are not required. In [22], we also showed that the agent's ability to reach depended on predicting whether motion to a node will or will not produce a bump, and this skill could also be applied for obstacle avoidance. By using a trajectory that was expected to bump the target object with only the final move and not bump any other objects with any of the moves, the agent avoided those collisions in 74% of trials, with 68% of trials succeeding at both reaching the target and avoiding a bump the other object. The principle that nodes that are unreliable for reaching because motion to them is unlikely to cause a bump will be reliable for obstacle avoidance is also true in the context of this work when the RGB-D camera and related features are used. In future work, the ability to identify these nodes can be used for both purposes, and the agent using the current vision system may be evaluated on a task where it reaches while avoiding obstacles.

With a fully reliable reach action, the agent can readily interact with objects placed in its environment. Once the set of example interactions is large enough, the agent can repeat the process of learning from unusual events. Because the reach action is reliable, its typical result is a bump. While a simple model of treating all changes in target object position that are not based on occlusions as bumps was sufficient for learning to reach, the agent can improve its model with the revision that some of the interactions with the object that result from successful reaches are qualitatively different. In the next chapter, the agent focuses on grasps, where the hand closes around the object and gains control over it. A grasp can be classified by the presence of a behavior where the object follows the motion of the hand while this control is maintained. Chapter 6 discusses features that allow the agent to predict which subset of reach trajectories will produce a grasp, and apply them to make a semi-reliable grasp action.

CHAPTER 6

Learning a Reliable Grasp Action

In our model, after the intrinsic motivation pattern has resulted in a reliable reach action, the pattern may be applied a second time to learn a grasp action. As the reach action toward a target object becomes more reliable, the result of causing a quasi-static change in the image of that object becomes more typical. However, there is an unusual result: during the interaction with the target object, the hand may reflexively close, providing sensorimotor experience with "accidental grasps".

Driven by intrinsic motivation, the grasp action becomes more reliable, toward becoming sufficient to serve as part of a pick-and-place operation in high level planning. Additional features may be considered in a flexible order or in parallel as the agent identifies and learns to satisfy the requirements for a reliable grasp action. In this work we present one such set of features considered by our agent in a set of distinct learning stages in a fixed order. This order was determined by the experimenter and facilitates incremental evaluation of grasp reliability as each feature was added. This chapter presents the grasp requirements and features used to satisfy them in the order they were considered within our agent's set of learning stages. The agent must begin with the Palmar reflex to observe the unusual results of a reliable reach action without consciously closing the hand with correct timing. Our agent then learned: how to most reliably set the gripper's aperture during the grasp approach, how to best align the hand, target, and final motion, and how to preshape the hand by orienting the wrist. Each stage is presented with a Methods-Experiments-Results description.

6.1 Reaching with an Innate Palmar Reflex

6.1.1 Methods

Human infants possess the *Palmar reflex*, which closes the hand as a response to contact of an object to the palm. Our work assumes that the Palmar reflex is innate and persistent during at least early stages of learning to grasp. Within our framework, the primary importance of this reflex is to enable the observation of accidental grasps as an unusual event while reaching. While the closing of the hand is unconscious, the agent learns the motor commands and sensations of closing the hand.

When conditions are correct, the Palmar reflex causes an accidental *grasp*, where the object is held tightly in the hand and becomes a temporary part of the self. This gives a much greater level of control over the pose of the object, as it can be manipulated with the agent's learned scheme for moving the hand until the relationship ends with an *ungrasp*, an opening of the fingers that releases the object. The variety of outcomes possible with the level of control a grasp provides imply a high potential reward from learning to predict the outcomes and actions to cause them, but it is also the case that grasps occur too rarely to learn immediately after learning to reach. Without enough examples, learning the conditions for a grasp may prove too difficult, leading to a modest rate of improvement and a low reward. In our model, the agent focuses next on an intermediate rare event.

The activation of the Palmar reflex is such an event that may be observed as an unusual result of successful reaches. When the hand's final approach to the target meets all necessary conditions of openness, alignment, and orientation, the target object passes between the grippers in a way that activates the simulated Palmar reflex, and the gripper fingers close. The openness of the grippers is a degree of freedom for the robot's motion, and is continually sensed by proprioception. As a result, accurate detection of when the Palmar reflex has been triggered does not rely on the visual percept, and can be observed in a rapid decrease of openness to a new fixed point.

The closing of the grippers, either by reflex or conscious decision, is necessary for the agent to gain control over the object with a *grasp*. In some cases, the initial interaction between the hand and object does not lead to the grippers closing around the object, and the attempt to gain control fails immediately. We refer to this event as a *Palmar bump*, as it often involves knocking away the object before the grippers can close on it. Like other bumps, this is a quasi-static change with an observably low IOU value between the object's before and after masks, and it is the result of a successful reach. While the Palmar bump is not a successful grasp, it serves as a useful near-miss example, promoting use of the conditions that allowed the reflex to trigger in this attempt and can be expected to allow it to trigger in future grasp attempts.

When a *grasp* occurs, the activation of the Palmar reflex is followed by the object shifting from its initial quasi-static state to a new dynamic state. Now held between the gripper fingers, the object begins to follow the hand with continued motion correlated with the motion of the hand. The agent can identify this corresponding motion by comparing masks and depth ranges during the return trajectory. A grasp is successful if and only if the stored masks and depth ranges for each node of the trajectory intersect with those of the target object in the visual percepts during the return to the home node. Note that the full hand masks and depth ranges are used since the gripper fingers, once closed, may obscure the portion of the object in the palm region. If all nodes of the trajectory have an empty mask or depth range intersection, control was never gained and the result is a Palmar bump. If at least one node fails the intersection check, but not all nodes, the grasp is considered to be a *weak grasp*. Here the grasp was initiated, but due to a loose or poor placement, did not persist through the return trajectory. Note that the loss of control of the object in a weak grasp does not involve an opening of the grippers, as an intentional *ungrasp* action would. Figure 6.1 provides an example of the agent's visual percepts of a trajectory that produced each type of result.

Since the Palmar bump and weak grasp cases fail to gain or maintain control of the object, both are successful reaches but failed grasps. By considering both situations to be failures, the successful grasps that emerge from this learning process are more likely to facilitate subsequent learning of higher order actions that require a grasp. However, Palmar bumps, weak grasps and grasps share the sensed result of reflexively closing the hand, and may be assumed to share similar preconditions as well. Until a sufficient number of successful grasps are observed, the agent will draw information from all cases where the Palmar reflex was activated to learn to grasp.

6.1.2 Experiment 6.1: Monitoring the Palmar Reflex During Reaching

We first attached a break-beam sensor between the tips of the Baxter robot's parallel gripper fingers to provide the agent with a simulated innate Palmar reflex. Then our agent repeated all trials of Experiments 5.2, 5.4, 5.5, and 5.6 in chapter 5, using the same target placements and planned trajectories. For each trial, the agent records whether the Palmar reflex was activated, and which category of result (grasp, weak grasp, Palmar bump, bump, or miss) it observed.

6.1.3 Experiment 6.1 Results

It is clear that learning to reach more reliably and with greater precision allows more Palmar reflex activations and grasps to occur. With the random trajectories of Experiment 5.2, one of 40 activated the Palmar reflex, and this was a successful grasp. Using the final reaching method of Experiment 5.6, the agent observed that the Palmar reflex was activated in 12 out of the 40 trials. Of these 12, 5 were successful grasp trajectories. These provide a baseline reliability of grasping with random motion trajectories (2.5%) and of grasping with a reliable reach trajectory (12.5%). These results and those for intermediate reach methods are tabulated below, and also shown alongside the

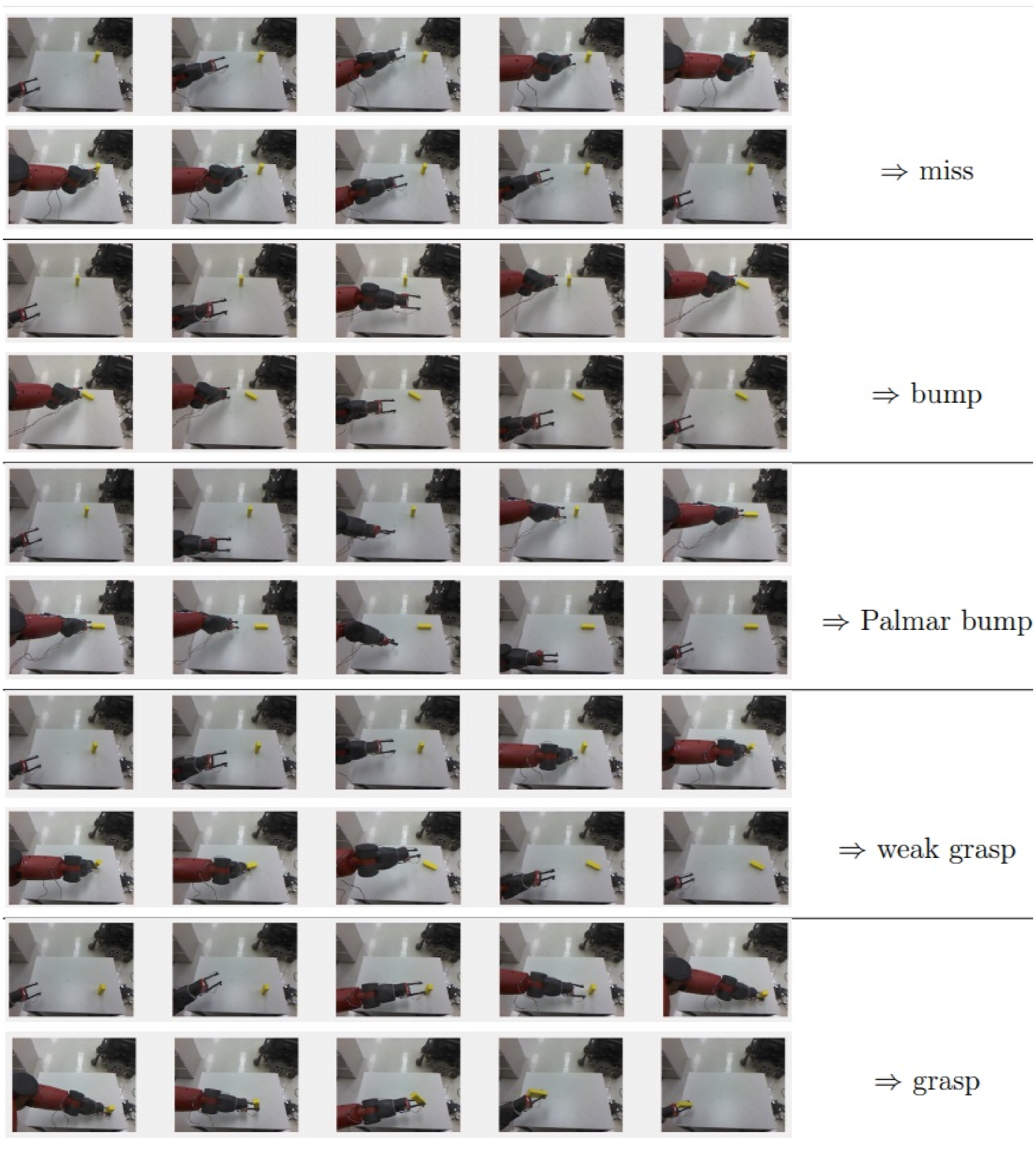


Figure 6.1: The agent's RGB percepts during attempted grasp trajectories. Images for the forward portion toward n_f are shown in the first of each pair of rows, and images for the portion to return to n_h are shown in the second rows. Images for some nodes in the middle of trajectories with more than five nodes have been omitted. The agent classifies the result of the grasp attempt by observing the state of the target object during the trajectory. In all cases but *miss*, there is a substantial change between the first and last observations, and the trajectory is a successful reach. In all other cases these observations should be significantly different, and the reach component of the grasp was successful. Further classification depends on the state throughout the return trajectory and whether the Palmar reflex was activated, as discussed in section 6.1.1. Only the result of the final example is considered to be a successful grasp.

rest of the results for this section numerically in Figure 6.4 and spatially in Figure 6.5.

Tabulated Results from Experiment 6.1, using trajectories from the reaching Experiments 5.2, 5.4-5.6 as indicated:

Results	**Grasp: Successful**	**Failed**			
	Palmar Reflex: Activated			**No Activation**	
	Reach: Successful				**Failed**
	Grasp	Weak Grasp	Palmar Bump	Bump	Miss
Experiment 6.1 (5.2)	2.5%	0%	0%	17.5%	80.0%
Experiment 6.1 (5.4)	2.5%	0%	7.5%	42.5%	47.5%
Experiment 6.1 (5.5)	5.0%	0%	12.5%	60.0%	22.5%
Experiment 6.1 (5.6)	12.5%	0%	17.5%	70.0%	0%

6.2 Initiating Grasps with the Gripper Fully Open

6.2.1 Methods

While exploring PPS and performing reaches, the agent is motivated to keep the hand fully open ($a = 100$). This presents the largest silhouette of the hand to keep in view, as desired, and the full extension allows for more interactions with objects when the extremities collide with them. As the PPS Graph was created, this setting also allowed a brightly colored block to be placed spanning the full width of the grippers, simplifying visual tracking of the “palm”.

With the new event of a Palmar reflex activation during the interaction, the agent may choose to investigate its degrees of freedom. Each of the joint angles in q have a role in modifying the placement of the hand that is modeled by a node’s local Jacobian estimate, but a does not appear to significantly affect the location of the hand’s center of mass and does not differentiate graph nodes. This allows it to be freely modified to investigate its influence on the frequency of Palmar reflex activations.

6.2.2 Experiment 6.2: Which gripper aperture setting is most reliable?

While it is intuitively desirable for the agent to approach targets with the grippers open for a Palmar bump or grasp, the agent does not yet have sufficient data to reach this conclusion. This is gathered by repeating the trajectories of Experiment 5.6, the final reaching method, with the Palmar reflex active and each gripper aperture of 0%, 25%, 50%, and 75% open. These four sets of results can be compared with those for the fully open gripper that were already obtained in Experiment 5.6.

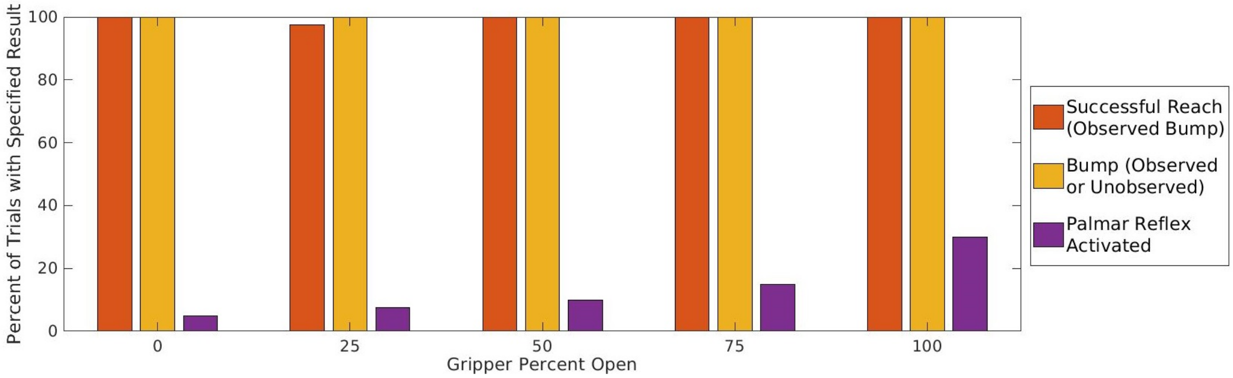


Figure 6.2: The portion of attempted reach trajectories that produce observed bumps (orange), ground truth bumps (yellow), and Palmar bumps, or bumps which also trigger the Palmar reflex (purple) for varying gripper apertures a. The high reliability of the reach action is independent of a, indicating it could be learned and executed with any setting. By contrast, triggering the Palmar reflex is much more likely as a increases, and is learned as a prerequisite for the Palmar bump event and later for the grasp action.

6.2.3 Experiment 6.2 Results

Two conclusions may be drawn from the results of this experiment, which are visualized in Figure 6.2. First, it is clear that the probability of activating the Palmar reflex increases with the openness a of the gripper during the approach. As a decreases, the opening of the hand narrows, and the object is less likely to pass inside with an approach of equal precision, so there are less activations. Once a is sufficiently low that the object cannot fit in the hand, the Palmar reflex never triggers. The agent will continue using the fully open setting $a = 100$ in future attempts to maximize its expected success rate.

Second, we see that the openness of the gripper has almost no affect the probability of a bump. In fact, only one trial was perceived to fail with any setting, and this was a false negative. We claim that this demonstrates the agent could have learned the reach action with the same process and ending reliability for any gripper setting, and at that point would learn to prefer 100% open. It is therefore not necessary for our model to assume any initial setting a for the gripper opening while learning to reach.

6.3 Planning the Approach with Cosine Similarity Features

6.3.1 Methods

When reaching, it is important that the candidate final nodes satisfying

$$p_f \cap t \neq \emptyset \quad \wedge \quad D(p_f) \cap D(t) \neq \emptyset. \tag{6.1}$$

are identified, and n_f is chosen to minimize $||c^t - c^p_f||$. To plan reaches that activate the Palmar reflex, additional features are needed to ensure not only that the final position is correct, but also that the hand orientation and the direction of final motion are suitable. These must be compatible during the approach, and must also be effective for the current target object. To learn to use satisfactory relationships between these vectors, the agent constructs this set of vectors using information from its stored and current visual percepts:

gripper vectors: pointing outward, near parallel to the gripper fingers.

$\vec{g}_p \equiv$ drawn from c^h_p through c^p_p

$\vec{g}_f \equiv$ drawn from c^h_f through c^p_f

motion directions: direction of motion along an edge or toward a target

$\vec{m}_{p,f} \equiv$ the direction of the edge-based final motion from c^p_p to c^p_f (6.2)

$\vec{m}_{p,t} \equiv$ the direction of the modified final motion from c^p_p to c^t

$\vec{m}_{f,t} \equiv$ the direction of displacement from c^p_f to c^t

object orientation: the perceived major axis of the target object

$\vec{o} \equiv$ drawn along the major axis of t.

As described in Section 3.4.4, the agent is instructed to extract these vectors from the perceived images. This instruction is justified as these vectors are a natural and general way to describe the relationships between binary image masks like those used to describe the positions of the hand and foreground objects in image-space. Without further instruction, the agent learns cosine similarity criteria for the vectors of final motions that most reliably cause Palmar reflex activations in Experiment 6.3. In Experiment 6.4, the agent plans trajectories with final motions that satisfy this criteria to improve the reliability of Palmar reflex activations and grasps.

6.3.2 Experiments

Experiment 6.3: Learning reliable cosine similarities. To discover the best relationship between these vectors for repeating the Palmar reflex activation event, the agent uses the data from repeating the final reach trajectories of Experiment 5.6 in Experiment 6.1 with the Palmar reflex enabled. For each trajectory, it considers the cosine similarity $C(\vec{v}_1, \vec{v}_2)$ of each pair $\vec{v}_1, \vec{v}_2 \in \{\vec{g}_p, \vec{g}_f, \vec{m}_{p,f}, \vec{m}_{p,t}, \vec{m}_{f,t}, \vec{o}\}$ and results. The cosine similarities are discretized to the nearest value in {-1, -0.5, 0, 0.5, 1}. The rate of Palmar reflex activations is observed for trajectories grouped by their discretized C values.

Experiment 6.4: Planning well-aligned final motions. The agent uses the results of Experiment 6.3 to plan the next set of trajectories to interact with the target. At this time, the agent does not have the ability to change any $\vec{g}_i$ to a particular direction to be perpendicular to $\vec{o}$. Therefore, instead of the nearest candidate final node, n_f is selected from the candidates such that $|C(\vec{g}_f, \vec{o})|$ is minimized. As before, $\hat{J}^+(n_f)$ is computed and used to modify the final configuration to a more reliable q^*_f by equation (5.4). The agent may apply $\hat{J}^+(n_f)$ again to create a preshaping position, a copy of the final position translated in the direction of $-\vec{g}_f$. This image-space translation has a magnitude of 21, the mean length of the final motion for all Palmar bumps and grasps previously observed. The preshaping position has configuration

$$q^*_p = q^*_f + 21(-\vec{g}_f/||\vec{g}_f||) \tag{6.3}$$

and will replace q_p. With this use of $\hat{J}^+(n_f)$, it is expected that $\vec{g}_p \approx \vec{g}_f$, and the motion from q^*_p to q^*_f should be in the direction of $\vec{g}_f$, opposite of the translation. In place of $\vec{m}_{p,f}$, $\vec{m}_{p,t}$, and $\vec{m}_{f,t}$, the direction of this motion is parallel to the gripper vector and near perpendicular to the target major axis. The three steps of choosing n_f, adjusting to q^*_f to match centers with the target, and translating to create a well-aligned preshaping position with q^*_p are visualized in Figure 6.3.

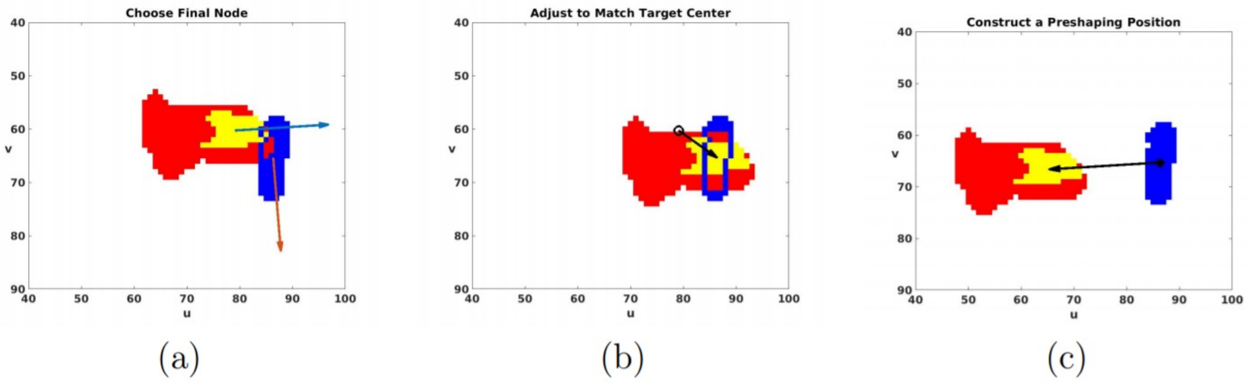
The agent must plan a trajectory that ends with this approach. q^*_p is not stored in $\mathcal{P}$, so to find a feasible path to q^*_p, the agent first identifies the nearest node $n_n \in \mathcal{P}$ that minimizes $||q^*_p - q_n||$. A graph search then yields the shortest path from the home node to n_n. After visiting n_n, the arm will be moved from q_n to q^*_p, and then make the final motion to q^*_f to complete the trajectory.

The reliability of the grasp action using this method for planning trajectories with aligned final motions is evaluated using the same layout of target placements as Experiment 5.6, with the Palmar reflex enabled as in Experiment 6.1. The agent also continues to record the frequency of all types of Palmar reflex activations.

6.3.3 Results

Experiment 6.3 results. When $\vec{v}_1 \neq \vec{o}$ and $\vec{v}_2 \neq \vec{o}$, the highest rate of Palmar reflex activations occurs in the $C(\vec{v}_1, \vec{v}_2) \approx 1$ group. For any $\vec{v}_1 \neq \vec{o}$, the trajectories where $C(\vec{v}_1, \vec{o}) \approx 0$ have the highest rate. The agent concludes that the ideal approach for the Palmar reflex activation event should use matching directions for all vectors describing the motion and orientation of the hand, $\{\vec{g}_p, \vec{g}_f, \vec{m}_{p,f}, \vec{m}_{p,t}, \vec{m}_{f,t}\}$, and all of these parallel vectors should be perpendicular to the target's major axis $\vec{o}$.

Experiment 6.4 results. Using trajectories planned in this manner, 39 of 40 reaches are successfully completed and 21 of these activate the Palmar reflex. 14 of these activations result in a grasp. By choosing the best aligned candidate final node instead of the closest candidate node and then adjusting the entire final motion to match its gripper vector, the reliability of grasping is nearly tripled to 35%. Figures 6.4 and 6.5 provide additional comparisons with results from other learning


Choose Final Node
Adjust to Match Target Center
Construct a Preshaping Position
40
50
60
70
80
90
v
40
50
60
70
80
90
100
u

(a) (b) (c)

Figure 6.3: The agent plans modifications to the end of the trajectory, and defines a preshaping configuration q_p^*. Human intuition and the agent's learning recognize that all vectors describing gripper direction and the direction of motion should be near parallel, with all of these vectors near perpendicular to the target's major axis. The agent plans a final motion with these features in three steps: **(a)** The agent chooses n_f from the candidate final nodes (equation (6.1)) to minimize $C(\vec{g}_f, \vec{o})$. This image displays the palm mask (yellow) and hand mask (red) for the chosen n_f, along with the target mask (blue). A blue outline is used to show the boundary of the intersection between the hand and target. $\vec{g}_f$ and $\vec{o}$ are displayed in light blue and orange, respectively. **(b)** The agent uses $\hat{J}^+(n_f)$ to estimate the change in joint angles necessary to cause the image-space translation shown here. This translation improves the approach accuracy by aligning c_f^p with c^t by moving to q_f^*. **(c)** The agent constructs q_p^* by equation (6.3), which is predicted to have masks translated as shown from those for q_f^*. A final motion from q_p^* to q_f^* has aligned gripper and motion vectors.

stages.

Tabulated Results from Experiments 6.1 and 6.4:

Results	Grasp: Successful	Failed			
	Palmar Reflex: Activated			No Activation	
	Reach: Successful				Failed
	Grasp	Weak Grasp	Palmar Bump	Bump	Miss
Experiment 6.1 (5.2)	2.5%	0%	0%	17.5%	80.0%
Experiment 6.1 (5.4)	2.5%	0%	7.5%	42.5%	47.5%
Experiment 6.1 (5.5)	5.0%	0%	12.5%	60.0%	22.5%
Experiment 6.1 (5.6)	12.5%	0%	17.5%	70.0%	0%
Experiment 6.4	35.0%	0%	17.5%	45.0%	2.5%

6.4 Orienting the Grippers with the Wrist

6.4.1 Methods

For our Baxter robot, the joint angle setting q^7, which controls the most distal twist joint, “wrist 2” or $w2$, affects only a small portion of the wrist with a roll of the hand relative to the axis of the forearm without changing this axis. This alters the orientation and perceived shape of the gripper opening, but leaves the position largely unchanged. The primary modification is to the plane in which the gripper fingers open and close. Adjusting this is analogous to a human’s preshaping techniques to ready the hand for grasping an object, though simpler, as there are fewer ways to configure parallel grippers than an anthropomorphic hand. For a grasp to be successful, the cross section of the object in the gripper plane must be smaller than the space between the grippers. Additionally, the angle at which the plane and the object meet must not be so steep as to squeeze the object out of the grip. Intuitively, the most reliable grasp approach rotates $w2$ so that the gripper plane is perpendicular to the target object’s major axis.

6.4.2 Experiment 6.5: Copying successful wrist settings

Without intuition for the correct orientation, the agent must find another criteria for predicting the wrist orientation that will be most reliable. By this time, the agent has observed that, like the gripper aperture a, q^7 does not have a significant impact on the hand’s location in the image. This allows the agent to consider modifying q^7 without considering the graph nodes visited to change. In the same way, these changes do not conflict with the learned requirements for reaching or the previous grasping method of choosing n_f such that $\vec{g}_f$ and $\vec{o}$ are approximately perpendicular. In order to avoid new failures from introducing large, sudden rotations of the hand near the target, when a new q^7 is chosen it will be used instead of the stored q^7 value of all nodes in the trajectory n_{T_j}.

To begin, the agent repeats each successful grasp, with a linear search over values of q^7 to identify the longest continuous range where the attempt still succeeds. The center of this range will be saved as the ideal q^7 value for this example grasp. The agent will then retry each trajectory from Experiment 6.4. For each of these grasp attempts, the adjusted final configuration q_f^* is computed by equation (5.4), as before. Using the Euclidean distance between all other joint angles, $\langle q_f^1, \ldots, q_f^6 \rangle$, the nearest neighbor example grasp is found for the current trial. The grasp is attempted with the ideal q^7 value from this example and all other angles unchanged.

6.4.3 Experiment 6.5 Results

Over the same set of 40 object placements from previous experiments, this technique increases the number of Palmar reflex activations (Palmar bumps, weak grasps, and grasps) to 30 (75%), and grasps to 20 (50%), as shown in Figures 6.4 and 6.5. These increases come at the cost of one bump, where the target is now missed because the rotation of the hand prevents a collision that used to narrowly occur. In principle, any time new successes are achieved, they can be treated as new example grasps with ideal q^7 values to consider for trials with nearby target placements, allowing for further improvements to the success rate. However, in this training set only two still unsuccessful grasp attempts have different nearest neighbor examples than previously, and neither changes to a success with the new q^7 value. Iterations of using new nearest neighbors therefore end, but may be returned to in future work once more examples are available.

Tabulated Results from Experiments 6.1, 6.4, and 6.5:

Results	**Grasp: Successful**	**Failed**			
	Palmar Reflex: Activated			**No Activation**	
	Reach: Successful				**Failed**
	Grasp	Weak Grasp	Palmar Bump	Bump	Miss
Experiment 6.1 (5.2)	2.5%	0%	0%	17.5%	80.0%
Experiment 6.1 (5.4)	2.5%	0%	7.5%	42.5%	47.5%
Experiment 6.1 (5.5)	5.0%	0%	12.5%	60.0%	22.5%
Experiment 6.1 (5.6)	12.5%	0%	17.5%	70.0%	0%
Experiment 6.4	35.0%	0%	17.5%	45.0%	2.5%
Experiment 6.5	50.0%	2.5%	22.5%	20.0%	5.0%

6.5 Addressing the Difficulties of a New Environment

Within the duration of this work, the robot's physical location was changed significantly on multiple occasions. Generally, these moves coincided with new versions of the peripersonal space graph or other changes in methodology, and the effects of the environment change were not directly measurable. One location change occurred between Experiment 6.5 and the remaining grasping work in this chapter, and this change was an exception to this pattern, as it reused the Explored PPS Graph and continued using the same learned features for planning grasp actions. We found that the agent initially had drastically reduced grasping performance in the new environment. While the exact causality could not be confirmed, we determined several factors that were not controlled for

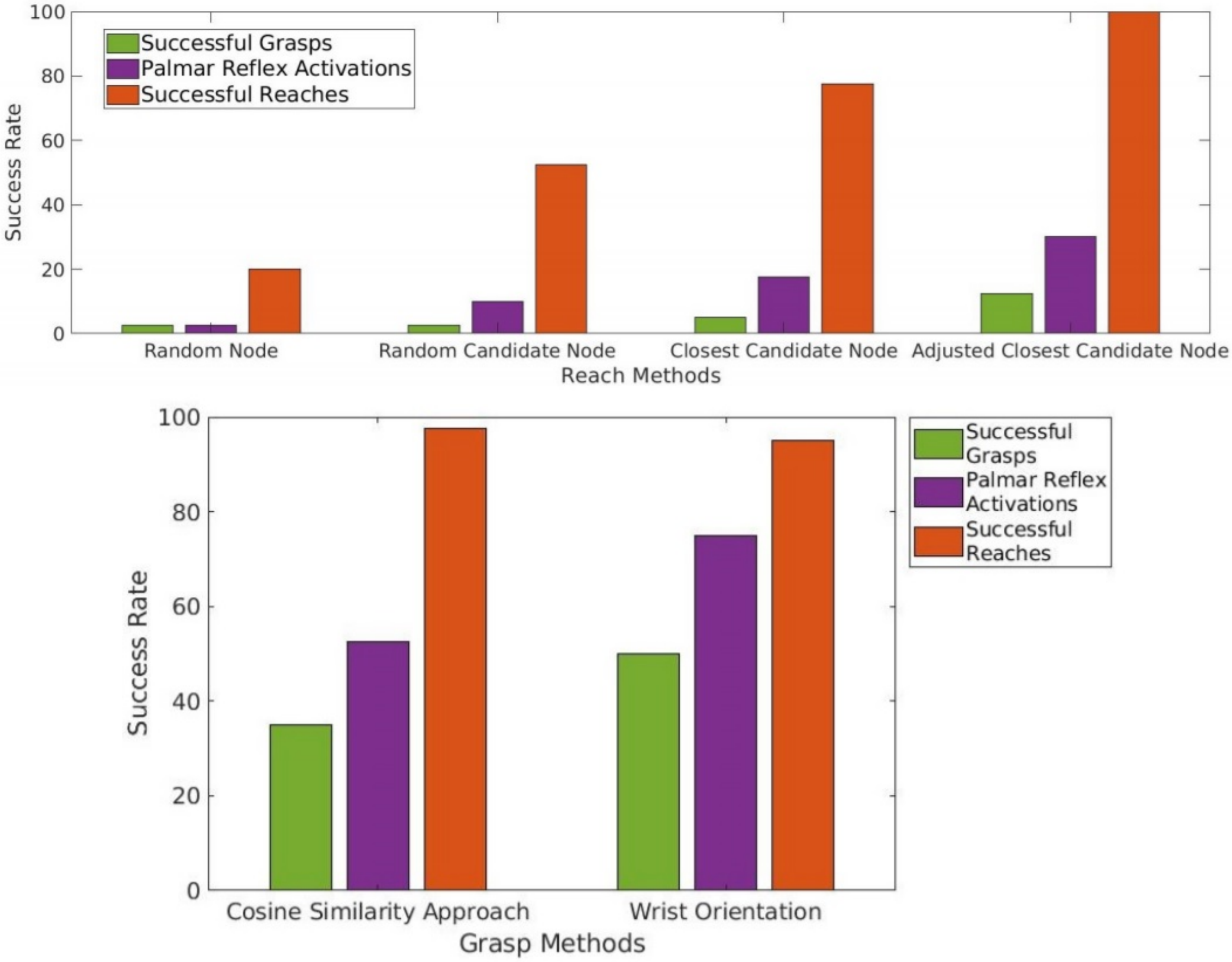
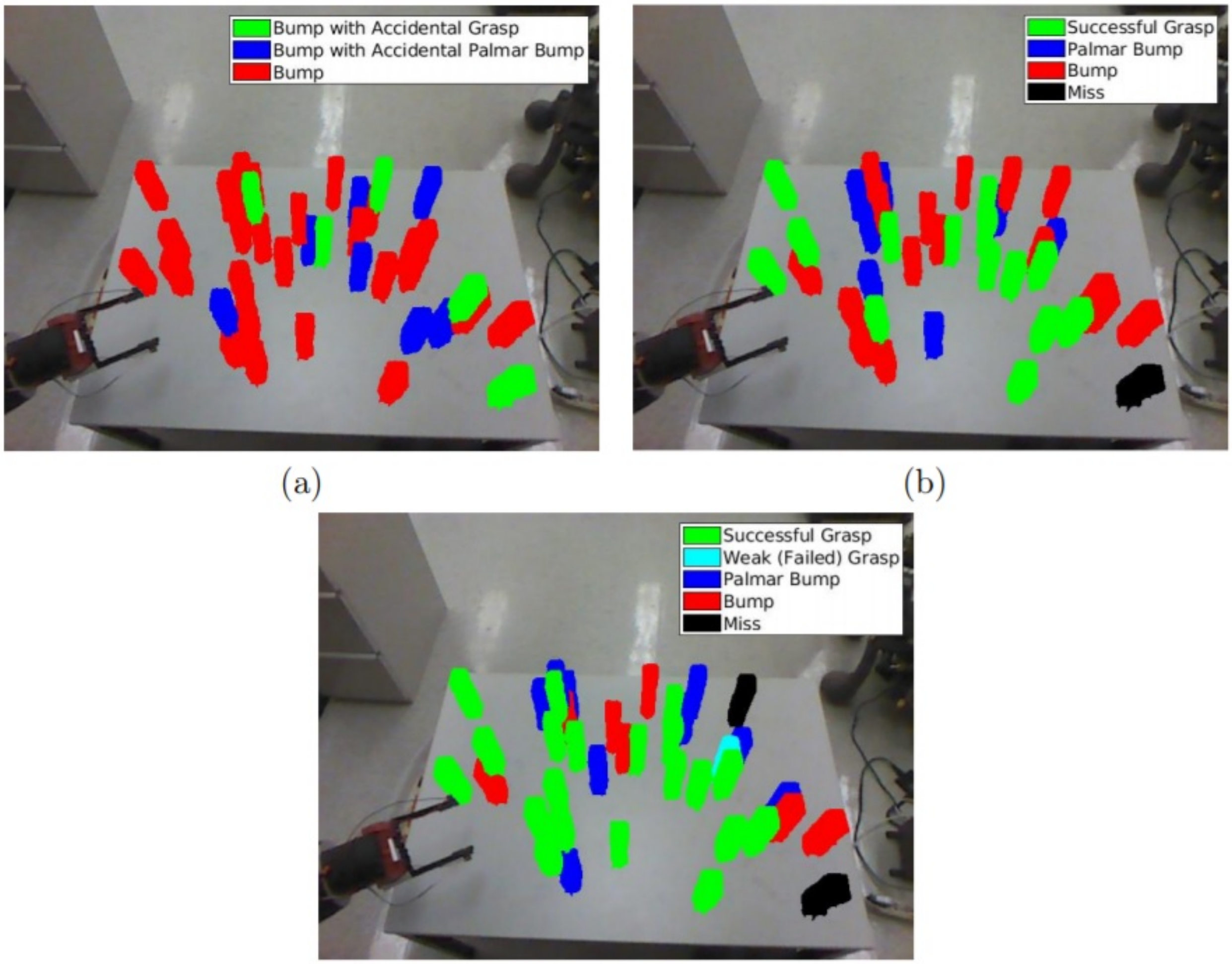


Figure 6.4: The top plot presents the overall results from the reaching methods as a baseline for the grasp action as found in Experiment 6.1. The final reach method, Adjusted Closest Candidate Node (Experiment 5.5), is always successful at reaching, but within these interactions only 12.5% are fully successful though accidental grasps. By considering additional features, the grasp methods in the bottom plot all achieve more than double this success rate for grasping with only modest decreases in reach reliability. The Cosine Similarity Approach Method (Experiment 6.4) aims to increase the number of Palmar Bumps, with n_f chosen from the candidates such that $|C(\vec{g}_f, \vec{o})|$ is minimized and with n_p replaced by a preshaping position so that all other cosine similarities are 1. Approaching with a motion parallel to $\vec{g}_f$ and perpendicular to $\vec{o}$ also increases the number of successful grasps. The Wrist Orientation Method (Experiment 6.5) further adds a technique to copy the most distal degree of freedom q_7 used at the nearest configuration to previously succeed, converting more bumps into Palmar bumps and grasps.



(a)

(b)

(c)

Figure 6.5: Spatial representations of the results of three methods for the agent's learned reach and grasp actions. Each shows a superposition of all placements of the single target object, colored according to the result of the agent's attempt to repeat an unusual event by executing a motion trajectory. **(a)** Experiment 6.1 uses the final reaching trajectories of Section 5.3 to successfully repeat the bump event for all target placements. Twelve of these reaches accidentally trigger the Palmar reflex, five of which become early examples of grasps. **(b)** Using cosine similarity features in Experiment 6.4 (Section 6.3), the agent modifies the final approach so that this motion causes significantly more Palmar bumps and more grasps are also observed. **(c)** In results from Experiment 6.5, the agent grasped from additional placements by changing the angle of the most distal joint, w2. The wrist orientation is copied from the final configuration of a trajectory that succeeded for a nearby placement (Section 6.4). The use of nearest neighbors applies best very close to existing successes, so most improvements can be observed in these areas.

during the move of the robot that either violated assumptions the agent had been allowed to make thus far, or otherwise made the grasping task more difficult. In this section, we discuss these factors and the steps taken to correct for the differences enough to return the grasp action to at least the level of reliability observed in the original environment for Experiments 6.1-6.5.

The theoretical formulation of the peripersonal space graph representation is such that it should not be necessary to recollect images when the appearance of the agent's environment changes. However, this requires strict enforcement of the assumptions that allow the agent to simplify the visual components of its tasks. In particular, it must be the case that the relationships between the positions of the workspace (table), the robot, and the camera are kept fixed, and the angle of the camera perspective must also be kept static. We also assumed that the agent would be capable of identifying the masks and depth ranges of the hand and target reliably, and the truth of this assumption can also vary in practice due to the environment. Due to violations of these assumptions, it was necessary to collect a new set of images and to edit the image processing code that allows the agent to extract object representations from the raw visual percepts.

While a PPS Graph can support changes to the workspace (either different placements of the table relative to the robot or camera or a different table entirely), the current implementation of the graph used rejection sampling to ensure that the nodes were gathered in an area concentrated above or near the table in its position at the time that the Explored PPS Graph was initially constructed. This means that changes to the table or its position relative to the robot will cause the set of nodes to drift from the most useful locations, as their positions will not change relative to the robot. We attempted to set the table in as similar of position and orientation as possible to minimize this effect, but it cannot be entirely corrected for without generating a new set of configurations for the nodes that are better fit to the exact new position of the table, or a set of configurations without a bias towards the table, for greater generality. This difficulty was increased by potential changes to the height and leveling of the robot, which was not controlled for in its move between buildings and resetting the brakes and anchors or the floor. An apparent consequence of the combination of these factors can be seen in Figure 6.6, where the nodes appear to be further from the table's surface overall. This increased displacement from the workspace may cause candidate nodes with nonempty intersection features to be more sparse, and requires larger adjustments using the local Jacobian estimates to bring the hand from its stored visual center to a target's perceived center. Both insufficient numbers of candidate nodes and extrapolating too far with the locally linear estimate of the Jacobian can make grasp attempts more prone to error.

Neither the theoretical PPS Graph formulation or the current implementation of it in the Explored PPS Graph can accommodate a change in the position or angle of the camera relative to the robot. Because the camera perspective is expected to be fixed throughout all experiments, when it is moved, the stored visual representations of the nodes become incorrect. When searching for nodes

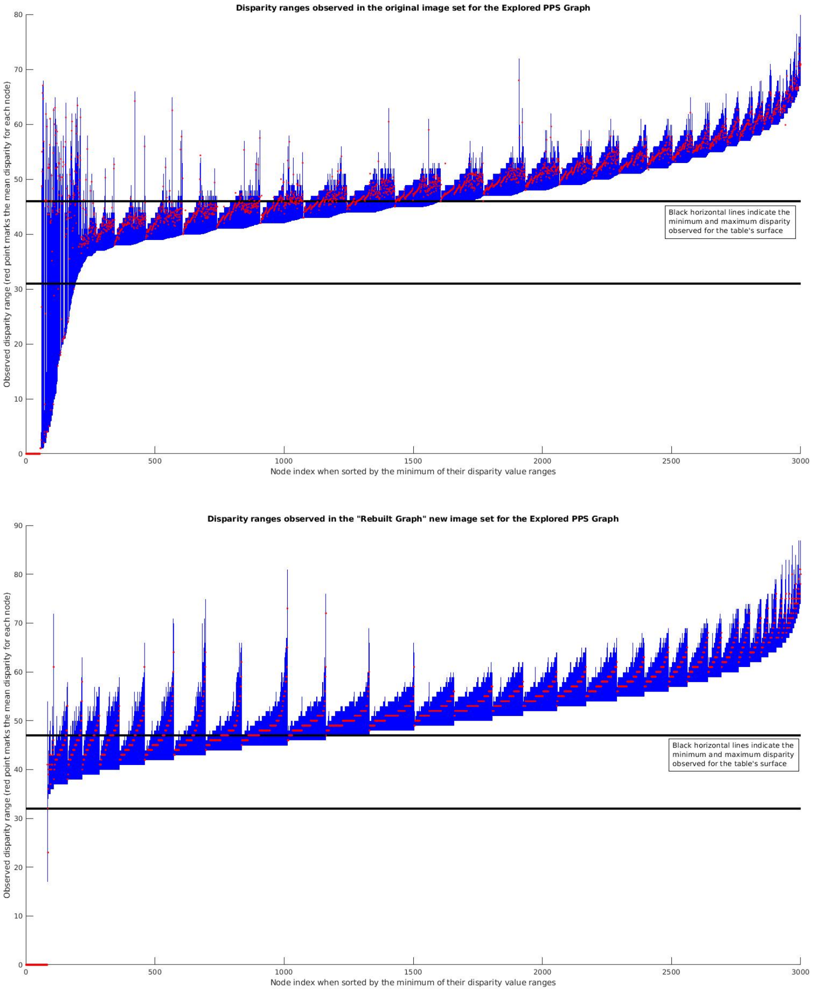


Figure 6.6: The top plot shows the palm depth range extracted from the original image set for all nodes, with the nodes sorted by minimum disparity in ascending order. The full range appears as a vertical blue line, and the mean disparity value recorded for that palm is plotted with a red dot. The horizontal black lines denote the minimum and maximum disparities observed for the surface of the table, meaning that blocks that can be reached and grasped from positions on the tabletop will occupy this range of [33,46] and just above it. The bottom plot shows the corresponding information for the palm depth ranges extracted from the new set of images after the move to the new environment and the camera angle was changed. The range of disparities for the table in these images was [32,47]. Despite the wider range occupied by the table in the new images, a smaller number of nodes appear to have the palm at this key range of disparities to coincide with objects on the table's surface. Excluding the nodes in the top plot with inaccurately large ranges and the undefined or zero valued ranges in the bottom plot, it appears that about 1400 nodes intersected with this range in the original data and only approximately 1100 nodes intersect with this range in the new data. This suggests that the position of the robot has changed relative to the position of the table, and nodes tend to appear further from the table in images than before. For some regions, this will reduce the number of final node candidates, and when candidates are found they are more likely to need larger adjustments to align the d component of their center with the d component of the center of a target on the table. Both cases make planning reliable grasps more difficult, and may explain some of the reduction in success rate that needed to be corrected for.

with nonempty intersection features, the agent will generate a significant number of false positives and false negatives since the target's position in the current images will not match up with hand positions in the same pixels of the stored images. That is, with a new camera angle, masks may occupy the same region in the current image as in the stored image, but those regions will no longer represent the same space in the environment, creating false positive candidate nodes. Alternatively, some nodes may be observed to have empty intersections when they would coincide with the target if the camera was correctly aligned, creating false negatives. The change in alignment also affects the relationship between current and stored center positions, and can cause the desired adjustments using the local Jacobians to be inaccurate. If the precise position and angle of both the new and old camera setup was known, a transformation between the current visual percepts and the stored visual percepts could be used so that intersections still carry their original meaning without false positives and negatives. This was not the case, and attempts to estimate the transformation would introduce additional noise, so the best solution was to gather a new set of images to associate with the existing configurations of the nodes of the graph.

In order to rebuild the existing graph with a new set of images, code was written that instructed the agent to move to each node's stored configuration $\mathbf{q}_i$ in order and record new RGB and disparity images while paused at each. The motion of the agent to each $\mathbf{q}_i$ used the built-in Joint Position Control mode, which commands each joint to a set point angle. The motion for each joint continues until its reported angle is within a threshold of 0.008 radians of the set point, and then comes to a stop. As a result, the angle reached by joint j may be equal to the original q_i^j, or up to 0.008 radians different. The agent may also reach a different joint angle due to any changes to the calibration or settings of the arm that may have occurred between experiments. For the highest accuracy, the new angle reached should be recorded and replace the original angle that the agent attempted to move back to. This may affect the local Jacobian estimates, which are based on the relationship between small changes in the joint angles and the corresponding small changes in image space. We introduced a small additional source of error by continuing to use the original $\mathbf{q}_i$ from the motor babbling of the first time the Explored PPS Graph was built. As a result, the angles may not be precisely those held in the new set of images, and the local Jacobian may overestimate or underestimate the impact of joint angle changes on the image-space center. We believe the noise introduced by this difference will not have a significant impact on the reliability of the agent's actions, as the motion controller was typically more accurate than the threshold required, and the threshold's magnitude is already less than 1/300th of the range for e1 (2.67 radians), the joint with the narrowest range, so it is unlikely that the actual configuration of the arm reached while taking the new set of images would produce a significantly different pose of the hand. However, we note the recording of the new joint angles reached as a best practice to eliminate this potential source of error for our use in future work or in any studies replicating the Explored PPS Graph method.

While the new set of images represent the same poses of the robot, there are significant differences from the original image set. These changes may contribute to the newly increased difficulty of the grasping task. As seen in Figure 6.7, the new camera angle appears to create a coordinate system for image space where each pair of dimensions is more correlated than when using the old camera angle. Each of the u and v components of the nodes' image-space centers is especially correlated with the disparity d of the centers, though the correlation between u and v is also increased. If the coordinates of the nodes are correlated, and thus conveying some redundant information, the agent is receiving less information about the 3D space than it would from coordinates on three independent axes. While it is not necessary for the agent to have a standard coordinate frame with independent axes, if the correlation is stronger when the new image set is used, the agent is tending to receive even less information as it plans actions than before, and this is a plausible explanation for some of the observed decrease in the success rate of the grasp action.

Another concern when moving the robot between buildings to a new environment is the change in lighting and the appearance of the background. Compared to pictures taken in the original lab setting, images in the new lab tended to have slightly darker lighting (at least as it registered in average RGB values for the camera) with a different white balance and shadowing from a different direction. For a robust image processing system, these changes would not cause the assumption that the agent can reliably segment the images to identify the masks, depth ranges, and centers for important portions of the self and the foreground objects. However, we have satisfied this assumed condition with a very simple image processing function that relies on custom ranges of RGB values to create the representations that are made available to the agent. Multiple ranges already existed for identification of each key segment (such as the yellow block and the green and blue gripper fingers), but the lighting in the new environment pushed some pixels in these object segments out of these ranges. Additionally, some aspects of the new background fell inside the original ranges, and needed to be ruled out. In order to improve the accuracy of the extracted binary image masks, rules needed to be added and modified to avoid false positive and false negative inclusions. While some anomalies remained, reducing the errors in these masks and the depth ranges that depend on the masks eliminates some failure cases that result from planning with poor data. While adjusting and adding rules for RGB ranges for the objects does involve expert knowledge from the experimenter and trial and error (which informed the changes when clearly incorrect masks were discovered), we claim that these updates do not detract from the autonomy of the agent. These updates were only necessary to maintain the agent's assumed (foundational) visual capabilities, and would not be needed in a more general and sophisticated computer vision method that could provide similar representations independent of the domain, the task at hand, or specific color assumptions.

For the new image set, we borrowed the mask location technique for the Improved Explored PPS Graph (Introduced in Section 3.2.2.3 primarily for consideration in future works, where it may

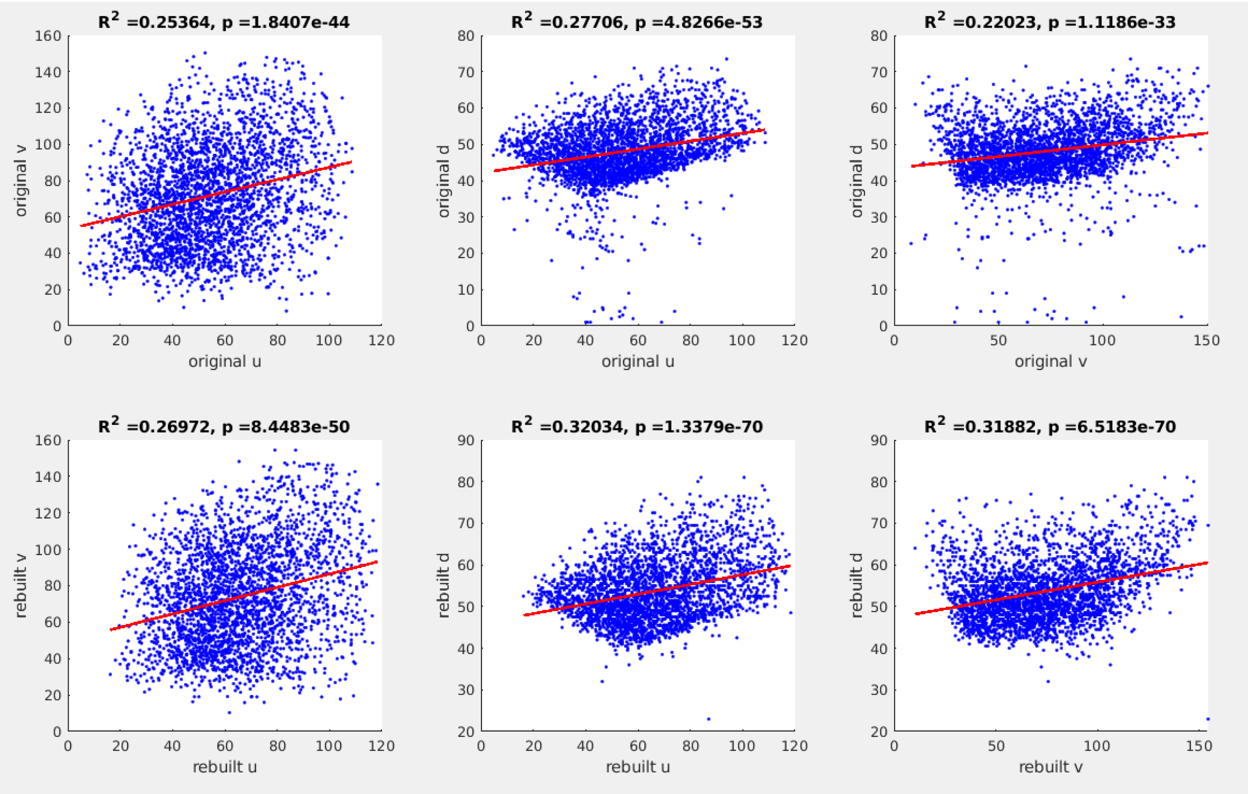


Figure 6.7: These plots show pairs of coordinates (u, v), (u, d), and (v, d) for all palm centers, with different coordinate pairs in each column. The top row displays information derived from the original image set for the Explored PPS Graph, while the bottom row displays information derived from the new image set taken for the Rebuilt Explored PPS Graph in the new environment. Comparing plots in the same column shows that the correlation between all pairs has increased when the new data is used (seen in the increased R^2 values), and increases most significantly for the pairs involving d. The p-scores reported are calculated using a distribution $F(p, n-p-1)$, where $p = 1$ in all cases and $n = 2945$ for the original data and $n = 2917$ in the new data, based on the number of nodes with defined palm centers. The very low p-scores indicate that there is very little chance that these coordinates would be determined to have this correlation when they are actually independent. The correlation between d and the other coordinates was expected as a line out from the camera to measure distance is not perpendicular to either axis of the RGB image. The observation that u and v are correlated is more surprising, but is explained in that for the agent to reached to the far right side of the image (where the palm center will have a high v coordinate) during motor babbling, it is more likely that the hand is also near the bottom of the image (where the palm center will have a high u coordinate) since it requires less additional extension of the arm. Some correlation is expected when using the nonstandard axes of the agent's image space rather than the independent axes of a standard Euclidean 3D space, and the agent shows a tolerance for correlated axes by performing well in the original environment. However, when all correlations are increased, this implies the new camera perspective provides more redundant information about the 3D space (at least at the sampled positions of the palm centers), and the agent has less information for planning. Therefore, this change is one of the hypothetical explanations for the drop in grasp reliability in the new environment.

allow additional features and methods to be used) that relies on color-coding the two gripper fingers, so that it is not necessary for the robot to grasp a block that fills the hand to distinguish the palm. In future work, this change will allow the agent to perform learning more naturally, with less fixed stages, as motor babbling with an empty hand allows the agent to experience accidental bumps and grasps during this time, and also would allow for motor babbling with the gripper aperture a. The images taken by the agent may also better convey the mask and depth range for the hand when the grippers are not adjacent to or partially obstructed by a brightly colored block.

In the new set of images, we observe that for 83 nodes, there were no valid disparity values reported within the palm mask, and the palm center cannot be defined due to the missing data for the d coordinate (for some nodes the u and v coordinates are also undefined, if the palm was not visible at all and had no mask). The local Jacobian estimate cannot be computed for 368 nodes given the new image set: these 83 nodes and an additional 285 nodes that have defined centers but fewer than 7 neighbors with defined centers. Both a center and a local Jacobian estimate are required for using a node as the final node of a reach or grasp, so the agent will not consider nodes without this information. This leaves 2632 nodes that will be considered for candidacy (with the subset of these nodes that become the final node candidate set still determined by the presence of nonempty intersections with a specific target). By contrast, the original set of images had 2950 nodes that could potentially be candidate final nodes. The reduced number of nodes with complete data may be a result of finding the palm representations without the grasped block - thin objects, such as the gripper fingers, appear to be more prone to registering with undefined disparity values. The use of pipe cleaners to increase the size of the grippers and to change their color and surface material that also seemed to be problematic for the camera provided more valid readings, but only up to this level. Another possible explanation is that the calibration of the camera may have deteriorated before these experiments were carried out, either during the long delay between experiments while the lab facilities were closed or when the angle was changed during the move or other occasions. If the camera was not able to properly align the disparity image with the RGB image, some or all of the disparities reported inside the mask may not have actually corresponded to the object. Values that did not belong to the object may have been too close or too far from the camera, or otherwise reported an undefined value, or they may have appeared as the disparity of those pixels in the base image taken without the hand present, and eliminated as noise values from the background. Note that since the agent still has the configurations recorded for all 3000 nodes, the agent may still plan trajectories that pass through these nodes that are excluded from all candidate sets. These nodes are only excluded for use as final nodes since it would be impossible to adjust their positions.

6.5.1 Methods

In order to evaluate the agent's grasp reliability after the change in environment and corrections for new factors were performed, we generated a new set of target object placements that the agent could attempt to grasp. At the end of Chapter 7, the agent will be asked to demonstrate its ability to perform a pick-and-place action to move an object from one qualitative location of interest to another (these locations are discussed further in Section 3.3). In addition to confirming that the grasp method can be transferred to the new environment, this new set of experiments will also be used to show that the grasp action is sufficiently reliable when grasping from those locations of interest for the pick-and-place demonstration to be feasible. To accomplish this, instead of generating random object placements across the full tabletop, trials will begin with the object placed by the experimenter into one of the locations at random coordinates (these coordinates are not available to the agent, which still relies on its visual percept to construct a representation for the target object in image space instead of standard axes for the 3D environment). Since the locations are rectangular, the coordinates will specify a displacement of the corner of the object between 0 and 11 inches down and between 0 and 8 inches right of the location's top-left corner. The target object will be the same yellow rectangular prism block as in previous experiments, and will always be placed in the same upright orientation with the second-longest dimension parallel to the table's longer dimension, presenting one of the largest faces toward the robot.

For experiments from this point on, the agent is given the ability to refuse to attempt a grasp if there are zero final node candidates[1]. This rejection occurs when the placement of the object is represented by a target mask and depth range that do not intersect with the palm mask and depth range for any node that is considered for candidacy. This replaces the agent's previous strategy of choosing the closest node in image space when there are no candidates. The previous strategy was sufficient for making reaching fully reliable, including to positions that had no candidate final nodes, but was typically not sufficient for producing successful grasps, since the additional factors, such as alignment of the final node, were not considered. As the agent has a clear criteria that identifies these trials (no candidates) and given the observation that these trials, if attempted, tend to result in misses or bumps that do not activate the Palmar reflex, it is feasible and more productive for the agent to reject these trials. As the methods will require further reducing the set of nodes eligible for candidacy, rejections will be based on the smallest set of nodes that can be considered, so that each success rate can be evaluated on the same set of object placements.

The evaluation took place over 40 trials with attempted grasps. The agent was presented with

[1] In the following discussion, "no candidate nodes" indicates when the set of candidates is empty after the first pass that identifies all nodes with the most desirable intersection feature combination, $D(p_f) \cap D(t) \neq \emptyset$ and $p_f \cap t \neq \emptyset$. While a strategy of expanding the set to include nodes with decreasingly desirable features until it is not empty and then using a modified configuration from the closest candidate node was sufficient for reaching, use of the nodes from the expanded set of candidates does not produce any successful grasps.

object placements in the green location L_1 until it had attempted 20 grasps from L_1, and then the agent was presented with object placements in the blue location L_2 until it had attempted 20 grasps from L_2. The agent accepted all placements in L_1 where the graph tends to be more dense, but rejects 9 placements in the more distant location L_2, one of the regions near the table where the graph is at its most sparse. For each experiment in this set, 49 grasping trials were initiated with an object placement and observation, 40 were attempted, and 9 were rejected by the agent.

6.5.2 Experiment 6.6 - Grasping with New Images and Masks

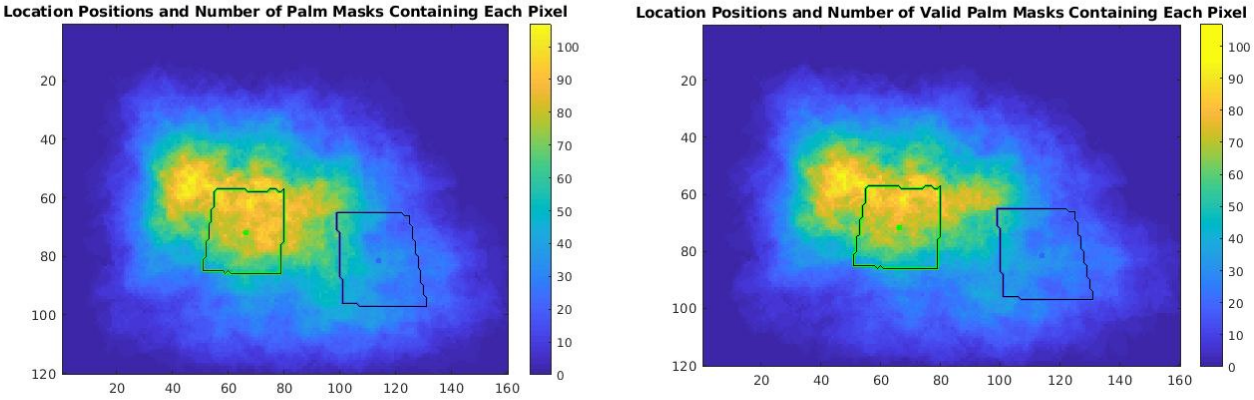
The agent attempts 40 grasps as described in order to measure the baseline performance of the grasp action when relying on the new image set and the code with new color thresholds and the mask-finding method from the Improved Explored PPS Graph (though in this work the agent does not make use of the individual green and blue gripper masks except to find the palm mask and vector without a block in the hand while motor babbling). The agent's decision making method is exactly the same as in Experiment 6.5, and merely differs due to reliance on new data and a larger set of nodes with missing data that must be excluded. The agent will record each successful grasp as well as the types of partial successes and failures observed and the number of rejected trials.

6.5.3 Experiment 6.6 Results

Prior to this experiment, grasp attempts using the original stored images or representations of the nodes with the agent in the new environment prevented any grasps from succeeding. These failures appeared to be primarily due to the change in perspective and the issues this introduced with false positive and negative intersections. However, in this experiment the agent was able to observe some level of success once the new image set was taken and the image processing code was revised to allow the palm mask and vector representations to be extracted.

For objects in the green location L_1, the agent successfully grasped in 2 (10%) out of 20 trials, with no rejected trials. For objects in the blue location L_2, the agent successfully grasped in 11 (55%) out of 20 trials, with 9 rejected trials. These results establish a baseline reliability of 13/40 (32.5%) for the grasp action when using the previous learned method and the new data. The results of this experiment are compiled with those of the remaining experiments in this section in Table 6.1.

Of the failure cases, 8 were Palmar bumps and 19 were bumps that did not activate the Palmar reflex. No miss results were observed, so the reach component of the grasp was still reliable. The limited number of Palmar reflex activations suggests that the approach is not aligned well enough or moving to a precise enough position to surround the object with the gripper fingers. We also see a departure from the pattern of results so far, where the agent had tended to be more successful in regions of space where the graph nodes were dense. This density should allow the agent to have a



(a)

(b)

Figure 6.8: **(a)** A heatmap displaying the number of palm masks (out of all 3000 nodes) contain each pixel (u, v) when derived from the new image set for the Explored PPS Graph. These numbers reflect palm masks at that position and any disparity, so the number of masks at the level of the tabletop or just above to interact with blocks on its surface will tend to be lower. The heatmap is overlaid with the boundary and center of each of the two locations of interest. The presence of these boundaries demonstrate that L_1, the green location, is at one of the most dense areas of the graph, and that L_2, the blue location, is more sparse because only when motor babbling had the arm close to fully extended across the body were nodes recorded there. The periphery of the image has even fewer nodes due to the biasing of the motor babbling to generate useful nodes where the hand was fully in view and above the table. The density of the graph is important for planning reach and grasp actions, as it determines the number of nodes that will have intersections with a given target placement, and can be used to plan the final pose. **(b)** The same heatmap and overlaid locations, but only counting the palm masks of nodes that would be considered for candidacy after the more extensive exclusions in Experiment 6.8. The dense and sparse regions are generally the same, with lower numbers of nodes throughout. Note that this heatmap uses the same scale as the original so that it can be directly compared by colors, even though its range is [0,96] instead of the original range, [0,107].

larger set of final node candidates, and within the variation of these candidates at least one should have reliable features that can be identified. However, the agent was much more successful in L_2, a location on the far end of the table that is sparsely covered by the graph, than it was in L_1, near the center of the table and the graph where there is a high concentration of nodes. The difference in graph density near the two locations is visualized in Figure 6.8. Based on this observation, we hypothesize that the agent is still receiving inaccurate data, and that a significant number of nodes with inaccurate representations have desirable-looking features. In the following experiments, we will attempt to improve the image processing so that the inaccuracies can be minimized or nodes that appear to retain inaccurate information can be eliminated from consideration as final node candidates.

6.5.4 Experiment 6.7 - Grasping with Trimmed Masks and Depth Ranges

One possible cause of the agent's poor choices of final nodes, especially where the graph is dense and may have numerous false positive candidates, may be the representations of the palm when computed with the new method. The new palm masks and depth ranges tend to be larger than when they were computed by the old method, and it appears to be the case that the new sizes are too large, including too many pixels and disparity values to accurately convey the grasping region.

We can determine that the palm masks are larger than before by examining the distribution of differences of areas seen in the new and old mask for each n_i, with area measured in the number of pixels included. This can be observed in the first panel of Figure 6.9. We found evidence that the increased size, while possibly explained by the new camera angle for some nodes, for many nodes involved the inclusion of pixels around the periphery that were included but were actually parts of the gripper fingers, the base of the hand, or a part of the palm opening but past the end of the grippers. In the remaining panels of Figure 6.9, we examine the distribution of changes in palm mask size when the new masks have a variety of binary image erosions applied to them. We found that the distribution is best centered around 0, and thus the areas are most similar, with an erosion using a disk of radius 1. The eroded palm masks will be used in this experiment to determine if an intersection is present.

We can perform a similar examination of the changes to the depth ranges for the palms when they are based on the new disparity images and take the values for the set of pixels within the new palm masks. While the original method attempted to remove all error values and apparent noise, it is still possible for some of the included disparity values to not represent the palm correctly. We switched to use the original resolution of the disparity image rather than the downsized image, to avoid averaged values from pooling the pixels together. Since it is typical for disparity values along the table, hand, or target object to increase by less than one unit per pixel, we also exclude any reading that is at least three units different from one of its neighbors as a likely error. Given how many pixels must typically be traveled through to see a one unit change in disparity, it would be quite surprising to see this large of a change in one step legitimately, and it is much more likely that it arose as part of an error or disparity reading from a neighboring region or object. We also continue to remove any 0 values, which are reported in a variety of cases: when the object is closer than the minimum distance from the camera, when the object is further than the maximum distance from the camera, and when the data is missing, which seems most common with narrow objects and boundaries, as well as objects presenting certain materials, colors, or angles. Still, it is possible for the agent to have depth range to include extremes based on the surface of the table, the floor, or other objects in the environment, and this can occur either from reading disparities from false positive pixels in the palm mask, or from pixels that were part of the object in the RGB image but not the depth image, which is offset and in some places not calibrated to correct for this adequately. When

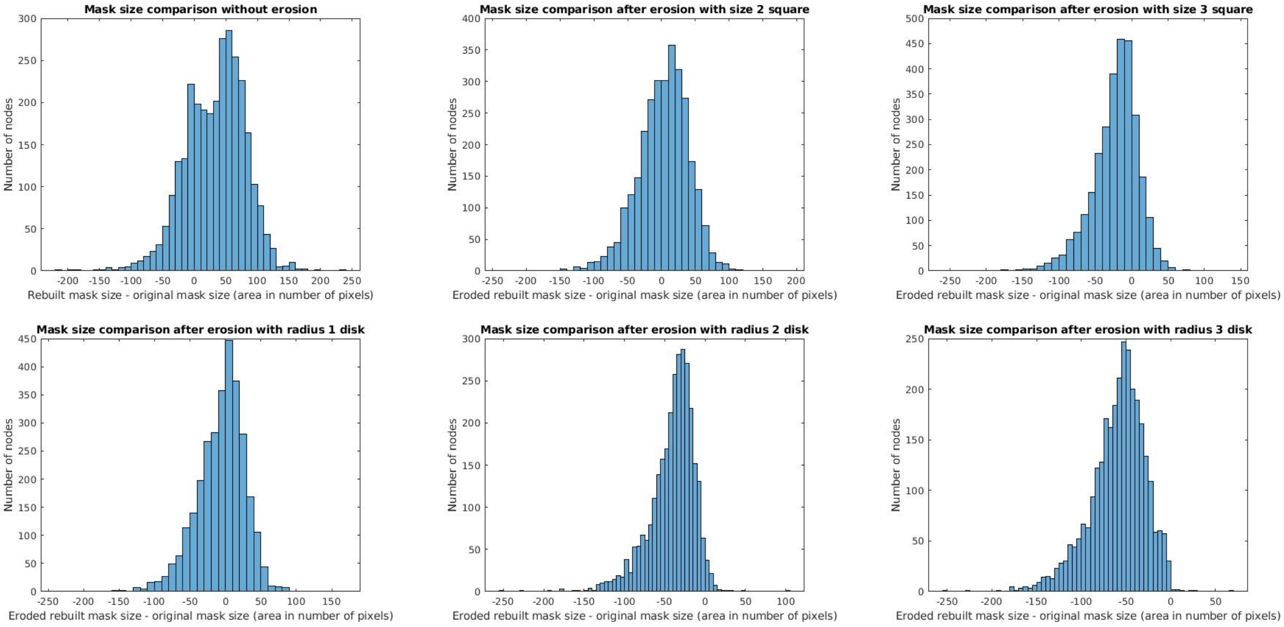


Figure 6.9: The top left panel shows a histogram of the differences in the area of each node's palm mask when derived from the image taken for "rebuilt" graph in the new environment and when derived from the original image. The range of this distribution is [-212,238], with the extreme values coming from 5 cases where a node was not visible in one set of images but was in the other due to the change in camera perspective. In general, the new palm masks tend to be larger, by a mean of 33 pixels, due to differences in the raw images and the image processing methods. The agent was observed to fail grasp attempts after selecting nodes that appeared to intersect with the target object, but did not produce a successful reach or grasp when used. This may be due to the larger palm masks over-representing the size of the palm, and the selection of final nodes that were only candidates because of false positive intersection features. To compensate, we added a final step to the image processing algorithm that extracts the palm mask representation. Once the palm mask is extracted, a binary image erosion with a small disk or square is performed to create the final mask. The remaining panels show the resulting histograms in the difference of areas, with the bottom left panel displaying the best result (Note that erosion with a square of side length one produces no change in the binary images, so its results histogram is omitted). When the erosion is performed with a disk of radius one, the distribution narrows and centers nearest to zero, better matching up with the original areas of the palm masks. These eroded palm masks will be used to check for the presence of an intersection with the target mask in all experiments from this point on.

any of these problems occur, the depth range for the palm can become much wider than accurately represents the hand. Fortunately, each problem typically introduces only a small number of pixels, and the accuracy of the depth range can be improved by introducing a rule to exclude these outliers. Similar to the areas, the distribution of depth range widths (the maximum disparity value minus the minimum disparity value) is initially skewed to be larger for most nodes. By calculating the mean and standard deviation of the disparity values reported for each node's palm, the creation of the depth range representation can set bounds for the depth range to exclude outliers. In the original image set and representations, the depth ranges ignored values that were more than three standard deviations from the mean. This same technique leaves the ranges still larger than the previous ranges, but ignoring values that were more than two standard deviations from the mean creates the most similar distribution. This trimmed depth range where the minimum and maximum are capped at two standard deviations below and above the mean will be used to check for depth intersections when determining if a node is a candidate. As before, any candidate must have a defined center, vector, and local Jacobian estimate, so 2632 nodes will be considered.

The agent will perform the same 40 grasp attempts with the final node candidate set determined using the new eroded palm masks and trimmed depth ranges. Using the same experimenter placements of the target object will allow a direct comparison of the performance without considering if there was a benefit or detriment caused by random variation in the placements. As a result, the agent will again reject 9 of the placements in L_2, and will perform the 40 grasp attempts over 49 initiated trials.

6.5.5 Experiment 6.7 Results

The results were successful grasps in 10 (50%) out of 20 attempts from L_1 and 10 (50%) out of 20 attempts from L_2, for an overall success rate of 20/40 (50%). This is already equivalent to the performance observed in the original environment of the Explored PPS Graph during Experiment 6.5, and demonstrates that the systematic changes to the palm representations were influential and allowed the agent to perform significantly better with the improved data. However, the failure cases included 10 Palmar bumps, 8 bumps, and 2 misses, which suggests that only half could be considered near success. Many of these more pronounced failures seem to involve choices of final nodes or local Jacobian modifications to the final pose that are not reasonably in line with the agent's learned methods. When this is the case, the agent is often still acting on incorrect or incomplete data, and the following experiment will attempt to rule out cases where the erosions and trimmings have not sufficiently reduced the number of false positive candidates and their apparent quality. The full results of this experiment are repeated in Table 6.1, alongside the other experiments from this section for comparison.

6.5.6 Experiment 6.8 - Grasping with Additional Excluded Nodes

While the agent's grasp action is back to the 50% reliability observed in the previous environment, it can still be observed that it is choosing some final nodes that are well off and qualitatively different than those it chooses when it is successful or close. This suggests that for some nodes, some bad data has slipped through (or that some component of the simplistic methods for extracting the image features don't work well for nodes with noisy data, small visible portions of the hand, or other problem cases, and this produces a poor representation). These nodes will be unreliable whenever they are chosen, but cannot be ruled out by the agent because the representation it is given for them is wrong (violating our assumptions that the vision system, while simple, works and can be trusted by the agent). These nodes can be especially problematic because the incorrect representations are unique, and their unrealistic features can be especially desirable for selection. This provides a plausible explanation for the continued observation that having higher graph density near a target is still not helpful for grasping in the new environment. In particular, having a higher number of candidates increases the chance that at least one is a false positive, and that its incorrect features will appear reliable and lead to selection.

While systematic changes to the image processing and feature extraction methods risk leaving in some errors or eliminating some values that are valid but unusual, we choose to use them rather than manual revisions to the representations for individual nodes to prevent any potential biasing from the experimenters, as well as to eliminate potential sources of human error and a time-inefficient process. In this case, we use an expanded set of criteria to exclude nodes from consideration as candidates. This set will include the past criteria where missing data would make the nodes impossible to use or impossible to predict what their use would accomplish, as well as new criteria where the representation is so unusual that the agent should not rely on it being valid and select those nodes. The criteria are not mutually exclusive, and in some cases are subsets of each other. The full list is still used for efficiency, as some of the less exclusive rules can be evaluated more quickly. The list is as follows:

- Existing Criteria:
  - 1 node is rejected due to being the home node, where the hand is assumed to never intersect with the starting position of the object so that it can be observed without obstruction.
  - 5 nodes are rejected due to being bad positions that should have been rejected during motor babbling, but were not. Either because of the nodes themselves or the set of edges connected to them, a high portion of moves to these nodes require a slight or glancing collision with the tabletop to reach them.
  - 83 nodes are rejected due to having undefined palm centers or palm vectors
  - 368 nodes are rejected due to having undefined local Jacobian estimates (includes the 83 above)
- New Criteria:
  - 267 nodes are rejected because both gripper fingers aren't visible (green is not visible in 54 and blue is not visible in 216). This prevents an accurate segmentation of the palm since it could be on any side of a single gripper (or overlapping it at a different depth), and with no visible gripper the agent has no reliable information about the palm.
  - 637 nodes are rejected because they do not have a reliable estimate of the palm vector. In the case of the 267 nodes above, this is because the palm cannot be located. For the other nodes, the two estimates (one is a vector drawn from the center of the base of the hand through the center of the palm, and the other is a vector perpendicular to the line between the centers of the green and blue gripper fingers) for the palm vector's direction to not sufficiently agree, so the observation is deemed unreliable. The measures of agreement are shown in Figure 6.10, where a cosine similarity of 0.707 is chosen as the threshold. Cosine similarities below this threshold are both uncommon and imply significant disagreement of at least 45 degrees.
  - 256 nodes are rejected because their palm depth range has $\max(\text{disparity}) - \min(\text{disparity}) > 10$ units. This is after the image processing code was updated and the trimming to within two standard deviations of the mean was performed, so as seen in the histogram in Figure 6.11, 10 units is abnormally large. Examples of these exceptionally large ranges were observed to be cases where an anomaly mask-finding problem or a significant offset between the RGB and depth images that resulted in a large region of incorrect depth values being present. This region would be large enough to have internal pixels

that would not be removed by erasure or by ignoring depths that changed too far from the neighboring pixels, and also large enough to spread the standard deviation and make the trimming ineffective. It is possible that some nodes where the hand appeared especially large to the camera could have legitimately had a depth range spanning more than 10 units, but these would all be high above the table and close to the camera, and would never be good choices for final nodes to reach or grasp objects on the table's surface. Therefore, it is not necessary to be concerned about ruling out a small number of nodes where this range is accurate, as it will not affect the agent's decision making.

After all criteria are applied, the agent has excluded 952 nodes and considers a set of 2048 nodes that can be final node candidates if they satisfy the conditions of nonempty intersections for mask and depth ranges for the specific placement of the target object. The agent will be evaluated again for the same 40 attempted grasps, both to determine the success rate with the additional exclusions and to see if the failure cases are closer to success in a way that suggests the agent is not being asked to use unreliable data that violates the assumptions of reliable vision. Note that the agent's decision making as it plans each grasp attempt will still be identical to the method in Experiment 6.5, and simply uses the new data and measures to correct for the challenges of the new environment and images.

6.5.7 Experiment 6.8 Results

With the additional nodes excluded from consideration, the agent was successful in 12 (60%) out of 20 grasps attempted from L_1 and 8 (40%) out of 20 grasps attempted from L_2, with the additional 9 rejected trials when the block was placed into L_2. Overall, the agent's grasp reliability has not improved from 20/40 (50%), but the full results of this experiment (Table 6.1) illustrate desirable effects of the new criteria. The failure cases include significantly more Palmar bumps (14 rather than 10) and one less miss (one instead of two), suggesting that the agent is closer to success when it fails. When generalized from these specific initial placements of the target object, this tendency to be more towards successful grasping is likely to produce additional successful grasps.

We also see in these results a return to the expected pattern, where the agent performs better in dense regions of the graph where it has more final node candidates available. Intuitively, a larger set of candidates increases the chance that one will be oriented nearly perpendicular to the target's major axis. Further, when the block being in a region that is only sparsely covered by the graph, a higher portion of the candidates tend to have small peripheral intersections with the target, and if one of these candidates is selected the creation of the final and preshaping poses requires large, potentially inaccurate, adjustments with the psuedo-inverse of the local Jacobian estimate. In the first set of trials for the new environment in Experiment 6.6, the agent grasped 10% of objects from

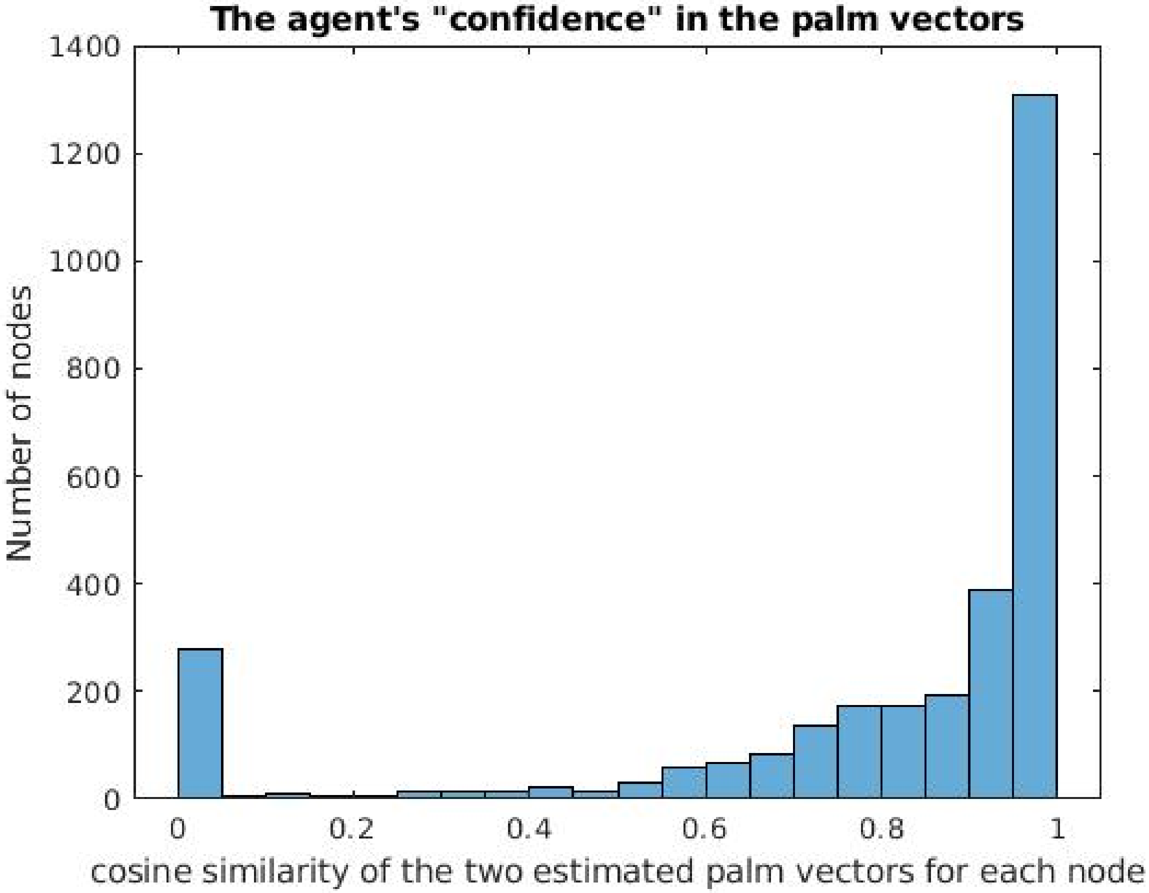


Figure 6.10: The cosine similarity between the two estimates of the palm vector's direction in 2D (u, v)-space can be used as a measure of confidence that the resulting vector is a reliable representation of the orientation of the hand in the agent's simple understanding of space. There is a large bar at zero due to the 267 nodes where one or both gripper fingers are not visible, and the estimation method that relies on them cannot be computed. With one of the estimates undefined, the cosine similarity is also undefined, but is set to zero as the agent has no confidence in those nodes - the estimates cannot be evaluated through comparison. With the exception of these nodes, it can be observed that the confidence is typically very high, near the maximum cosine similarity of 1. (Note that the cosine similarity is never negative because the estimate that is perpendicular to the line between the gripper fingers can be flipped to have the same sign as the estimate based on the palm's relationship to the base of the hand.) The number of nodes in each bar drops off in a long tail near the selected threshold value of 0.707, which corresponds to a 45 degree difference between the estimates for the vectors. Because cases that fall below this threshold are uncommon and show that two generally reliable ways of estimating the vector have produced very different angles for the same pose, these nodes will be excluded from consideration as final node candidates. If they were considered, it would be impossible for the agent to know which of the estimates, or some point between them, is an accurate portrayal of the orientation, and the fitness for grasping could not be determined.

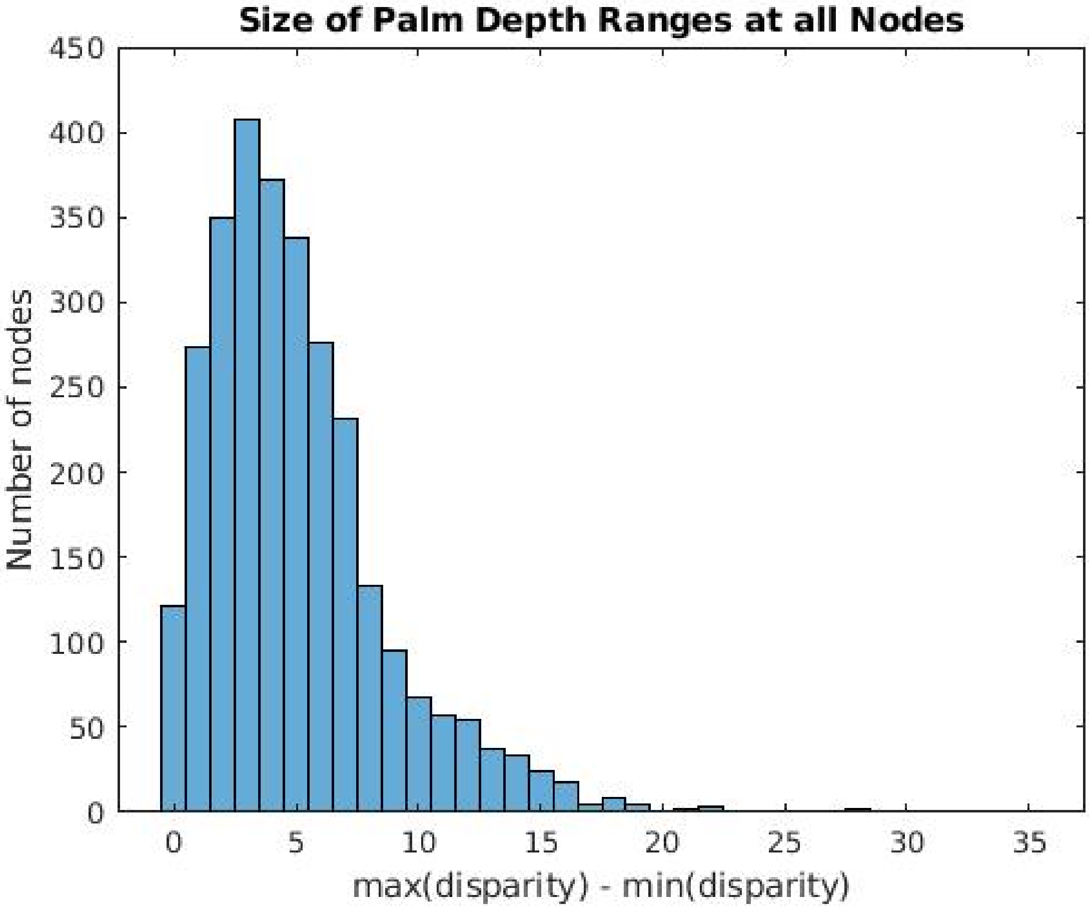


Figure 6.11: A histogram of the sizes of the palm depth range of all nodes, measured as the maximum disparity observed within the masks for the gripper fingers minus the minimum disparity observed within the masks. These sizes are calculated after all measures have been taken to eliminate noise and error readings and the extremes have been trimmed to be at most two standard deviations from the mean disparity value (with both the mean and standard deviation calculated for each individual node). For a large majority of nodes, the size of the palm depth range was between 0 and 10 units. Once the size is greater than 10 units, the hand is spanning an abnormally high range of disparities, and these nodes are excluded from consideration as final node candidates starting in Experiment 6.8. While it is possible for these large ranges to be accurate representations, it is much more likely that they are the result of errors that could not be eliminated in the systematic procedure, and would have to be corrected for manually and on an individual node basis in order for the agent to predict the result of their usage. We want to avoid this level of experimenter involvement with individual nodes, so the agent will simply not be allowed to use these nodes that may have inaccurately reported depth ranges.

the densely covered L_1 and 55% of the objects from the sparsely covered L_2, which conflicts with this intuition due to errors in the data that appear to become more frequently encountered and more influential with more candidates. In this experiment, the agent had 60% success in L_1 and 40% success in L_2, in agreement with this intuition.

We do see an undesirable side effect that the performance in L_2 has decreased since Experiment 6.6. As seen in panel (b) of Figure 6.8, the graph may be too sparse across too much of L_2 for more reliable grasping from that location. The exclusion criteria were made generally and without consideration for individual nodes or their location in the graph, configuration space, or image space, and may be too aggressive, removing some nodes that could be suitable for use in grasp attempts but with data that could not be verified with the automatic process. These unnecessary removals have little impact where the graph is dense and another node will be expected to be nearly as reliable according to its features, but where the graph is sparse an unnecessary removal may take away the only suitable choice, and the agent will fall back on a candidate that appears much less reliable. Additional tuning of the criteria may allow fewer nodes to be excluded and improve performance when grasping from sparsely covered regions like L_2. We leave this issue to be addressed in future work, and continue forward with this set of exclusion criteria due to this result's clear advantage over the other results in this section: the agent no longer appears to be misled by unreliable image data, and is not failing by wide margins after selecting unreasonable final nodes and modifications.

6.6 Revisiting Cosine Similarity Features for Alignment

6.6.1 Methods

With the change in environments sufficiently accounted for so that the success rate matches the rate prior to the move, the agent can continue to attempt to add features that would make grasping more reliable. One area with room to explore is the alignment of the hand for the grasp approach, the final motion in a grasp trajectory that depends on the preshaping pose and the final pose. Since Experiment 6.4, the agent has selected the final node from the candidate set that is most perpendicular to the target's major axis, that is, the node n_f such that the absolute value of the cosine similarity $C(\vec{g}_f, \vec{o})$ is minimized. This cosine similarity was prioritized because the agent has no method for modifying either of these vectors, but can use the psuedo-inverse of the local Jacobian estimate $J^+(n_f)$ when constructing the preshaping pose and final pose so that all other cosine similarities are expected to be approximately 1, as desired. However, this assumes that $J^+(n_f)$ and the calculated configurations are perfectly accurate, and will produce the desired positions in image space that would make the vectors fully aligned.

$J^+(n_f)$ is only an estimate, and only intended to be used locally, with growing error as the desired image-space coordinates become further from the original image-space center of the palm

Results	**Grasp: Successful**	**Failed**			
	Palmar Reflex: Activated			**No Activation**	
	Reach: Successful				**Failed**
	Grasp	Weak Grasp	Palmar Bump	Bump	Miss
Experiment 6.6 (L_1)	10%	0%	25%	65%	0%
Experiment 6.6 (L_2)	55%	0%	15%	30%	0%
Experiment 6.6 (All)	32.5%	0%	20%	47.5%	0%
Experiment 6.7 (L_1)	50%	0%	25%	25%	0%
Experiment 6.7 (L_2)	50%	0%	25%	15%	10%
Experiment 6.7 (All)	50%	0%	25%	20%	5%
Experiment 6.8 (L_1)	60%	0%	20%	20%	0%
Experiment 6.8 (L_2)	40%	0%	50%	5%	5%
Experiment 6.8 (All)	50%	0%	35%	12.5%	2.5%

Table 6.1: Tabulated results for grasp attempts in the new environment using the new image set and feature extraction code (Experiment 6.6), using those as well as eroded palm masks and trimmed palm depth ranges (Experiment 6.7), and when with those methods as well as the exclusion of nodes with potentially invalid data (Experiment 6.8). The results for each experiment are presented in 3 sets, for the 20 attempted grasps from the green location L_1 that is densely covered by graph nodes and 20 attempted grasps from the blue location L_2 where the graph is sparse, as well as the overall performance out of all 40 attempts. There were 9 rejected object positions in L_2 where a grasp was not attempted, and additional trials were generated so that 20 attempts (and 40 total attempts) were made. The lower success rate in Experiment 6.6 shows the continuation of the added difficulty of grasping after a change in environment with uncontrolled factors that cause violations of the assumptions about the image data. These are corrected for in Experiments 6.7 and 6.8, with the strongest result coming with the exclusion of nodes as the failure cases are closer to successful grasps.

c^p_f. $J^+(n_f)$ is used to move the preshaping position backward along the direction of $\vec{g}_f$ according to Equation 6.3. However, if the original position of the final node was such that $C(\vec{g}_f, \vec{m}_{f,t})$ is low, the modified preshaping position is moving especially far from c^p_f. This is because the direction from the original final position stored in the node to the target's center is going to be different from the intended direction from the modified final position to the target's center, with the two positions of the palm on different sides of the target. This larger modification will be less reliable, and the inaccuracies will tend to produce less aligned vectors. A particularly large move may also cause unmodelled changes to the orientation of the grippers, causing an additional misalignment.

This can be addressed by considering $C(\vec{g}_f, \vec{m}_{f,t})$ as a new feature. The agent has observed previously that the local Jacobian and its pseudo-inverse are more reliable in smaller neighborhoods around the observed pair of configuration and image-space coordinates. We assume the application of this lesson to this situation, such that the agent will prefer to maximize this cosine similarity so that the original alignment is better and requires a smaller adjustment, the results of which can be more reliably predicted and used for better grasping success.

The agent cannot always maximize $C(\vec{g}_f, \vec{m}_{f,t})$ and minimize $|C(\vec{g}_f, \vec{o})|$, as these will often be done with different candidates. This suggests a set of experiments. Experiment 6.8 has already evaluated the agent's performance when it only considers perpendicularity to the target's major axis. Experiment 6.9 will evaluate the reliability of grasping when only the alignment between the palm vector and direction of displacement from the target is considered. Finally, Experiment 6.10 will suggest a straightforward combination of the two measures of fitness and evaluate the performance when both alignment and perpendicularity are considered.

6.6.2 Experiment 6.9 - Maximizing Alignment of $\vec{g}_f$ and $\vec{m}_{f,t}$

The agent is presented with the same 49 placements of the target object, and based on the exclusion criteria of Experiment 6.8 rejects to attempt grasps in 9 of them. The final node selection criteria that was used in Experiments 6.4 - 6.8 is replaced by a criteria to maximize the alignment of the palm vector (which is assumed to be the same at both the final pose and preshaping pose when they are constructed with the pseudo-inverse of the local Jacobian estimate) and the displacement from the original palm center to the target center. That is, of the candidate nodes with nonempty intersection features, the selected final node will have the maximum value of $C(\vec{g}_f, \vec{m}_{f,t})$. This experiment will answer the question, "Would it be better to only focus on alignment instead of perpendicularity?"

6.6.3 Experiment 6.9 Results

The answer to the question posed is no - the agent's success rate for the grasp action drops if it only considers alignment. The success rate with this method is 9/20 (45%) in L_1 and 7/20 (35%) in L_2, for an overall grasp reliability of 17/40 (42.5%). The results are compared to those of Experiments 6.8 and 6.10 in Table 6.2. Because the agent ignores perpendicularity to the target's major axis, many failures now involve approaching the block nearly vertically, aligned with the major axis, and others approach the block from an odd angle that attempts to close around a corner or other unfavorable cross section, squeezing the block out of the hand in a Palmar bump.

6.6.4 Experiment 6.10 - Maximizing Alignment and Perpendicularity

Since it is generally the case that no single candidate is both the best aligned and the most perpendicular to the target, the agent needs a way to combine these fitness measures and select a single final node n_f. While other combinations could be explored in future work, for this experiment the agent combines the features by a straightforward product. In order to facilitate the combination of one cosine similarity that should be maximized and another that should be minimized, $|C(\vec{g}_f, \vec{o})|$ will be replaced by $1 - |C(\vec{g}_f, \vec{o})|$ so that both measures should be maximized. Then, the agent will select the final node n_f that maximizes the product

$$C(\vec{g}_f, \vec{m}_{f,t}) \cdot (1 - |C(\vec{g}_f, \vec{o})|), \tag{6.4}$$

which will have a value of 1 in the ideal case that the agent has learned to expect to be the most reliable. The remainder of the process of planning the trajectory, which moves along the shortest path set of edges from the home node to the nearest graph node to the preshaping pose, and then moves to the preshaping pose and final pose, remains the same. The results of this experiment will answer the question, “Is there a straightforward way for the agent to consider both alignment and perpendicularity that outperforms planning with either measure alone?”

6.6.5 Experiment 6.10 Results

In this case the answer to the question is yes, the agent can perform grasps more reliably by considering the product in Equation 6.4. The success rate for grasping was observed to improve to 57.5% over the 40 attempts, with performances by location of 11/20 (55%) in L_1 and 12/20 (60%) in L_2. The additional constraints for selection of the best node for grasping appear to affect the reliability of the underlying reach, as the failure cases include 3 misses along with 9 Palmar bumps and 5 bumps. A reach is most reliable when the closest candidate is selected as the final node, and this distance is not considered in the grasping node criteria. The three miss results are examples where the final node selected was far from the target and the local Jacobian estimate of that node was not accurate enough at that distance to move the palm to the object’s center, or even in contact with the object at all. Even with these failure cases, this is the agent’s best grasp performance within the duration of this work, as can be seen through tabulated results of sets of grasp experiments. Table 6.2 provides a comparison with the results of Experiment 6.8 and 6.9, and Table 6.3 provides a comparison with Experiment 6.5, with the best results achieved in the original environment, and Experiment 6.11, a final alternative method in the new environment. A spatial representation of the results of each trial given the initial target mask and number of candidates considered is shown in Figure 6.12. As these results indicate it is the most reliable, this grasping method will be used in Chapter 7 as a component of the pick-and-place action.

6.6.6 Experiment 6.11 - Using only u and v Vector Components

Another source of noise or error in the processed image data given to the agent is the d component of all palm vectors $\vec{g}$ and the vector for the target’s major axis, $\vec{o}$. Recall that the agent is not provided with 3D models of the hand or the target object, and that it does not have the visual

Results	**Grasp: Successful**	**Failed**			
	Palmar Reflex: Activated			**No Activation**	
	Reach: Successful				**Failed**
	Grasp	Weak Grasp	Palmar Bump	Bump	Miss
Experiment 6.8 (L_1)	60%	0%	20%	20%	0%
Experiment 6.8 (L_2)	40%	0%	50%	5%	5%
Experiment 6.8 (All)	50%	0%	35%	12.5%	2.5%
Experiment 6.9 (L_1)	45%	0%	10%	45%	0%
Experiment 6.9 (L_2)	40%	0%	15%	35%	10%
Experiment 6.9 (All)	42.5%	0%	12.5%	40%	5%
Experiment 6.10 (L_1)	55%	0%	30%	15%	0%
Experiment 6.10 (L_2)	60%	0%	15%	10%	15%
Experiment 6.10 (All)	57.5%	0%	22.5%	12.5%	7.5%

Table 6.2: Tabulated results for three experiments that all rely on the new image set, trimmed palm representations, and candidate node exclusion criteria. In Experiment 6.8, the candidate selected as the final node minimizes $|C(\vec{g}_f, \vec{o})|$, and is the most perpendicular to the target's major axis. The grasp success rate decreases in Experiment 6.9 when the agent instead considers only the alignment of the palm vector and the approach direction, choosing the final node that maximizes $C(\vec{g}_f, \vec{m}_{f,t})$. In Experiment 6.10, when the agent considers both perpendicularity and alignment with the final node that maximizes $C(\vec{g}_f, \vec{m}_{f,t}) \cdot (1 - |C(\vec{g}_f, \vec{o})|)$, performance increases. The 57.5% reliability of grasps planned in this way is the most reliable grasp action achieved in this work.

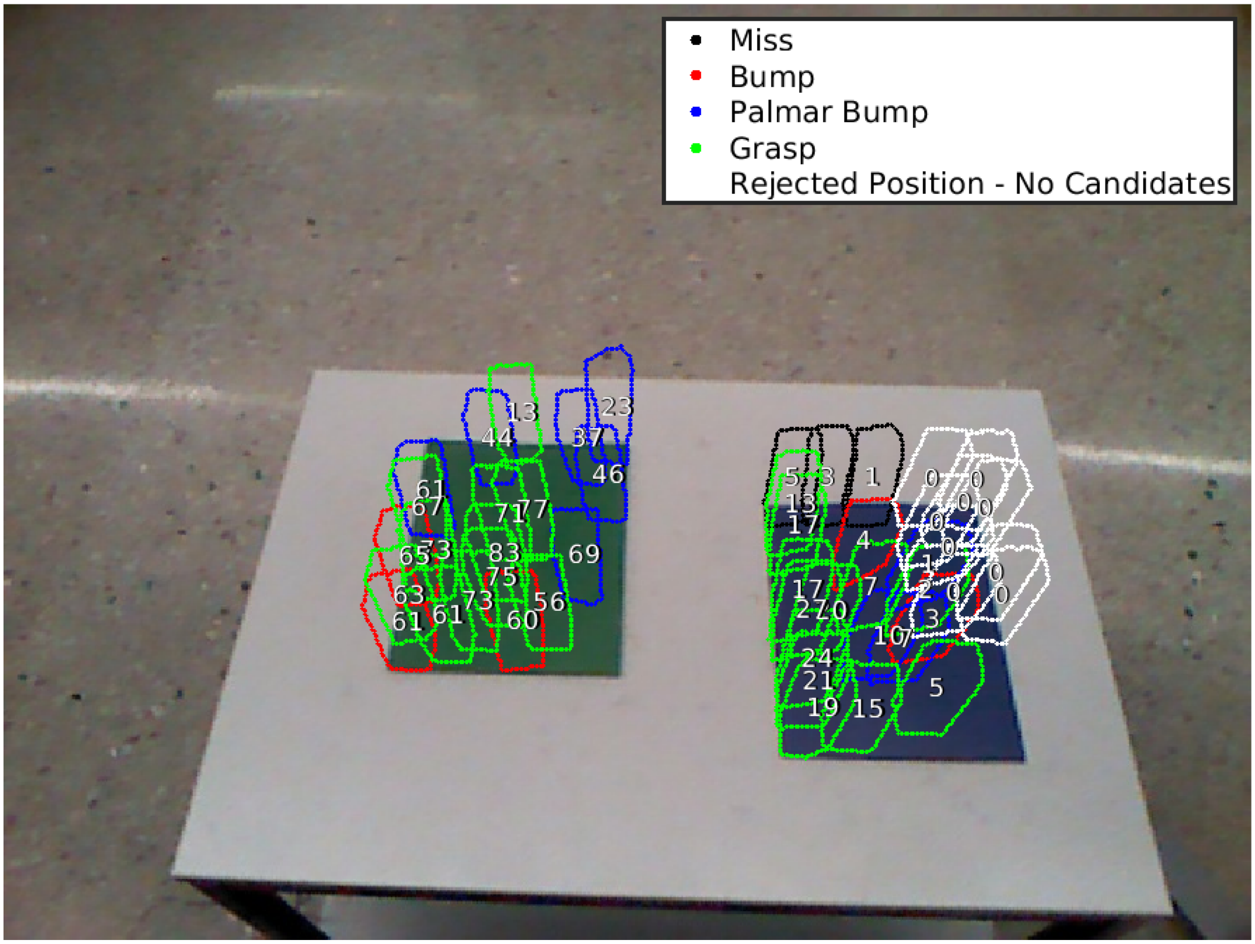


Figure 6.12: A visualization of all 49 target placements for Experiment 6.10, where grasping was 57.5% reliable, with successful grasps in 23 out of 40 attempts, and 9 rejected trials. Each placement is shown as the boundary of the target mask as observed at the start of the trial, colored according to the result of the trial. These results can be compared with Figure 6.5 to see a continued trend that while the positions where successful grasps can be performed are spread around the table, and in both sparsely and densely covered regions, grasping tends to be most reliable in localized lines and areas, and a progression from success to failing to activate the Palmar reflex to failing the reach component as the object's placement is further from these most reliable regions. This appears to have some relationship to the number of final node candidates (shown here near the center of the mask for each trial), which makes intuitive sense since more options will be available to find at least one that has a well-aligned orientation and approach direction and where these vectors are perpendicular to the target's major axis. However, there appear to be other factors, as some of the failure cases in green L_1 have significantly more candidates than the successful cases in blue L_2. The bumps in the lower left corner of L_1 may reflect a difficulty of the agent to line up a successful grasp approach so close to the home node, while the rejections and misses in L_2 may indicate the graph is too sparse to have any reliable candidates when the arm must be so extended to reach an area. The regions where grasping is most reliable may reflect where the hand happened to have clearly identifiable and desirable features when motor babbling near that region, and in other regions the hand may be more difficult to observe accurately or tend to have an undesirable pose. The factors that cause localized success and failures will be identified and addressed in future work.

information or motivation to construct its own 3D models. Instead, the agent considers the mask in (u, v) space and the range of disparity values as separate representations, with the exception of the combination of the mean of all three dimensions to define a center (u, v, d). One of the limitations of this representation is that the agent does not have a standard and accurate way of calculating the d component of the vectors $\vec{g}$ or $\vec{o}$. Instead, the agent will rely on rough estimates of each vector's depth component that are computed by a custom method.

When images of the hand are segmented using the new method where the gripper fingers are wrapped with colored pipe cleaners, binary image masks of subsections of the hand can be created as shown in Figure 6.13, one each for the green gripper finger, the blue gripper finger, the base of the hand (the portion of the image changed from the static background near the grippers), and the palm (the region between the gripper fingers). The u and v components of the vectors are estimated in two ways, and then averaged. The first finds the vector from the (u, v) center of the base to the (u, v) center of the palm, and the second finds the line through the center of the green gripper finger and the center of the blue gripper finger, and computes the vector perpendicular to it. Since this vector could be one of two opposite directions that are both perpendicular to the 2D line, the one that has a nonnegative cosine similarity with the first estimate vector is chosen. (It is this cosine similarity that is tested by the new exclusion criteria introduced in Experiment 6.8, and if it is less than 0.707, it is assumed that the agent cannot rely on the accuracy of this node's representation and it is not allowed to be a final node candidate.) The d component of each palm vector $\vec{g}_i$ is calculated as a single estimate based on the change from the mean disparity value observed within the base mask to the mean disparity value within the palm mask (which itself is based on the disparities observed in the union of the two gripper masks, as the palm mask is the empty space between them and its own disparity readings would be those of the background). While this tends to include a functional estimate of the d component of the palm vector, there are many orientations of the hand where the full trend of depths is not captured well by observing the changes between averages. No additional adjustments are made to the scale of the components other than a final step to ensure the 3D (u, v, d) vector is a unit vector.

The problem of estimating the d component of $\vec{o}$ is more challenging because there are no defined subregions of the target object - this could perhaps be resolved in future work that makes parts of the object visually distinct from each other, but at the cost of providing additional structural information to the agent (either directly, or indirectly through the image processing code that the agent can treat as a black box that provides the representations it is given, which would be more detailed or accurate to the 3D object). Without subregion masks, the method cannot rely on the difference between their average depth values. Instead, the agent finds the major axis of the target mask in (u, v) space, and then creates a "forward" vector from the target center to the boundary along the major axis, and a "backward" vector from the target center in the opposite direction along

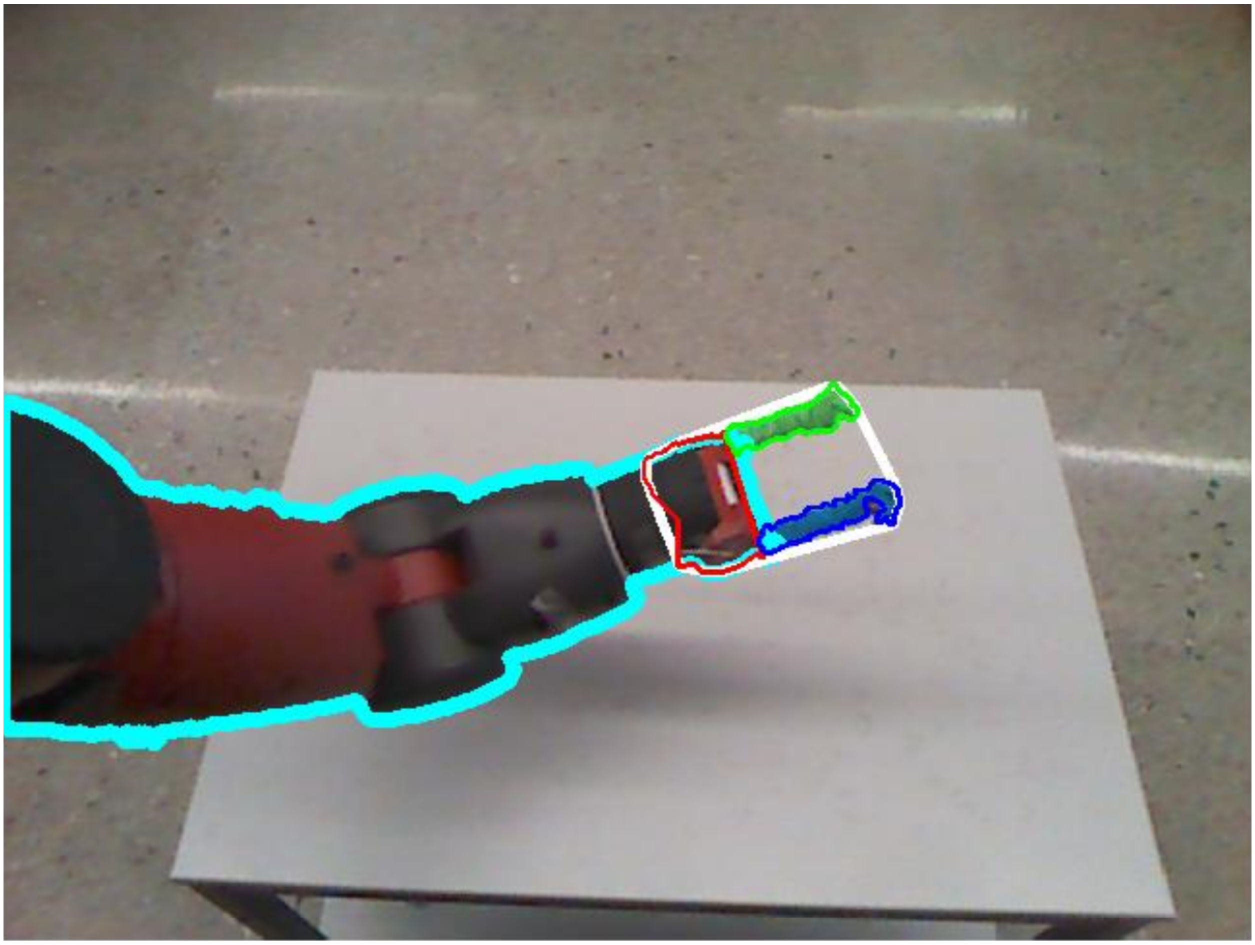


Figure 6.13: A visualization of the masks for subregions of the self that can be segmented using the method that assumes the gripper fingers will have distinctive colors. The masks are shown as boundaries rather than solid masks to allow the original color of the RGB image percept to show. The gripper fingers are identified by their color, and are bordered in green and blue. The mask for the robot's arm is bordered by cyan, and is the largest region of the image that has changed from the static background, minus the gripper masks. The mask for the full arm is used minimally in this work, but may be important for obstacle avoidance tasks in future work. The base of the hand is the intersection of a dilation of the union of the gripper masks and the robot arm mask, and is shown here with a red border. The full hand of the robot is found as a convex hull around the base and grippers, shown here in white. The palm mask occupies the pixels inside the hand mask that are not part of either gripper mask or the base of the hand. In order to make sure the palm corresponds to the grasping region between the gripper fingers, both gripper fingers must be visible and their masks nonempty. If only one or zero gripper fingers is visible, it is possible for the convex hull to indicate a space outside the hand where grasping will not be possible. When the palm mask can be successfully found, the methods for palm vectors use these subregions of the hand, where one estimate finds a vector from the center of the red base through the center of the palm, and another estimate finds a vector perpendicular to the line from the center of the green mask to the center of the blue mask. Starting in Experiment 6.8, the agent excludes nodes without both grippers visible and nodes where the estimates for the palm vector are at least 45 degrees apart.

the major axis until reaching the boundary. The agent creates a “forward” set of disparity values read from pixels that the forward vector passes through, and a “backward” set of disparity values from the pixels that the backward vector passes through. The d component is estimated as the mean of the forward set minus the mean of the backward set. The magnitude of the existing (u, v) vector for the major axis is set to half the length of the major axis, as this is the distance between the halfway points of the forward and backward pixel sets. Then the values are combined into a 3D vector (u, v, d), which is converted to unit length.

In addition to problems with estimating the d component to complete the 3D vector, there are also problems with the scale of this dimension. Distances along the u and v axes can be measured in pixels, which are square for the Kinect RGB-D camera that we use in this work. This means their values are in the same units and the same scale. By contrast, the d component is based on measured disparity values, which do not share the same units. Further, disparity has an inversely proportional relationship to depth or distance in the 3D space, so changes are not on a linear scale. In the current method, these differences are ignored, and the 3D vector is constructed in the straightforward way without conversions. The agent is able to use features with depth components such as the palm centers and target centers despite these differences, simply understanding how to increase or decrease the value of each coordinate according to the relationships modeled by the local Jacobian estimate for each node. However, the difference in scale and inclusion of nonlinearity for d make the calculated palm vectors and major axis vectors imply different orientations than the true poses of the hand or object in 3D space. The vectors used by the agent are also inaccurate when compared to the pose because the disparity values measured within the masks reflect the front face of the object visible to the camera, and not necessarily the values for a line passing through the core of the object. This tends to affect the target object as a single rectangular block more than it affects the hand, which exposes several more faces and a more complete set of the disparities occupied.

Given these limitations on the accuracy of the d component of each vector, we chose to test if the current estimation methods are another source of misleading data given to the agent. If this was the case, the reliability of the agent’s grasp action would decrease. To test this hypothesis, we provided the agent with only the 2D vectors in (u, v) instead of the 3D vectors in (u, v, d), and evaluated performance on the same set of 49 object placements, of which 9 are rejected and 40 are attempted. If the agent performs better, this will be evidence that the current estimates are not sufficient and should be eliminated or replaced. If the agent performs the same or worse, then we can conclude that the estimates are accurate enough to provide some useful information that increases success rates. In that case, the 3D vectors should continue to be used, at least until a more reliable estimate can replace them.

6.6.7 Experiment 6.11 Results

The agent performs slightly worse at grasping tasks when using only 2D vectors in (u, v)-space than the full 3D vectors in (u, v, d)-space. In particular, there is one less successful grasp. With this method, the agent grasps the object from 12/20 (60%) of placements in L_1 and from 10/20 (50%) of placements in L_2, giving an overall reliability of 22/40 (55%). The performance can be considered even more similar to when the 3D vectors were used because one of the failure cases was a weak grasp, where the agent temporarily gained control over the object but was not able to maintain it for the full return trajectory to the home node. The agent also achieved one more Palmar bump and one less miss, suggesting the approach was closer to correct in those cases. Table 6.3 compares the full results to the results of Experiment 6.10, where 3D vectors were used, as well as the final results in the original environment from Experiment 6.5, which this performance remains stronger than.

Results	**Grasp: Successful**	**Failed**			
	Palmar Reflex: Activated			**No Activation**	
	Reach: Successful				**Failed**
	Grasp	Weak Grasp	Palmar Bump	Bump	Miss
Experiment 6.5 (All)	50%	2.5%	22.5%	20%	5%
Experiment 6.10 (L_1)	55%	0%	30%	15%	0%
Experiment 6.10 (L_2)	60%	0%	15%	10%	15%
Experiment 6.10 (All)	57.5%	0%	22.5%	12.5%	7.5%
Experiment 6.11 (L_1)	60%	5%	20%	15%	0%
Experiment 6.11 (L_2)	50%	0%	30%	10%	10%
Experiment 6.11 (All)	55%	2.5%	25%	12.5%	5%

Table 6.3: Tabulated results for Experiment 6.5, which featured the highest grasp success rate in the original environment, and the two Experiments that outperform it in the new environment by considering an additional cosine similarity measure for alignment. Experiment 6.5 is only presented with its full set of 40 trials from random positions across the table and no rejected positions, while Experiments 6.10 and 6.11 have results split by location as well as the overall result. The agent's grasps are slightly more reliable in Experiment 6.10 when they are planned using the 3D vectors in (u, v, d) image space, compared to Experiment 6.11 when they are planned using only the 2D vectors that include the (u, v) components only. We can conclude that the d components provide some useful information, so the method using them will be selected for continued use by the agent for grasping and pick-and-place actions. Future work may find a more informational or more accurate d component for the vectors that will improve performance further.

Despite some improvement in the failure cases, only considering 2D vectors achieved slightly less successful grasps than considering the 3D vectors. This supports keeping the method of Experiment 6.10 as the final method for grasping within this work, as well as the use of that method in Chapter 7 as a component of the pick-and-place action. However, the inclusion of the d

component as the third dimension only produces one additional success with what should intuitively be important and highly relevant additional data. While this supports continuing to use the current information for d instead of simply omitting it, this suggests a next step for improving the grasp action in future work. If either the image processing could produce better depth information or the agent could better use the limited representation used here, this would likely translate to a higher success rate for the grasp action.

6.7 Conclusion

By the end of Chapter 5, the agent had learned a fully reliable reach action. Within this chapter, the agent used the information available from successful grasp trajectories and another iteration of the learning from unusual events pattern to learn a semi-reliable grasp action.

In Section 6.1, the agent identified subsets of the results of reaching trials that produce unusual results, with a qualitative difference from the typical result of a bump causing a change to a new quasi-static position. In one subset, the simulated Palmar reflex is activated and the agent has the distinctive experience of the grippers closing, which it can measure by the change in aperture a. In a subset of these trials with Palmar reflex activations, the agent observes the target object take on a new property where it temporarily becomes part of the self and follows the motion of the hand. Achieving a state where the object has this property is defined to be the result of a successful grasp action.

In Section 6.2, the agent repeats its final set of reach trajectories with a range of settings for the gripper aperture a. Intuitively, one of the requirements for a reliable grasp is that the hand is sufficiently open. Without being informed of this requirement, the agent learns to satisfy it by observing that grasps succeed most often when the hand is fully open ($a = 100$). Using this setting and the final reaching method the agent triggered the Palmar reflex in 30% of trials, and 12.5% of trials were successful grasps.

In Section 6.3, the agent addresses the requirement for reliable grasps that the hand is properly aligned. It learned desirable cosine similarities between the vectors defined in Equation 6.1 and Section 3.4.4. In particular, the agent observed that grasps occur most frequently when the approach and hand opening are aligned (cosine similarity $\approx$ 1), and when both of these are perpendicular to the target's major axis (cosine similarity $\approx$ 0). Planning grasp trajectories with approximately these cosine similarities increased the success rate to 35%.

In Section 6.4, the agent experiments with the value of q^7. With expert knowledge it is clear that this degree of freedom sets the angle of the most distal wrist joint w2, which will be the best choice for orienting the hand without a significant change in the hand's position. The agent can justify this choice without prior knowledge by observing that the change in (u, v, d)-coordinates of the hand in image space for each unit of change to q^7 is much smaller than for the other joints, as calculated in the local Jacobian estimates. A general feature that predicts whether an angle of w2 will be suitable for a grasp has not yet been identified, so the agent simply applies the q^7 value from a previous successful grasp to its new grasp attempts. In combination with the features so far, this technique produces a 50% success rate for grasps.

In Section 6.5, we discuss the move of the robot that our agent controls to a new environment in a new building, and many factors that were changed in the process of this move and over time between experiments. These initially prevent the agent from achieving any successful grasps using the learned method. However, a process that including recording a new RGB and depth image for each node and updating the image processing code that extracts representations from the raw images was able to restore some reliability. In order to return to the success rates that were typical before the relocation of the robot, we introduced a set of criteria that prevented nodes with unreliable data from being considered as final node candidates. When using the set of nodes and otherwise the same decision-making method as the previous section, the agent was again able to achieve a 50% success rate for grasping.

In Section 6.6, the agent is instructed to evaluate additional options for using the vectors in Equation 6.1 and Section 3.4.4 to plan a well-aligned grasp approach. Since Section 6.3, the agent has been focused on finding a final node where the orientation of the hand (measured by the palm vector) is near perpendicular to the target's major axis, and relied on adjustments with the pseudo-inverse of the local Jacobian estimate to produce the aligned vectors that have been observed to be reliable for the final motion of a grasp attempt. However, the adjustment may fail to sufficiently align these vectors, so the agent considered selecting final nodes where the alignment property measured by $C(\vec{g}_f, \vec{m}_{f,t})$ was already maximized and required the smallest adjustment. This was detrimental to the overall success rate until the agent also considered perpendicularity to the target major axis along with the alignment feature, maximizing the product in Equation 6.4. We evaluated this final method using both 2D vectors and 3D vectors to ensure the d component of the vectors was sufficiently accurate to improve performance over a method where it was ignored. When planning with the 2D vectors, the agent was 55% successful, and 57.5% of grasps succeeded when planned with the 3D vectors.

While the current best grasp action policy is only semi-reliable, we leave further improvements in reliability for future work. We instead continue in Chapter 7 of this work by instructing the agent to move on to a learning phase for the next action, the ungrasp. This transition differs from the transition from reaching to grasping since an ungrasp is not a specific grasp trajectory in the way a grasp is a specific reach. However, each successful grasp produces a qualitatively new state due to the grasped property of the object. When presented with this new state, the agent can explore by modifying each degree of freedom, and in this process observes ungrasps as an unusual event.

The ungrasp action can be refined into a place action, giving the agent more control. The current grasp method can be combined with the placing method in the next chapter to form a pick and place action. While its success rate is limited by the current reliability of the component actions, especially the grasp action that remains only semi-reliable, we demonstrate in Section 7.5 that the resulting pick and place action is also semi-reliable.

CHAPTER 7

Learning the Pick-and-Place Action

7.1 Introduction

Thus far, we have discussed the peripersonal space graph representation that the agent builds, and how this facilitates learning of actions to move the arm to reposition the hand, and where more specific trajectories can be planned to reach or grasp a target object. In this chapter, the agent will learn actions that are applicable when an object is currently grasped - as the object cannot be reached or grasped again while it is already in the hand.

Once an object has been grasped, the agent gains control of the object. With fully successful grasps, this control can be maintained indefinitely and the object will not slip out of the hand without action from the agent. In Section 7.2, the agent observes the typical result of the maintained grasp where the object moves along with the hand with highly correlated direction and magnitude of each motion. It will observe an unusual result where the grasped state ends, and will describe this as an *ungrasp*. In Section 7.3, the agent will conduct experiments to learn how to repeat the ungrasp event with a reliable action.

In Section 7.4, we recall the definition of *locations* as qualitative regions of interest in the agent's environment, initially discussed in Chapter 3. The agent also learns to refine the ungrasp action into a *place* action, where the hand is moved to a suitable pose prior to opening the grippers so that the ungrasped object will transition to a quasi-static pose in a specific desired location.

Section 7.5 presents the final experiment of this dissertation, where the agent combines the learned grasp action from Chapter 6 with the place action from Section 7.4 in sequence to produce a *pick-and-place* action. We will demonstrate that the agent can semi-reliably grasp the object from one qualitative location and place it into a different qualitative location, gaining a high level of control over the object and setting up for future learning.

7.2 Observing the Unusual Event of an Ungrasp

7.2.1 Methods

Once the agent has completed a grasp it produces a state of the environment that is relatively unfamiliar, even once grasping is semi-reliable. In the majority of the time and observations in the agent's experience so far, motion of the arm is not correlated with motion of the block object in the foreground. Most commonly, motion of the arm does not cause any motion in the object, and it remains static with the background. In the event of an accidental bump or a bump caused as the result of a successful reach, the block changes position over a short period of time, and then comes to rest in a new position (or disappears, if it has been knocked out of view). Only in the subset of reaches that produce grasps does the block enter a state where it has a qualitatively different behavior of moving along with the robot's hand, with magnitude and direction of its motions highly correlated with those of the hand.

The state reached after a successful grasp is also unique because of the value describing the gripper aperture, a. As the agent motor babbled to create the PPS Graph, a was fixed at 100 (fully open - recall that the value of a specifies a percent of the maximum aperture of the grippers) to increase the apparent size of the hand in the visual percepts and to allow a distinctly colored block to span the space between the grippers, aiding in the segmentation of the hand to create binary image masks. a was also fixed at 100 while the agent learned to reach, so that the pose of the hand best matched those in the stored images. The agent chose to set a to 100 while attempting grasps as it found that approaching a target object with the hand fully open was the most reliable option. In contrast, once the grasp occurs, the Palmar reflex has been triggered and the grippers close, attempting to set a to 0 (fully closed), but often coming to a stop at some fraction open when contact with the grasped object prevents further closure. This new value, and the adjustment of any degree of freedom without an intentional command, are both new phenomenon for the agent.

As the state of the agent and its environment are unfamiliar after a successful grasp, and the property of having the block grasped in the hand violates the preconditions of the learned reach and grasp actions (the target cannot be in the hand and the hand should be empty and open to begin), the agent should perform a broad search over its degrees of freedom in order to determine which may produce new unusual events to study. However, some degrees of freedom can be ruled out under an assumption that they produce the typical result of maintaining the grasp. Recall that in order to verify that the grasp occurred and that the agent successfully gained control over the block, the agent performs a sequence of moves to random PPS Graph nodes and observes the location of the block in the visual percepts as it arrives at each node. When the current binary image mask and depth range for the block at each node intersects with the stored representation of the hand at that node, it is confirmed that the block is moving along with the hand. Since this process involves moving between nodes, and does not prevent the observation that the grasp is maintained, it can be concluded that a divergence from this typical behavior will not be observed by changing degrees of freedom that are essential to the definition of being at a particular node.

As discussed when learning to grasp, the setting of the most distal joint w2 does not make a significant contribution to the image-space center of the robot's hand. This can be quantified by the small magnitude of its coefficient in the local Jacobian estimates, where changes to q^7 are expected to cause negligible changes to c^p. Therefore, it is not essential to set w2 to the stored angle q_i^7 to be considered to be at a node n_i. This has been used previously to justify the rotation of the wrist to better orient the hand for a grasp. The gripper aperture a has no values recorded in the nodes, and is not part of their definition. This also allowed a search over the range of all possible values for a to learn that keeping the gripper fully open ($a = 100$) during a grasp approach gave the best chance of success. Further, changes to a have minimal changes to c^p and the observed shape of the hand. These findings that w2 and a can be freely adjusted without being considered to change the nodes and edges of the trajectory visited are also consistent with the ideas implemented in the Improved Explored PPS Graph, discussed in Section 3.2.2.3 and that will have applications in future work.

The agent's experience with states after the initiation of a grasp demonstrates that adjustments to the major arm joints (s0, s1, e0, e1, w0, and w1), those that remain part of the definition of a node due to their significant impact on the hand's image-space position, produce the typical result of maintaining the grasp. This leaves adjustments to w2 and a to test for unusual results.

In order to more efficiently conduct trials, each begins after the block has been placed in the robot's hand by the experimenter. This avoids having to repeat trials where the grasp attempt fails. The Palmar reflex is activated as normal when the block passes between the gripper fingers, resulting in a similar value of a and a grasp with similar properties as the agent's own grasp actions. Due to these similarities, we assume that the agent will observe the same events with the same motor commands. Trials will continue to start after an experimenter-aided grasp in Sections 7.3 and 7.4 when the agent is aware of ungrasps and learning to repeat them reliably or use them for the place action, but we similarly assume that the learned actions will not rely on this assistance from the experimenter. We confirm that this is the case during the demonstration of the full pick and place sequence in Section 7.5, where the agent performs a grasp independently and then ungrasps the object to place it into the desired location using the techniques learned in these experiments.

7.2.2 Experiment 7.1: Motor Babbling with w2 and a after a Grasp

In this experiment, the agent modifies the value of either q^7, the angle of the smallest wrist joint w2, or a, the gripper aperture, and checks for an unusual result as a side effect to the intended change. The effect of changing the major joints of the arm has been observed, and this is for the active grasp to be maintained, such that the grasped block continues to follow the hand. With this considered the typical result, the agent wishes to determine if changing either of the final two degrees of freedom produce a qualitatively different result.

The agent will conduct 50 trials beginning at a random node n_i (sampled separately for each trial) and with the object grasped, such that initially $q^7 = q_i^7$ and a reflects that the grippers were closed as far as possible until stopped by the grasped object. The agent does not have information to suggest that either w2 or a will be more likely to produce an unusual event of interest, so it will randomly select between them with uniform probability (50% chance each). To avoid the potential of confounding, the agent will only modify exactly one of the two in each trial, and never both. Once a degree of freedom has been selected, the agent will select a random delta to adjust it by in the same manner as was used for the motor babbling to build the Explored PPS Graph. If q^7 is selected, then w2 will be rotated by

$$\Delta q^7 \sim N(0, 0.6117), \text{ as } 0.1 \cdot range(q^7) = 0.6117, \tag{7.1}$$

and if a is selected, the gripper aperture will be adjusted by

$$\Delta a \sim N(0, 10), \text{ as } 0.1 \cdot range(a) = 10. \tag{7.2}$$

For each trial, the agent will record the sampled delta as the intended change as well as the before and after values of the degree of freedom so that it can compute the actual change realized. After the delta has been applied, the agent selects another random node n_j with a minimum path length of at least 3 edges between n_i and n_j. The agent will execute the graph path trajectory from n_i to n_j (changing only the major arm joints to their stored values, leaving q^7 fixed) and use the visual observations taken at each node visited to determine whether the grasp was maintained or the object's grasped state has ended - an unusual event which we will describe as an *ungrasp* event.

7.2.3 Experiment 7.1 Results

The agent randomly chose to attempt to change the gripper aperture a in 20 of the 50 trials. The attempted and actual changes for each trial are shown in Figure 7.1. 19 of these trials exhibited the typical result where the grasp was maintained. Only the single marked trial produced the unusual ungrasp result. The presence of at least one ungrasp justifies further testing of a to produce ungrasps. We can also conclude from the relationship of intended and actual changes that no intended change $\Delta a < 5.6$ produces any actual change. Future trials will only be performed with intended changes above this threshold, as ungrasps will not occur without an actual change.

In the other 30 trials, the agent attempted to change the angle of w2, q^7. The attempted and actual changes for each trial are shown in Figure 7.2. The grasp was maintained in all 30 trials. As no adjustment of w2 produced an unusual event, future experiments for ungrasping and placing will

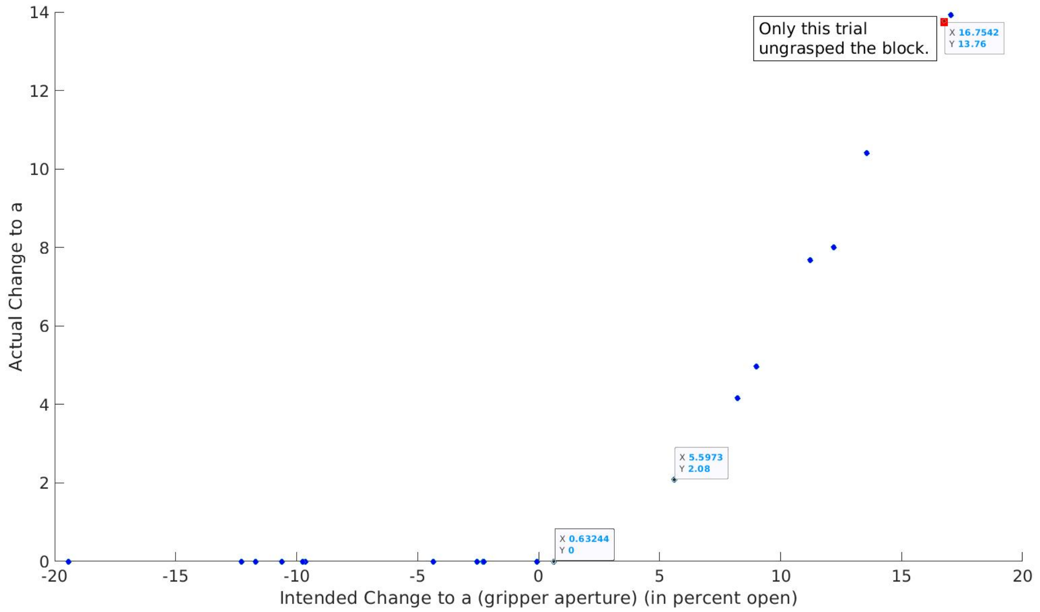


Figure 7.1: The intended and actual changes to the gripper aperture in 20 trials where the degree of freedom set by a was selected for adjustment. Each intended change was sampled from a normal distribution. We see that for negative intended changes there is no actual change, intuitively because the grippers are already closed as far as the grasped block will allow at the beginning of the trial, so the grippers cannot close to a narrower setting. We also observe that for positive intended changes, the actual change tends to be smaller than the intended change. In these cases, the grippers only open until the value of a enters the tolerated range of errors from the intended value, with this tolerance specified by the built-in controller for the grippers. The leftmost annotated trial demonstrates that an intended change of $\Delta a = 0.63$ is too small to produce an actual change as the setting is already inside the tolerated range before any motion occurs. The next annotated trial shows that an intended change of $\Delta a = 5.6$ is sufficiently large to be above the error tolerance, and produces an actual change of 2.08. All trials with intended changes of at least this magnitude had a nonzero actual change, so this value is treated as a threshold for future experiments to ensure the grippers move. The final annotated trial, marked in red, has the second largest intended and actual changes, and was the only trial in this experiment that produced an ungrasp. Using normally distributed deltas for a has a 5% reliability for producing ungrasps.

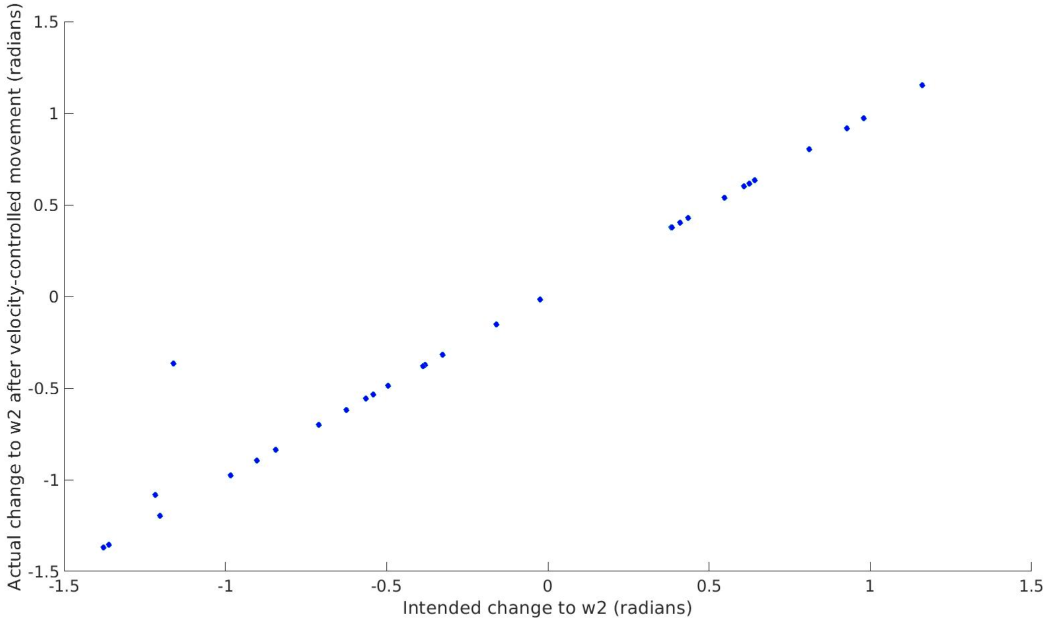


Figure 7.2: The intended and actual changes to the angle of the wrist joint w2 in 30 trials where the degree of freedom set by q^7 was selected for adjustment. All trials produced the typical result where the grasp was maintained after the adjustment, confirmed by moving along a graph path trajectory and observing that the block continued to move along with the hand. While the absence of any unusual events will rule out this degree of freedom for use in further ungrasping experiments, we can take away the additional observation that the robot and the controller used are proficient at achieving the intended change in most cases. It is unclear at this time what caused the smaller magnitude of the actual change for the two trials that did not fit this pattern.

not use w2. We can observe, however, that the intended change to w2 matches up well with the actual change in almost all cases. This differs from a and the major joints, which tend to fall short of completing the full intended change once the remaining error from the set point falls is less than a set tolerance.

7.3 Creating an Action to Ungrasp Reliably

7.3.1 Experiment 7.2: Ungrasping with Larger Adjustments to a

From the results of Experiment 7.1, the agent concludes that modifications to a can produce the unusual ungrasp event. It is not clear what values of Δa will be most reliable. However, the agent can eliminate some unproductive values. The observed relationship between intended changes and actual changes in Figure 7.1 suggests that if the intended change is less than 5.6, then no actual change will occur. As trials with no actual change cannot produce an ungrasp, the agent will use rejection sampling until the intended $\Delta a \geq 5.6$ to be above this threshold. As before, the agent will conduct 50 trials, with each starting with the hand at a random node n_i and after a block has been grasped.

7.3.2 Experiment 7.2 Results

Now that the agent is always adjusting a and only attempting adjustments above the minimum threshold for the controller to produce an actual change, more ungrasps are observed, as should be expected. Out of 50 trials, 6 result in an ungrasp. The 50 new trials are shown together with the previous 20 trials that also attempted to adjust a in Figure 7.3. The ungrasp action policy of this experiment, using a and the minimum threshold, is 12% reliable. We can also observe that five of the new ungrasps occur with some of the largest changes toward more open. Of the trials with intended changes of at least $\Delta a = 17$, 5/7 (71.4%) are ungrasps. While the sample size is smaller, the trials with intended changes $\Delta a > 21.5$ are even more reliable, with 2/2 (100%) succeeding. This informs the next experiment, where the agent will attempt to consistently ungrasp with significantly larger changes to a.

7.3.3 Experiment 7.3: Finding a Minimum Reliable Change to a

The results of Experiment 7.2 show that when the intended change $\Delta a \geq 17$, the ungrasp action is 71.4% reliable, and that when $\Delta a > 21.5$, the ungrasp action is 100% reliable. However, these conclusions are based on only 7 and 2 examples, respectively, casting uncertainty on their accuracy and generality.

Let Δa^* denote a threshold such that all ungrasp attempts so far using $\Delta a > \Delta a^*$ have succeeded. This threshold is valuable because if the agent plans a new ungrasp attempt that also uses $\Delta a > \Delta a^*$, it can be confident that the ungrasp will succeed. After Experiment 7.2, $\Delta a^* = 21.5$. In order to test if this is the correct value for Δa^*, the agent performs 50 additional trials that all use a Δa greater than the current value of Δa^*. If the ungrasp fails (that is, the grasp is maintained) in one of these trials, the value of Δa^* is updated to the Δa used in that trial, so that it is still the case that all ungrasps using $\Delta a > \Delta a^*$ have succeeded. Any remaining trials will use Δa greater than the newly updated threshold.

While it is possible to continue to use rejection sampling from the distribution $N(0, 10)$, the initial rejection threshold of $\Delta a^* = 21.5$ is already more than two standard deviations above the mean and it will be rare for the sampled Δa to be accepted. Assuming there may be new failed ungrasps, Δa^* will increase, and samples above the threshold may become prohibitively rare. Instead of $N(0, 10)$, each Δa will be sampled from a uniform distribution over the half-open interval $(\Delta a^*, 100 - a]$, where a is the current openness of the gripper at the beginning of the trial. Like Experiments 7.1 and 7.2, each trial in this experiment will begin with the grippers closed as far as possible around the object after performing an experimenter-aided grasp, and the value

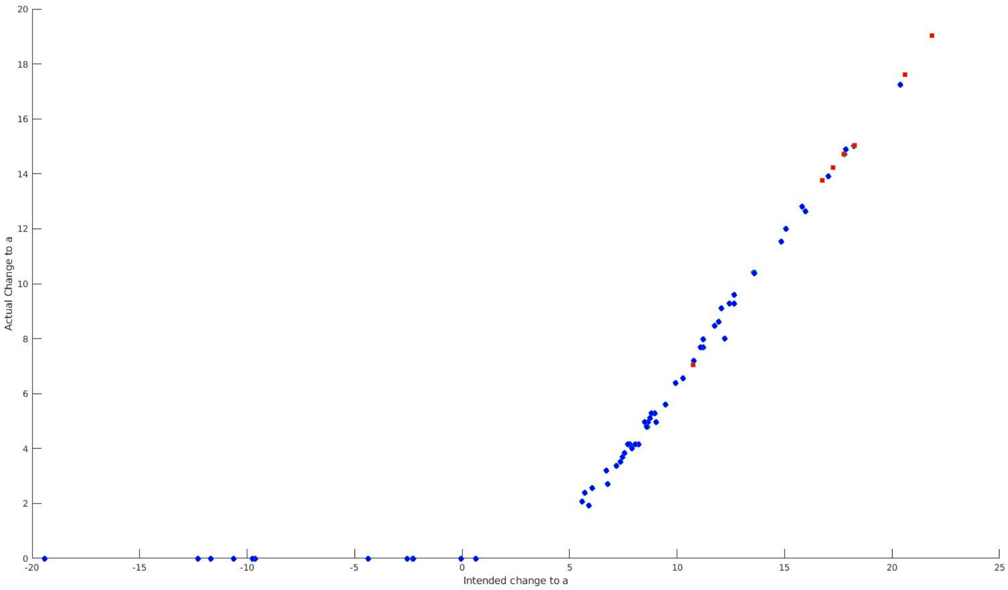


Figure 7.3: The results of Experiment 7.2, where the agent performs 50 more trials where the gripper aperture a is adjusted after determining in Experiment 7.1 that changes to a can produce an ungrasp but that changes to w2 have not. The plot shows the 50 trials of Experiment 7.2 along with the 20 trials of Experiment 7.1 that also adjusted a. We can observe a similar trend as in Experiment 7.1 between the intended changes and the slightly smaller actual changes. Due to the rejection sampling, all new trials from Experiment 7.2 increase the opening by at least 5.6% of the full range. 6 of 50 (12%) of ungrasps attempted with this threshold enforced succeeded, and 7 of 70 (10%) of all ungrasps attempted with adjustments to a have succeeded. Improving the reliability of the ungrasp action further relies on the observation that the successful ungrasps are concentrated in the higher changes to a. Five of the new ungrasps occur with a change of at least 17 percentage points , and only two trials above that threshold had the typical maintained grasp result instead. This provides an estimate that a policy using only changes ($\Delta a \geq 17$) will be 71.4% reliable (5 out of 7). Another estimate that 100% of attempts so far with intended changes of $\Delta a > 21.5$ suggests that a policy's reliability may be improved by using increasingly high minimum thresholds.

of a will reflect this. The introduction of a maximum change is necessary to define the uniform distribution as well as to guarantee the grippers are never commanded to more than 100% open. Another advantage of the uniform distribution at this point is that larger intended changes will be just as likely as those just above the rejection threshold, unlike in the normal distribution where they are less likely. This will allow the agent to more efficiently explore previously unexplored portions of the range of possible Δa values and determine the reliability for ungrasps using those Δa.

7.3.4 Experiment 7.3 Results

Planning ungrasps by changing the gripper aperture by $\Delta a \in (\Delta a^*, 100 - a]$ is a highly reliable method, succeeding in 48 (96%) out of 50 trials. For trials 1-14, $\Delta a^* = 21.5$ was used. The agent observed a failed ungrasp in trial 14, while using an intended change of $\Delta a = 32.74$, which produced an actual change of 29.60. Afterwards Δa^* was updated to 32.74, and used in trials 15-40. Δa^* was updated to the Δa used in trial 40, 41.94, which had produced a 38.72 change in the gripper's openness but failed to ungrasp the block. This value was used for the remaining trials 41-50, and no grasps above this threshold failed. The full set of results from the ungrasp trials in this experiment and experiments 7.1-7.2 is shown in Figure 7.4.

At this point, the agent can conclude that ungrasps will be fully reliable if they are performed by opening the grippers by at least $\Delta a^* = 41.94$. However, the agent has made similar conclusions in the past with lower thresholds Δa^*, and has observed failures when using $\Delta a > \Delta a^*$ until the most recent update to the threshold. Had Δa^* been set to 41.94 initially, the agent's data suggests that it would not have observed any failures.

These two observations - that the agent would not have observed failures with a higher Δa^* and that the agent has been surprised by failures when using Δa that it believed were high enough to be fully reliable - support an extrapolation to conclude that the most reliable threshold choice will be as large as possible. In practice, the agent can only open the grippers to 100% open, so this can be achieved in each future ungrasp attempt by using $\Delta a = 100 - a$. This strategy allows the agent to fully open the gripper in order to perform an ungrasp. Ungrasps planned in this way have been 100% reliable, with the agent always observing the block no longer moves with the hand after this change to the grippers, signaling that the block has transitioned out of the grasped state.

7.4 Refining the Ungrasp Action into a Place Action

7.4.1 Methods

Recall from Section 3.3 that the agent's environment can be described by a reachable space $\mathcal{R}$ and a visible space $\mathcal{V}$, and that the peripersonal space graph is a sparse approximation for the space that is both reachable and visible, $\mathcal{R} \cap \mathcal{V}$. We also define the workspace surface $\mathcal{W} \subset \mathcal{R} \cap \mathcal{V}$,

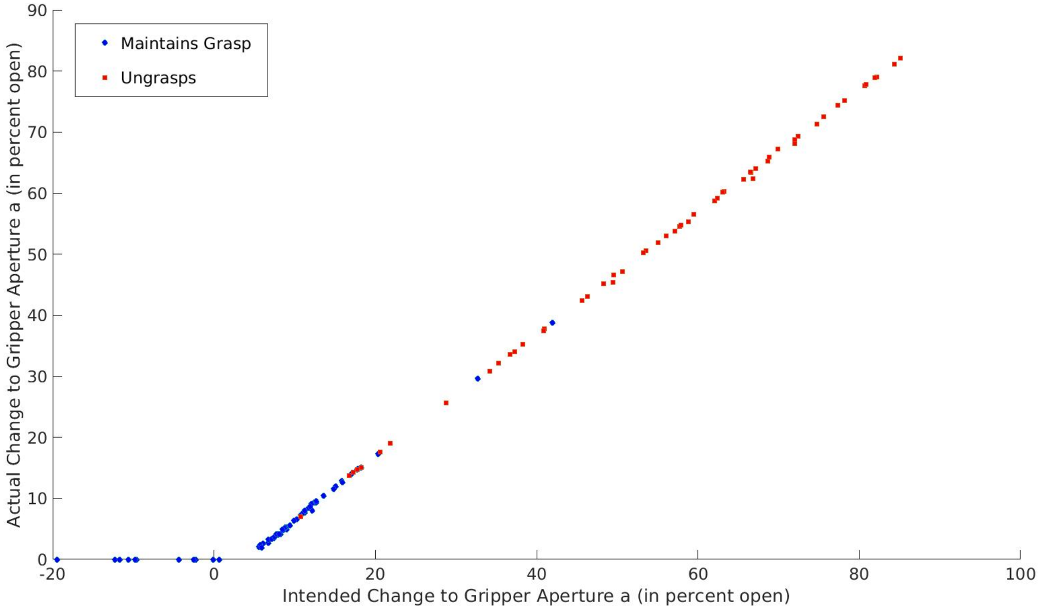


Figure 7.4: The intended and actual changes to the gripper aperture as a percent openness a used in experiments 7.1-7.3. The 50 trials from experiment 7.3 are generally the largest changes, due to sampling the intended changes Δa from $U(\Delta a^*, 100 - a)$ instead of $N(0, 10)$, which provides experience in previously untested ranges of values more efficiently. Sampling in this way also meant that in experiment 7.3, the agent only attempted ungrasps by opening the gripper by at least an intended amount Δa^*, a threshold equal to the largest change that has been observed to fail to produce an ungrasp so far. With this choice, the agent expects all of the attempts to succeed. However, two attempts unexpectedly fail, and the agent revises Δa^* from an initial value of 21.4 to 32.74 and finally 41.94. This pattern demonstrates that the agent may need to increase this threshold further indefinitely as it encounters rare failures in the future. Instead, the agent will extrapolate from the trend that larger Δa tend to be more reliable and conclude that it should always fully open the grippers to perform an ungrasp.

which is the tabletop where the foregound objects that the agent interacts with can have stable positions. Section 3.3 also defines a set of qualitative locations in the agent's environment. Of primary importance to this work are two locations of interest within $\mathcal{W}$, L_1 and L_2, which we assume the agent can visually identify (in this work, this is facilitated with distinctive colors). For these locations, the agent will consider an object to be *in* these locations if its mask has any nonempty intersection with the mask of the location. This lenient definition does not require precise coordinates or a particular orientation of the block, and is therefore a better fit for the agent's primitive visual sense and early motor skills than if the locations were smaller or if a more strict definition for *in* was used. We also define $L_3 \equiv \mathcal{W} - (L_1 \cup L_2)$ that describes the rest of the tabletop, and can also describe locations outside of $\mathcal{R} \cap \mathcal{V}$, such as the invisible location $\neg V$. Certain bumps and ungrasps can cause objects to come to rest out of sight, transitioning to $\neg V$. While this is an unusual outcome, the agent will not focus on this in this work since these objects cannot be observed or interacted with further without intervention by the experimenter. While objects can still be observed and interacted with in L_3, we assume for this work that only L_1 and L_2 are of interest. These locations can be compared to bins that it is desirable to sort certain objects into, where they will have new states or affordances.

For each trial of Experiment 7.1, the agent modified the degree of freedom for either the wrist joint w2 or the gripper aperture a after the hand was at a position corresponding to a random node n_i, and one of these modifications produced an ungrasp. The ungrasp attempts in Experiments 7.2 and 7.3 similarly began at random nodes n_i. By performing the ungrasps from different points in space, the agent was able to observe a variety of post-ungrasp states for the object. Of the 55 successful ungrasps, the post-ungrasp state of the block was observed to be in the blue location L_2 14 times (Figure 7.5), in the green location L_1 19 times (Figure 7.6), in the remainder of the tabletop L_3 11 times (Figure 7.7), and in the out of view location $\neg V$ 11 times (Figure 7.8). For each location, the object came to rest in it in a minority of trials, and an end state outside of any particular location can be expected as a typical result. With the assumption that L_1 and L_2 are regions of interest for the agent or are desirable positions for the block, the agent will focus on the unusual events of the object coming to rest in these regions.

A *place* action, also referred to as a *placement*, will be defined as an ungrasp that is intended to achieve the unusual result of a non-self foreground object being in a specified location. The place action will always include the opening of the grippers to perform the ungrasp component, and may also include an earlier step of moving the arm along a graph path trajectory to a node that is reliable for the intended placement. Experiments 7.4-7.6 will allow the agent to determine how to identify nodes where ungrasping will reliably place the object into the desired location. Specifically, in Experiment 7.4, the agent will evaluate the reliability of the nodes suggested by Experiments 7.1-7.3 and will identify a feature that that can predict the reliability of candidate placement nodes

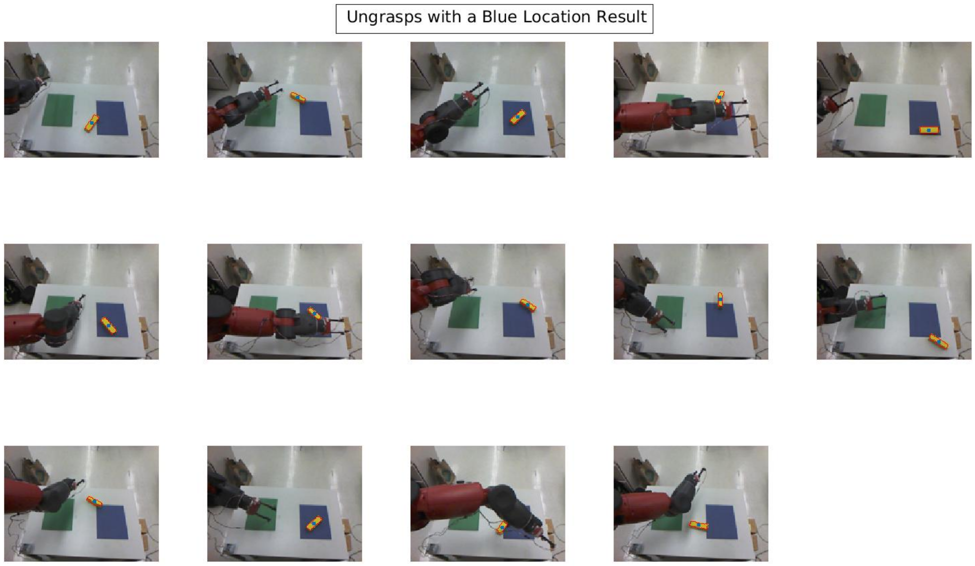


Figure 7.5: Of the 55 successful ungrasps during Experiments 7.1-7.3, these 14 resulted with the object coming to rest into the blue location L_2. The RGB images shown are those recorded immediately after the change to a that caused the ungrasp. The nodes these ungrasps were performed from, where the arm is still positioned in these photos, will be evaluated for their reliability for placing into the blue location in Experiment 7.4. The agent will learn and evaluate a technique for placing into the blue location in Experiment 7.5.

in order to determine a potential policy for planning reliable place actions. In Experiment 7.5, the agent will determine the success rate of the place action when using this policy for the location where the learning took place (L_2), and in Experiment 7.6, the agent will determine the success rate when this learned policy is transferred to be used for another location (L_1). This process will not yield an exhaustive list of the nodes or other positions that can place into the locations with some reliability, rather, its goal will be to find at least one policy for placements that is fully reliable.

7.4.2 Experiment 7.4 - Testing the Reliability of L_2 Placement Nodes

The agent first selects the more rare event of placing into the blue location L_2 to study. From the ungrasps performed from random nodes, the agent has observed a set of 14 nodes where it is possible for an ungrasp performed at those nodes to result in a successful placement into L_2. However, the agent does not have information about how reliable each of these nodes is for the place action - attempts from one of these nodes could succeed consistently, or the observed success could have been a rare fluke. The agent will gather information about the reliability of each node by performing 4 additional ungrasps at each, and counting the number of successful placements observed (out of 5, as the total success rate will include the first placement when the node was used after random selection for ungrasping).

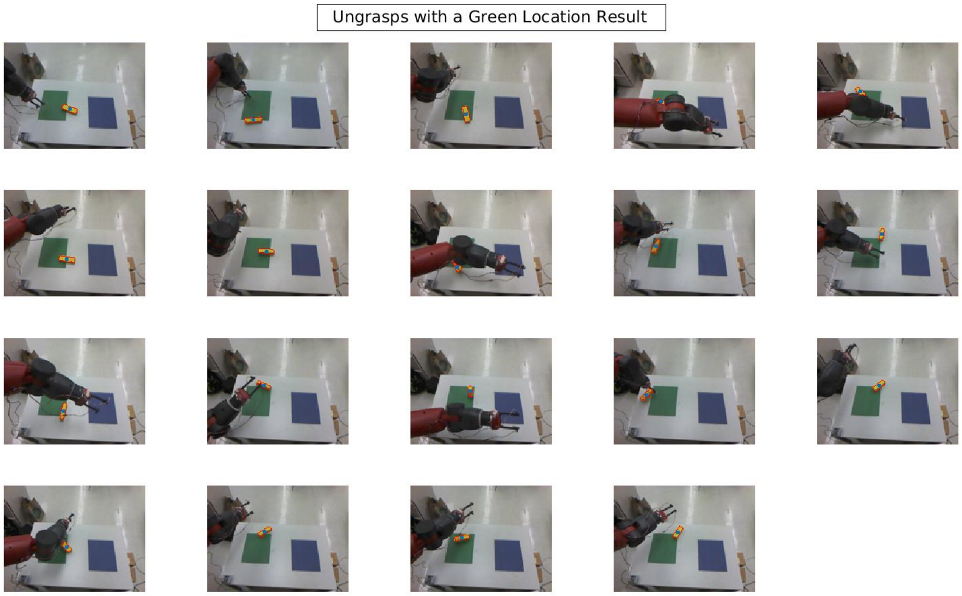


Figure 7.6: Of the 55 successful ungrasps during Experiments 7.1-7.3, these 19 resulted with the object coming to rest into the green location L_1. This was the most common result. The agent will learn to reliably place into the green location as a transfer of the learning it performs to reliably place into the blue location. This will be tested in Experiment 7.6.

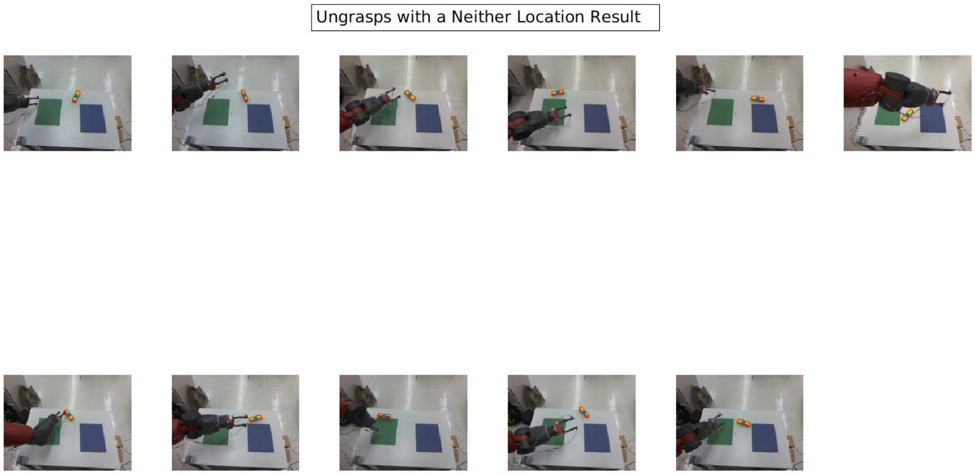


Figure 7.7: Of the 55 successful ungrasps during Experiments 7.1-7.3, these 11 resulted with the object coming to rest in view and on the table's surface, but outside of both the blue and green locations. We expect a technique for purposefully placing in this region (L_3) to be more difficult to learn than a technique for the blue and green locations, due to the more complex shape and the presence of "holes" where the other locations appear. Learning this technique will be left for future work.

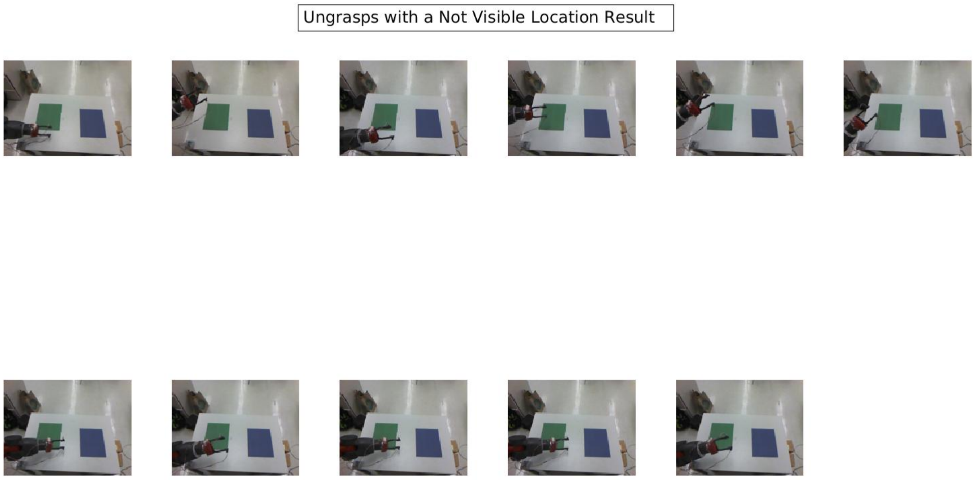


Figure 7.8: Of the 55 successful ungrasps during Experiments 7.1-7.3, these 11 resulted with the object coming to rest out of the agent's field of view. While this disappearance of the object is qualitatively unique, the agent will not study this phenomenon until future work. We have made this decision as placing the object out of view and out of reach cannot be part of a continuing sequence of pick and place actions (it will always be the final time the object can move without experimenter intervention), and as we expect it to be more difficult to make a general rule for which nodes or positions can be used to repeat this event of placement into an out of view location.

After collecting enough data to more accurately estimate the reliability of each node, the agent will examine properties of the nodes in search of a predictive feature for which nodes will be reliable for placing. Perhaps the most straightforward relationship that can be calculated between the nodes, which have stored image data, and the locations, which are only represented visually, is the center to center Euclidean distance in image space, $||c_{L_2} - c_i^p||$. The agent will consider this feature first, and may end its search if a clear trend is found that may be sufficient for prediction of a place action's success.

7.4.3 Experiment 7.4 Results

The agent performed an additional four ungrasps (all of which succeeded using the learned policy of fully opening the gripper) with the hand at each of the 14 nodes where an ungrasp was observed to place into L_2, and the number of successful placements is recorded in the first column of Table 7.1. Figure 7.9 displays the image-space position of the location along with the (u, v, d) coordinates of the palm centers of the 14 nodes, color coded by the amount of successes. The agent observed a full range of reliability within this sample, from nodes with no additional successful placements occurring during Experiment 7.4 to nodes where all ungrasps were successful placements.

Successful placements observed	$\|\|c_{L_2} - c_i^p\|\|$
5 (100%)	10.22
4 (80%)	12.56
5 (100%)	15.17
5 (100%)	15.59
3 (60%)	26.00
2 (40%)	26.63
1 (20%)	30.48
1 (20%)	31.58
3 (60%)	32.93
2 (40%)	35.68
1 (20%)	42.71
2 (40%)	43.07
1 (20%)	50.19
2 (40%)	52.49

Table 7.1: The agent has identified 14 nodes where successful ungrasps also successfully placed the object into the blue location, L_2. The first column shows the number of successful placements observed out of five successful ungrasps. Note that this value is always at least one due to the selection of these 14 nodes as those where one placement had been observed while performing ungrasps from random nodes. Since four additional ungrasps were performed from each node, the highest possible number observed is five. The success rate out of five is shown along with the number of successes observed, as this can be used by the agent as an estimate for the reliability of the node for the place action. This table is sorted by the second column, which is the Euclidean distance in image space between the center of the palm mask for each node, c_i^p, and the center of the mask for the blue location, c_{L_2}. From this ordering, we can observe that nodes that are closer to the location are more reliable for placing into it, though the reliability does not monotonically decrease as distance increases. This trend is examined further in Figure 7.10.

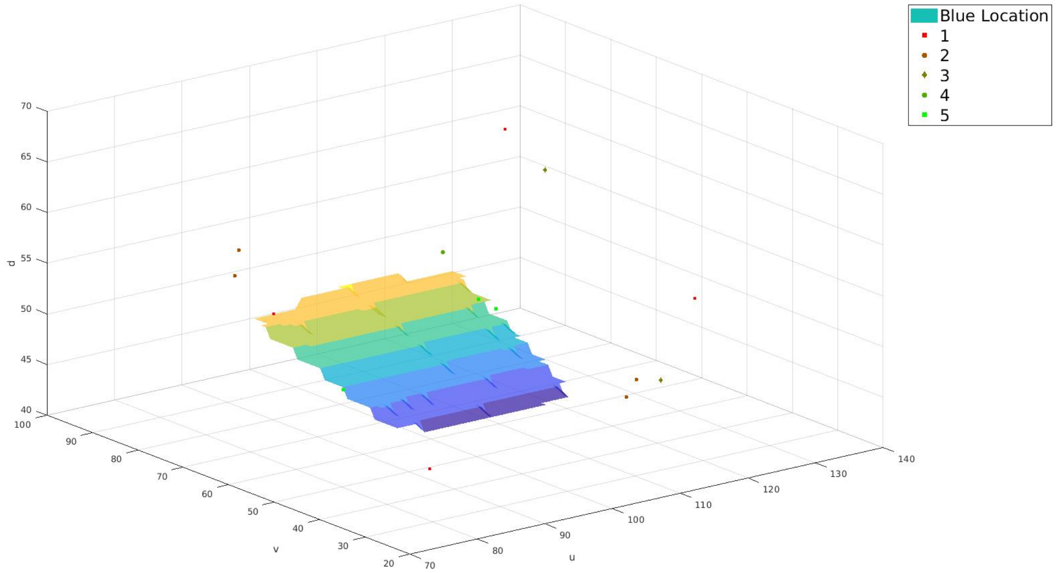


Figure 7.9: The colored surface depicts the approximately planar set of observations (u, v, d) of the blue location L_2 in image space. The 14 blue placement nodes are also plotted by the (u, v, d) coordinates of their palm centers. The color and shape of the plotted points varies with the number of successful placements were observed for that node out of the five attempted. Note how the green points with 4 and 5 successes tend to be near and just above the location (these four nodes are all in the closest 100 nodes to the location's center out of all nodes in the graph), while the red nodes that only had one success are further away (the closest of these is more distant than 493 other graph nodes). The agent continues looking at the relationship between the success rate of the place action using a given node n_i and the distance $||c_{L_2} - c_i^p||$ in the remainder of Experiment 7.4.

As discussed, $||c_{L_2} - c_i^p||$ is a natural value to examine for each node n_i used in this experiment given the visual representations of both the location and each node. The distance for specific nodes relative to the distance for all nodes in the graph gives the first suggestion that this will be a useful feature for distinguishing between reliable and unreliable placing nodes. Out of the 2946 nodes in the Explored Peripersonal Space Graph with defined distances to the blue location's visual center (the other 54 are NaN due to missing or invalid depth readings in the original set of images collected for the Explored PPS Graph), the three nodes that were observed to always place into the blue location (L_2) were the 29th, 85th, and 94th closest to its center. The node with a 4 out of 5 success rate was the 52nd closest. By contrast, the nodes that were only observed to place into the blue location once (during ungrasps from random nodes and never again when intentionally repeating them) were the 494th, 530th, 964th, and 1285th closest. These rankings create a clear division between reliable and unreliable placing nodes. The distance from the center of the location is provided in the second column of Table 7.1, which the rows are sorted by. The decrease in reliability as distance increases that is shown in Table 7.1 is also shown in Figure 7.10 as a conditional probability of success given that the placement is attempted with a node less than a certain distance from the location. There is a clear trend that the conditional probability of success decreases as the distance increases, so the agent will attempt to use a policy of placing the object from the node that is least distant from the center of the location.

7.4.4 Experiment 7.5 - Placing into L_2 from the Nearest Nodes

From the analysis in Experiment 7.4, the agent has learned that nodes with shorter distances between their centers and the center of the desired location L_2 tend to be more reliable for placing. If this is the case, the node $n_i \in \mathcal{N}$ that minimizes $||c_{L_2} - c_i^p||$ will be expected to be the most reliable choice. In this experiment, the agent will test this closest node in five trials and verify that it is fully reliable for placing into L_2. In order to further support the conclusion that the nearest nodes are most reliable, the agent will also test the next four closest nodes (as with the single closest node, these are selected from the entire PPS Graph) for five trials each. While verifying the reliability of only the single closest node is necessary to ensure the agent has a reliable policy, if additional nodes prove to be fully reliable, the agent will have alternative methods for placements in future work that may have additional considerations that rule out using the single closest node, such as obstacle avoidance or if the graph path to that node from the current node is overly long.

7.4.5 Experiment 7.5 Results

The agent attempted 25 placements, with five trials at each of the five closest nodes to L_2. Figures 7.11-7.15 show the before image of the object grasped at one of the five closest nodes along

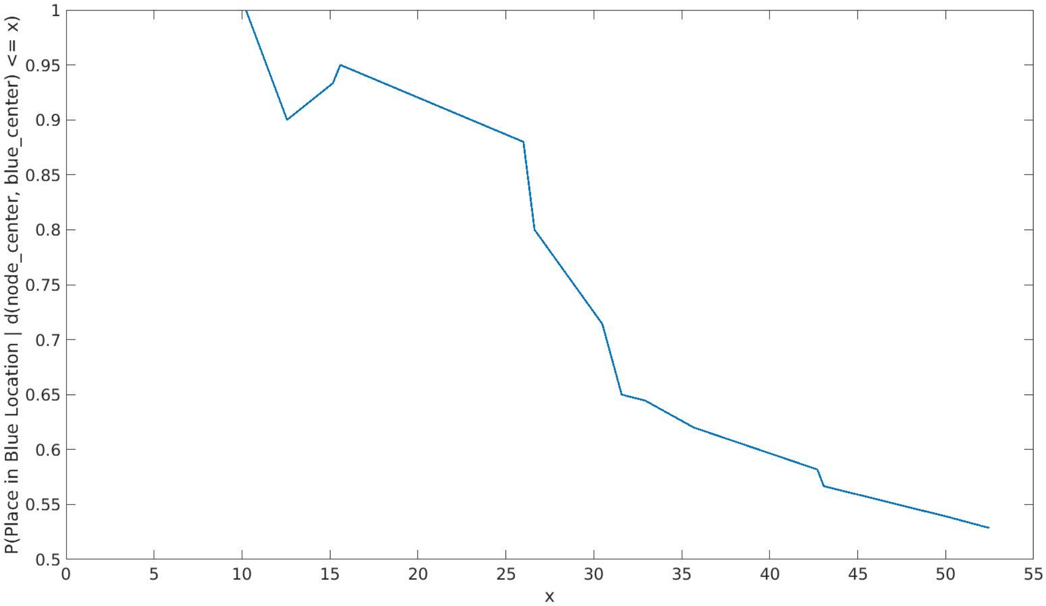


Figure 7.10: Conditional probabilities that an ungrasp will result in a successful placement into the desired location L_2 given that the node where the ungrasp was performed is at most x units from the location in image space. The probability values shown are based on the success rates of the 14 nodes tested in Experiment 7.4, so the values change when the value of x is equal to $||c_{L_2} - c_i^p||$ for any of those nodes n_i. At those points, the total number of successes in the calculation of the probability increases by the number of successes observed at that node, and the total number of trials increases by five. Due to a single failure to place into L_2 when using the second closest node, the estimate of conditional probabilities that the agent calculates is not monotonically decreasing as x increases. The failure at this node despite its closeness can be attributed to factors that are not modeled by the agent, such as the orientation of the object in the hand when a grasp is maintained at that node, or whether the displacement from the location's center in image space corresponds to a displacement in workspace that would have the object near the table's surface and squarely "above" the location for an ungrasp. Despite this anomaly, the agent can conclude that it should use nodes with minimal distances $||c_{L_2} - c_i^p||$ so that they fit into the condition of a lower threshold x, and so that successful placements are expected to be more likely.

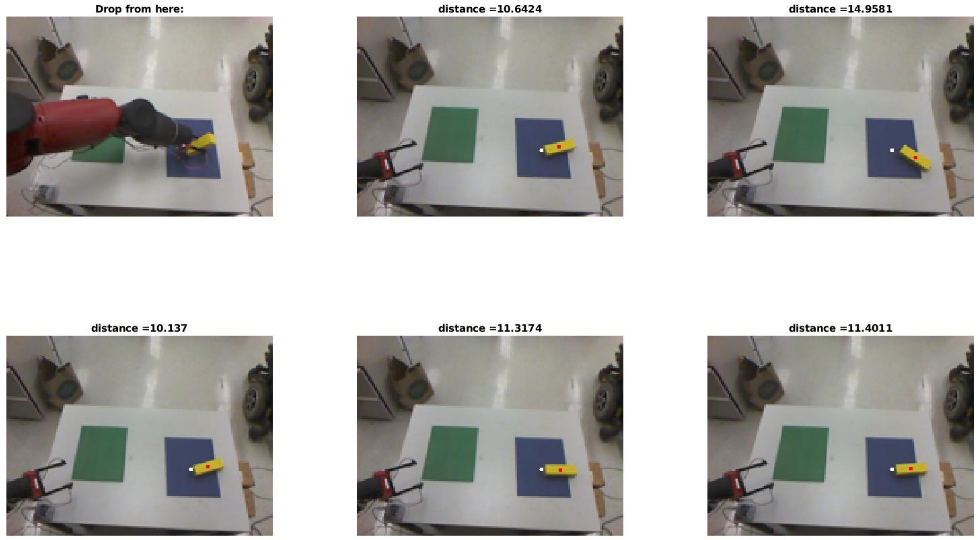


Figure 7.11: Images when placing into the blue location L_2 from the closest node. The top-left panel shows the observation just before the grippers are opened, and the remaining panels show images from after a placing trial when the agent has returned to the home node. As expected due to its minimum distance from the location's center, this node is fully reliable for placing. Each after image is annotated with the distance between the final target mask center and the center of the location, which is ideally low and consistent between trials.

with the after image for each place action trial performed at that node. Using these after images, the agent observes that the object is *in* the blue location in all 25 trials, and this evaluation matches the ground truth that every placement was successful. The agent has successfully identified a policy for the place action that is 100% reliable, and this consists of moving along graph edges to the node that minimizes $||c_{L_2} - c_i^p||$ and then fully opening the grippers to ungrasp. In the remainder of this work, the agent will always attempt to place into L_2 using the single closest node, but in future work it could use alternative nearby nodes - for example to use a node more likely to place the block in an upright position, like the second closest node in Figure 7.12 - and still expect a 100% success rate.

7.4.6 Experiment 7.6 - Transferring the Nearest Nodes Policy to L_1

Having learned a fully reliable policy for placing into L_2, the agent changes its attention to finding a policy for placing into L_1, the green location that the object was *in* at the end of 19 of the ungrasp trials. While the learned policy was based on observations of the nodes that placed into L_2 during ungrasping and was verified on the closest nodes to L_2, an analogous distance feature can be defined using the green location's image-space center instead: $||c_{L_1} - c_i^p||$. If using the node that minimizes this distance is fully reliable, the policy transfers between locations successfully, and it will not be necessary to evaluate the reliability of the original 19 nodes further, and the agent

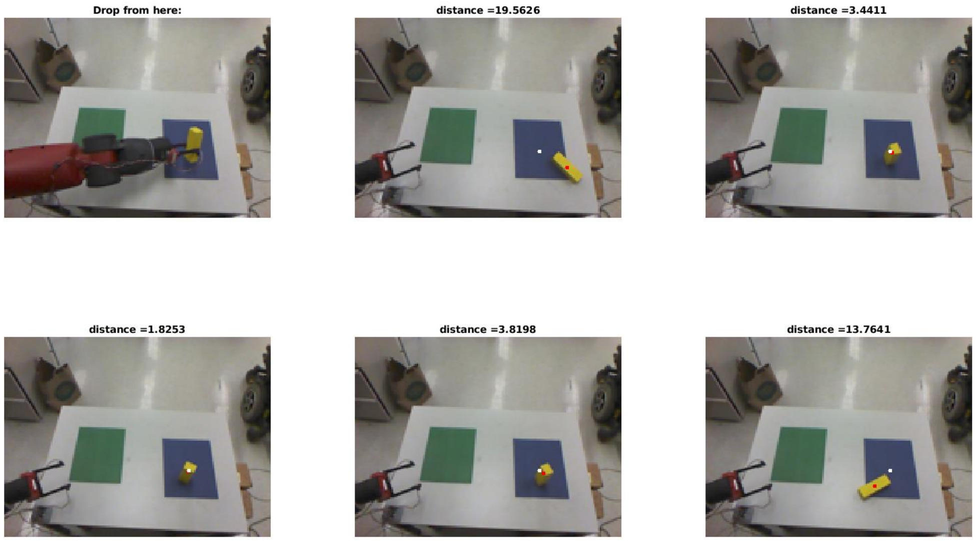


Figure 7.12: Before and after images for place action trials, similar to Figure 7.11 except using the second closest node. This node is also fully reliable. This node may be desirable for use due to its tendency to place the block in an upright position, as it did in three out of five trials. However, when the block does not land in a stable upright position, it seems to tip in any direction, causing the final position to be less consistent than when other nearby placing nodes. This may make the node undesirable as well. The agent will need to determine the fit of this and other reliable placing nodes for more specific future actions.

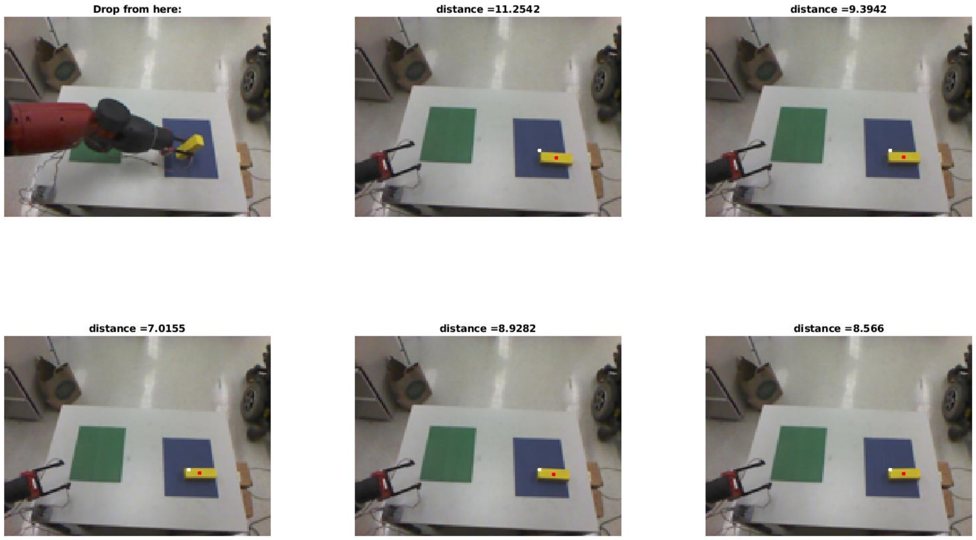


Figure 7.13: Before and after images for place action trials, similar to Figure 7.11 except using the third closest node. This node is also fully reliable.

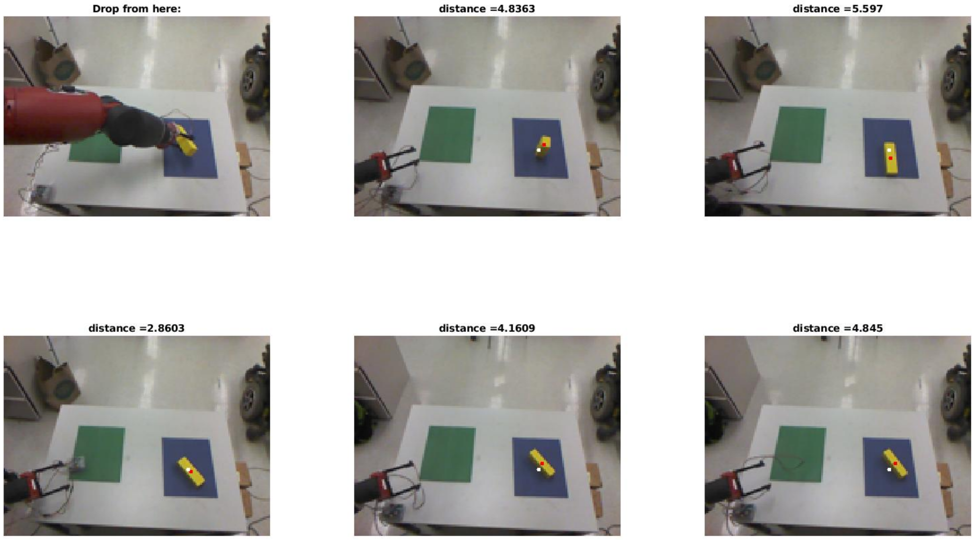


Figure 7.14: Before and after images for place action trials, similar to Figure 7.11 except using the fourth closest node. This node is also fully reliable.

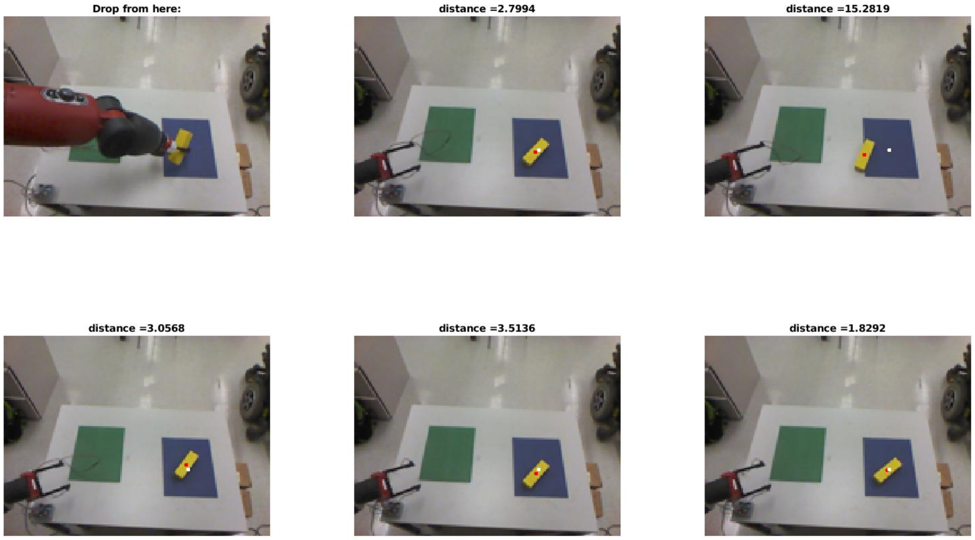


Figure 7.15: Before and after images for place action trials, similar to Figure 7.11 except using the fifth closest node. This node is also fully reliable. Despite the node's center being further away from the location's center, the orientation the hand holds the block in at this node allows it to be dropped a short distance into a nearly consistent final position which is very close to the location's center. Since the agent has found that each node tested was fully reliable, in future work it may wish to evaluate other characteristics of the resulting positions to determine which should be used.

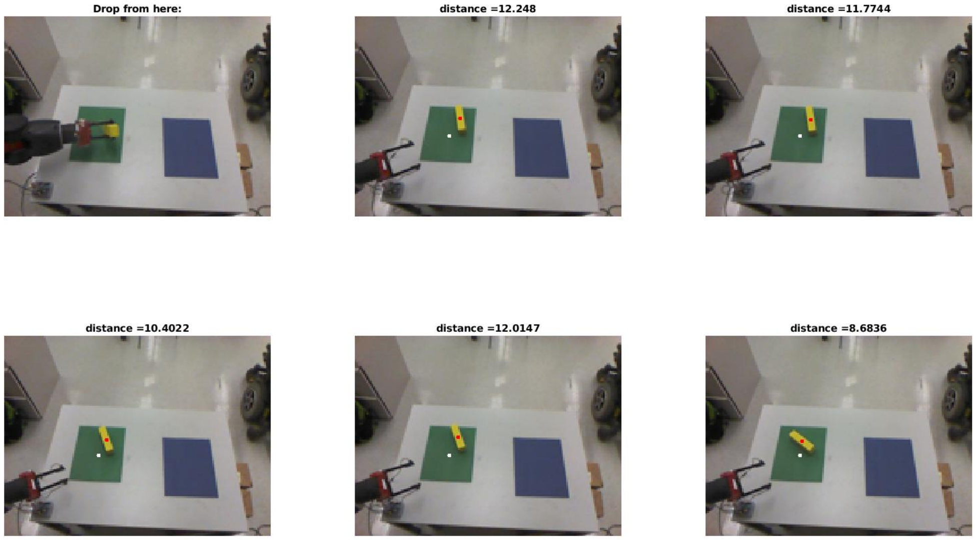


Figure 7.16: Images when placing into the green location L_1 from the closest node, which is expected to be reliable based on the minimum distance policy transferred from placing experiments with L_2. As before, the top-left panel shows the observation just before the grippers are opened, and the remaining panels show images from after a placing trial when the agent has returned to the home node. This node is fully reliable for placing, but despite the node's closeness to the location the final positions of the block do show some variation.

will not have to repeat the examination of the distance feature or other potential features. In order to test the transferred policy, the agent will again attempt 25 placements, this time into the green location L_1, and with five each being performed from the five nearest nodes to L_1's center in image space. That is, the nodes from the full set of graph nodes $\mathcal{N}$ that have the five shortest distances $||c_{L_1} - c_i^p||$ will be identified, and five placements into L_1 will be attempted from each.

7.4.7 Experiment 7.6 Results

Of the 25 attempted placements into L_1, the agent's evaluation of after images taken while at the home node determines that 24 succeeded (which agrees with the ground truth from the experimenter's assessment). The before and after images for the trials at each node are presented in Figures 7.16-7.20. Using this set of the five closest nodes had a 96% success rate. The failed placement in one of the trials using the fifth closest node (Figure 7.20), and the larger distance from the location's center in each trial with that node, suggests that the agent may have fewer policies that use alternative placing nodes that it can expect to be 100% reliable. However, it would be feasible for the agent to evaluate additional nodes, starting with the sixth closest, to find another node that is 100% reliable if all closer nodes cannot or should not be used according to a consideration that could be added in future work.

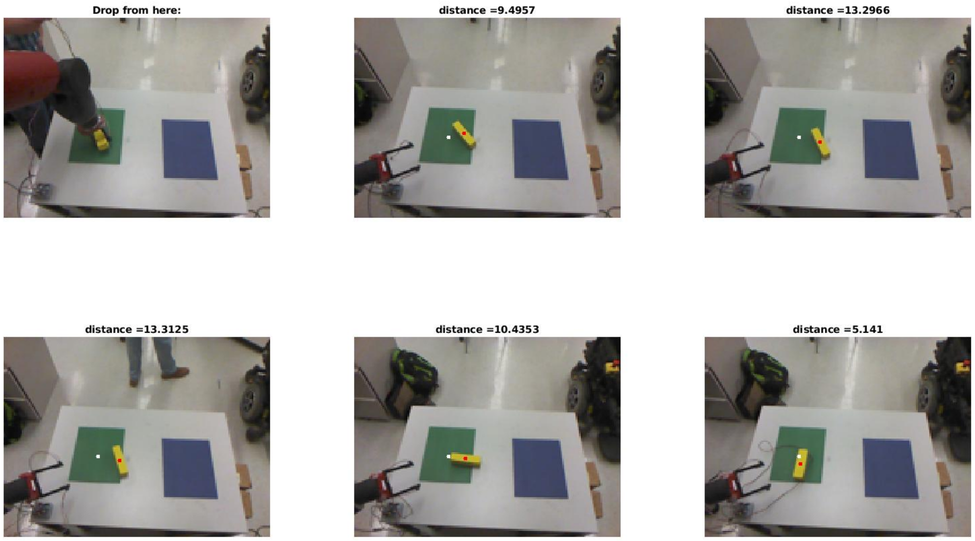


Figure 7.17: Before and after images for place action trials, similar to Figure 7.16 except using the second closest node. This node is also fully reliable.

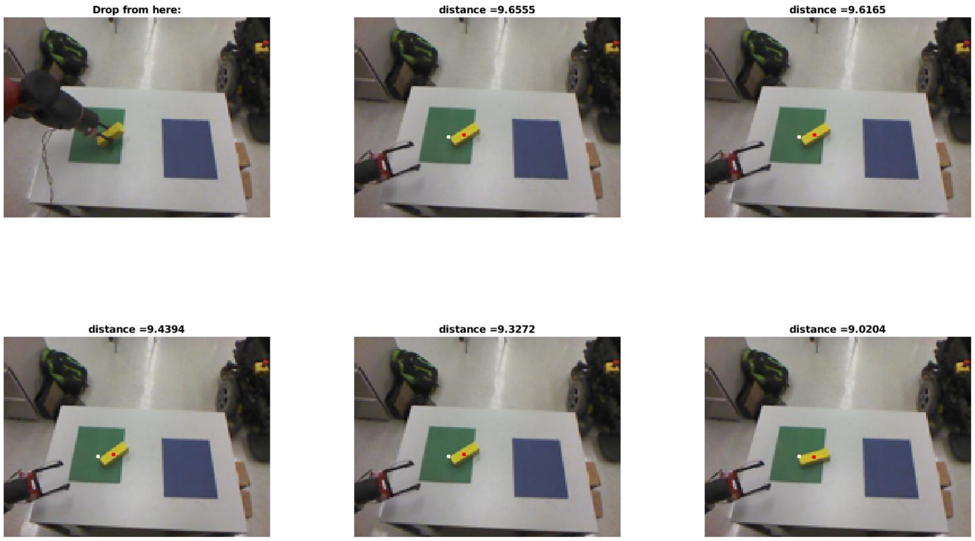


Figure 7.18: Before and after images for place action trials, similar to Figure 7.16 except using the third closest node. This node is also fully reliable, and has resulting position of the block has a more consistent center and orientation than in either of the closer nodes, which may make it desirable for use.

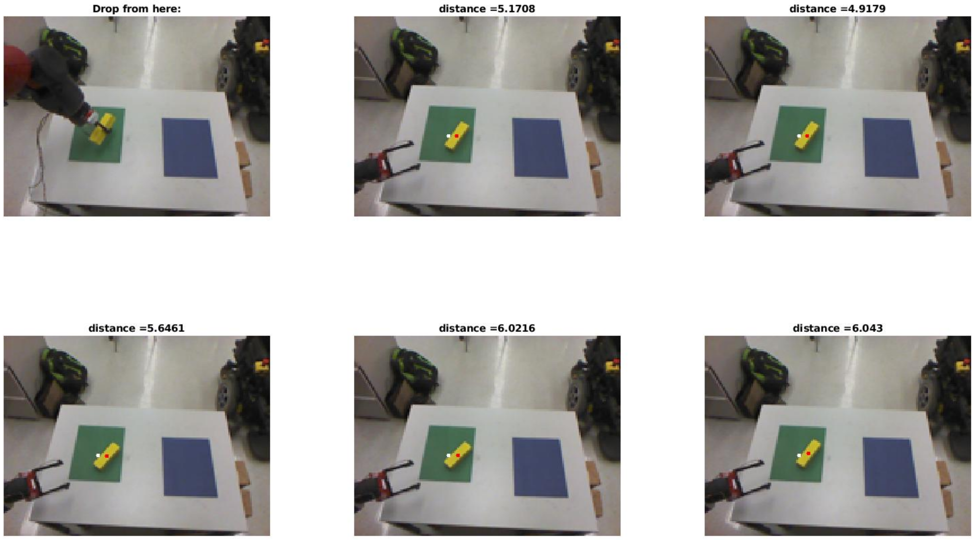


Figure 7.19: Before and after images for place action trials, similar to Figure 7.16 except using the fourth closest node. This node is also fully reliable.

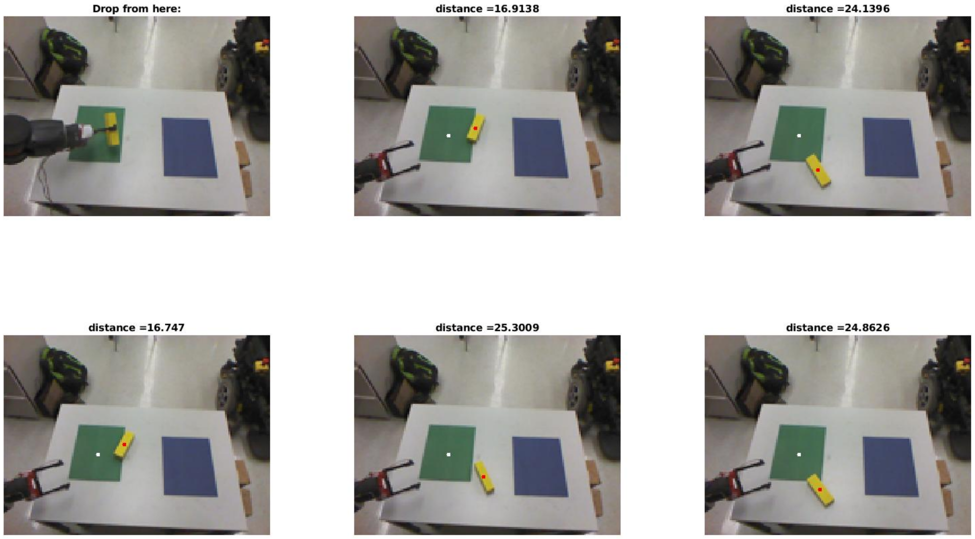


Figure 7.20: Before and after images for place action trials, similar to Figure 7.16 except using the fifth closest node. When placed from this node, the object ends up much further from the location's center, and in one case has no intersection with the location's mask at all. This does not meet the agent's definition for being in the location, so this node is only 80% reliable for placing into L_1. In order to retain a fully reliable place action, the agent can place from a closer node, or may continue to experiment to potentially find nodes that are further but still place into L_1 100% of the time. In this work, the agent is only concerned with being in the location, so it adopts the most straightforward strategy of only using the single closest node.

Treating the five trials as tests of a policy of only using the single closest node, and the agent has found this policy for the place action to be 100% reliable when the target location is L_1. This conclusion is based on only five trials, but it does not appear likely that additional trials would have different results, as the IOU between the object and the location is approximately 1 in each of the five trials, and the object's center in the image taken after placing is relatively consistent.

The transfer learning process was successful, and the method of minimizing the distance to the location before performing the ungrasp component of the place action appears to be equally applicable to the green location as it was to the blue location. Since the feature is not specific to these locations, but only parameterized with the center of a mask and depth range that could be found for any location, the agent can safely assume that this policy will work in general, including if either of these locations of interest is moved or if the agent is presented with additional locations.

7.5 Sequencing Component Actions to Pick-and-Place

7.5.1 Methods

As the agent learned to move the arm in Chapter 4, we assumed that one of its innate capabilities was to combine independent actions in parallel. This was important so that the agent could use the learned actions for moving single joints to set point angles to create an action that moved all the joints of the arm to set points simultaneously. For this experiment, we must assume that the agent is also capable of combining actions in sequence to construct higher order actions. When constructing a new action as a sequence of component actions, the order of the sequence is constrained by a need to ensure that the prerequisites of each action will be met when the previous action is completed successfully. Each trial in this section will begin with a single block placed in either L_1 or L_2 by the experimenter, and the agent's hand will be empty. As a consequence of this initial state, the agent may choose between the move arm, reach, and grasp actions for the first component action. If the agent chooses to move the arm or reach and bump the block, it is still the case that the block is in a quasi-static position on the table and the agent's hand is empty, so the same set of actions may be considered. Once the agent chooses to attempt a grasp and the grasp succeeds, the set of valid actions for the agent to attempt will change. In particular, with the block in the hand instead of on the table, the prerequisites for the reach and grasp actions are not met, and they cannot be attempted. Instead, the agent may move the arm, or it may attempt an ungrasp or place action. The sequences that may be carried out follow this pattern, and this pattern can be repeated for an indefinite number of cycles:

1. 0 or more actions from [move arm, reach]

2. grasp action

3. 0 or more move arm actions

4. 1 action from [ungrasp, place]

In this work, we have assumed that L_1 and L_2 are regions of interest, and that the agent will always either have an intrinsic motivation to place into these locations for continued learning opportunities or an extrinsic motivation to place into these locations, similar to an older child using a shape sorting toy or being rewarded for putting toys into their correct storage container. This is done much more reliably with the place action that uses a well-chosen node where the grippers should be opened than with the unrefined ungrasp action that opens the grippers at any node. Therefore, during the demonstration of the sequence in this work, the agent will always choose to perform the place action during the final step of the pattern. In order to collect samples of the sequence more efficiently, for this work we also restrict the agent's ability to perform option actions that do not change the set of actions that may be performed next. That is, the agent will not be allowed to choose to perform any actions in step 1 or 3 of the pattern. This leaves a streamlined set of steps that we have used to collect data on the reliability of the pick-and-place sequence:

1. The object is placed in a random position within L_1 (respectively, L_2) by the experimenter.

2. The agent grasps the object.

3. The agent places the object into L_2 (respectively, L_1) - the agent can verify the grasp was successful using observations of the object along the trajectory to the most reliable placing node.

4. The agent returns to the home node - the agent can verify the placement was successful using the after image taken from the home node.

In future work, the agent will be allowed to continue to practice and improve its action policies in autonomously chosen sequences that are only constrained by the prerequisites of the actions rather than the most efficient pick-and-place sequence of a grasp followed by a placement. The implications of the extension to autonomously chosen sequences is discussed in the conclusion of this chapter in Section 7.6. At the time of this work, the grasp action is prohibitively unreliable for objects that violate the assumed shape and orientation of an upright block with one long dimension. This is due to the Explored PPS Graph being especially sparse this close to the table's surface (the agent was prevented from motor babbling to positions low enough to potentially contact with the table) and because of the rarity of nodes that point downward at the table to be suitably perpendicular to the perceived major axis of object masks for blocks that are laying down. Given that a majority of placements leave the block laying down in the desired location at the time of this work, the first step of experimenter intervention to position the block before each pick-and-place trial is necessary. We will also discuss how this step could be eliminated in Section 7.6.

When performing the grasp action component of the pick-and-place sequence, the agent will use the criteria learned in Chapter 6 for choosing the most reliable grasping node for the specific target position. When performing the place action component of the pick-and-place sequence, the agent will use the fully reliable policy it has found where it always uses the single closest node to the desired location. This means the agent will always plan to place into L_1 with the closest node to its center, n_{79} (Figure 7.21), and similarly will always plan to place into L_2 with the closest node to its center, n_{1251} (Figure 7.22). Note that all work with the pick-and-place sequence was performed using the second set of images for the Explored PPS Graph with the robot in the new environment, and these nodes are the closest according to that image data. Because the agent uses observations from the next steps of the sequence to check if the previous step was successful, it will always attempt to carry out the full sequence even if an early component action fails.

7.5.2 Experiment 7.7 - Performing the Pick-and-Place Action Sequence

In order to test the reliability of the full sequence of learned actions together to perform a pick-and-place action, the agent will perform 40 trials and evaluate the number of successes. Each sequence will consist of a grasp, a placement, and a return to the home node to observe the result. The agent is given the option to reject any trial where the initial placement of the block - performed by the experimenter according to randomly generated coordinates in one of the locations of interest - is determined to have no candidate nodes to perform the grasp portion with. In order for there to be no candidate nodes, there must be no node with valid image and depth information that has intersections between both its stored mask and the perceived target mask and the stored depth range and perceived target depth range. When a trial is rejected, the placement of the block is recorded, but no movement is attempted. The experimenter will use a new set of random coordinates to replace the block, and this new placement will be accepted for an attempt or rejected again if there are still no candidate nodes. In order to test the performance evenly, the agent will make 20 attempts to grasp the block from the green location L_1, where the Explored PPS Graph is dense, and the agent will also make 20 attempts to grasp the block from the blue location L_2, where the graph is much more sparse and requires an extension of the arm across the robot's body to reach. When grasping from L_1, the agent will place into L_2, and when grasping from L_2, the agent will place into L_1.

7.5.3 Experiment 7.7 Results

The agent rejected 8 of the initial placements of the target block, so it was necessary to generate 48 positions for the agent to attempt 40 pick-and-place actions. Of these 40, 21 were successful

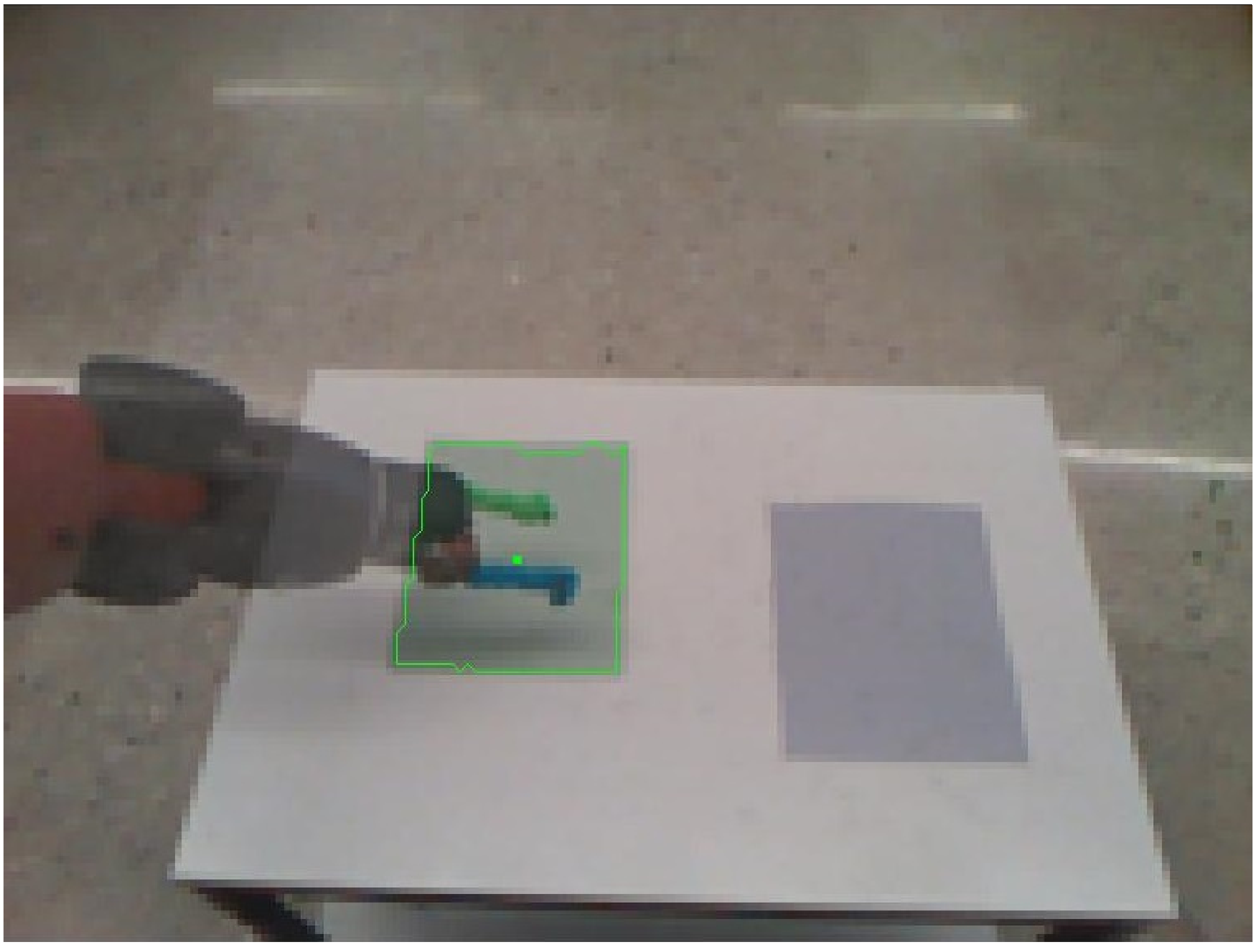


Figure 7.21: The stored RGB image for n_{79} superimposed with an observation of tabletop locations of interest L_1 and L_2 that was taken without the arm present. The boundary and center of the mask for the green location L_1 are highlighted in green. n_{79} will be used for all placements into L_1 because its palm mask center is the closest in image space to the center of L_1, that is, it minimizes $||c_{L_1} - c_i^p||$. The agent has learned that it can expect placements from the closest node to be 100% reliable.

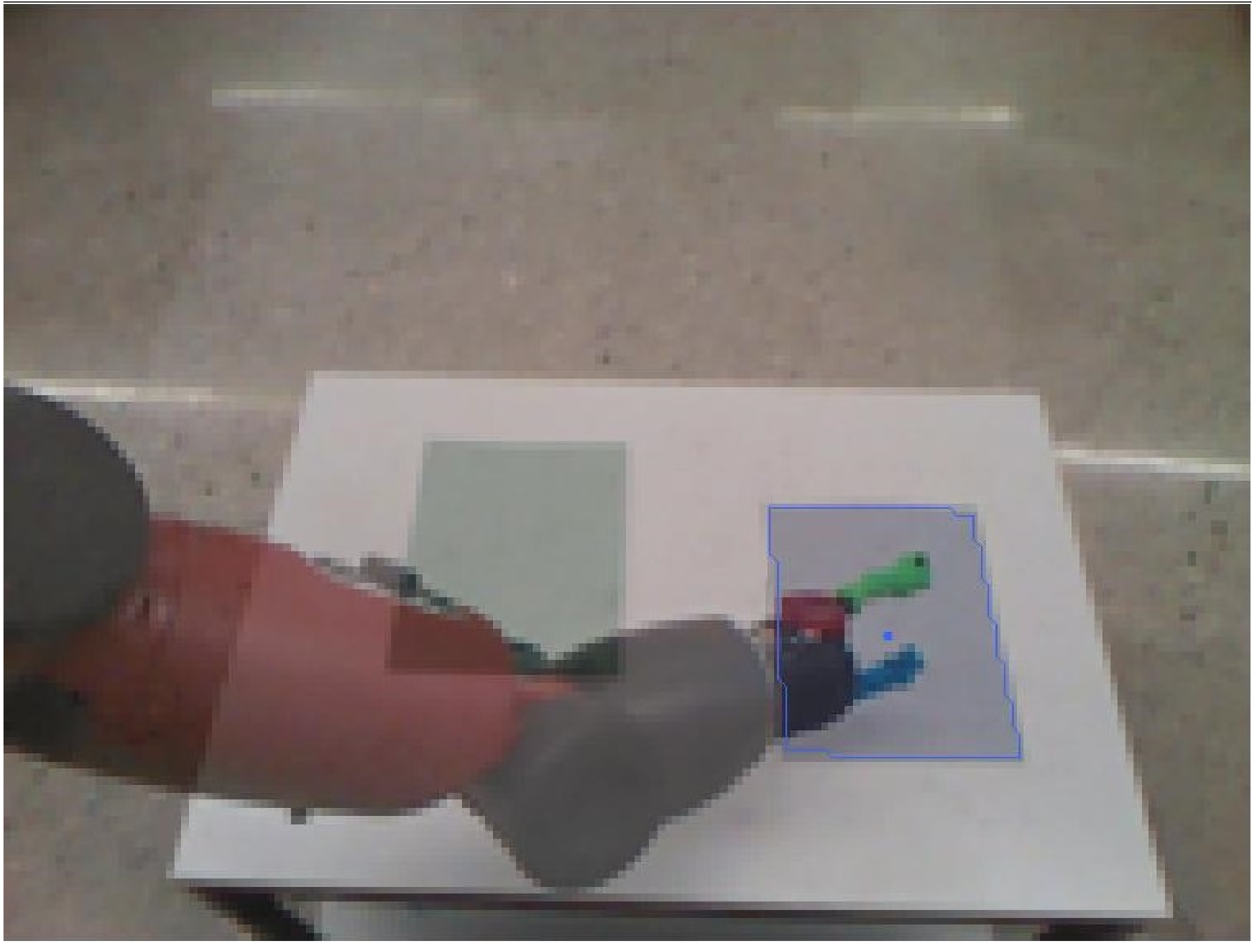


Figure 7.22: The stored RGB image for n_{1251} superimposed with an observation of tabletop locations of interest L_1 and L_2 that was taken without the arm present. The boundary and center of the mask for the blue location L_2 are highlighted in blue. n_{1251} will be used for all placements into L_2 because its palm mask center is the closest in image space to the center of L_2, that is, it minimizes $||c_{L_2} - c_i^p||$. The agent has learned that it can expect placements from the closest node to be 100% reliable.

pick-and-place actions, with a successful grasp followed by a successful placement into the desired location. 10 of the successes were trials where the agent grasped from the green region L_1 and placed into the blue region L_2, and the other 11 were in the opposite direction, moving the block from L_2 to L_1. As seen in the flowchart in Figure 7.23, the 19 failed trials consist of 3 misses (failed reaches), 6 bumps (successful reaches but where neither the Palmar reflex or an intentional closing of the hand creates a grasp), 8 Palmar bumps (successful reaches that activate the Palmar reflex but where the closing grippers fail to temporarily bind the object to the hand), and 2 weak grasps (successful reaches that do gain control over the block, but in a way that is not stable and the block is dropped before an intentional ungrasp). While the agent cannot distinguish between upright blocks and blocks that are laying down, we also note three of the 21 successful placements where the block is upright at the end of the trial. Any orientation satisfies the agent's definition of being in a location, but upright block results like these are important and desirable as it would enable the agent to continue to interact with the block in autonomous action practice in future work.

Another depiction of the pick-and-place results is given in Figure 7.24, where boundaries of the target mask of all 48 initial object positions are shown in colors according to their results. This figure also shows the number of candidate nodes for each target mask that was generated. All 8 of the rejected positions are in the furthest corner of L_2 where the graph is most sparse, and the 3 trials with miss results are also on the far edge with a very low number of candidates. The failure of the reach in these cases was due to the selection of a final node for grasping that actually had false positive intersection features with the target, and should not have been considered as a candidate. For the rest of the placements, the successes appear to be mostly clustered in regions with more candidates and where these candidate nodes correspond to poses that are well aligned for grasping. The localization is likely related to where in the image the agent is best able to identify desirable palm vectors and accurate masks and depth ranges (which appears to be generally about halfway up the vertical axis before the zoom used in Figure 7.24). An alternative explanation is that it is localized to the highest number of candidates, but this would not explain failures in the green location with more than 40 candidates while many succeed in the blue location with less than 20. As the initial positions become more distant from these high performance areas, the results tend to shift to weak grasps and palmar bumps, and then bumps and misses at the furthest points. We can also note that the 3 trials where the block was placed upright all began with the block in the blue location. In these cases, the block was grasped from an upright position with the hand and posed very similarly to how it would be posed at n_{79} for placing into the green location, so the block was upright again when it was placed.

Prior to this experiment, the agent had only performed place actions in isolation, where it had been presented with the block instead of independently performing a grasp of an object on the table. A key observation from this experiment is that the success or failure of the pick-and-place

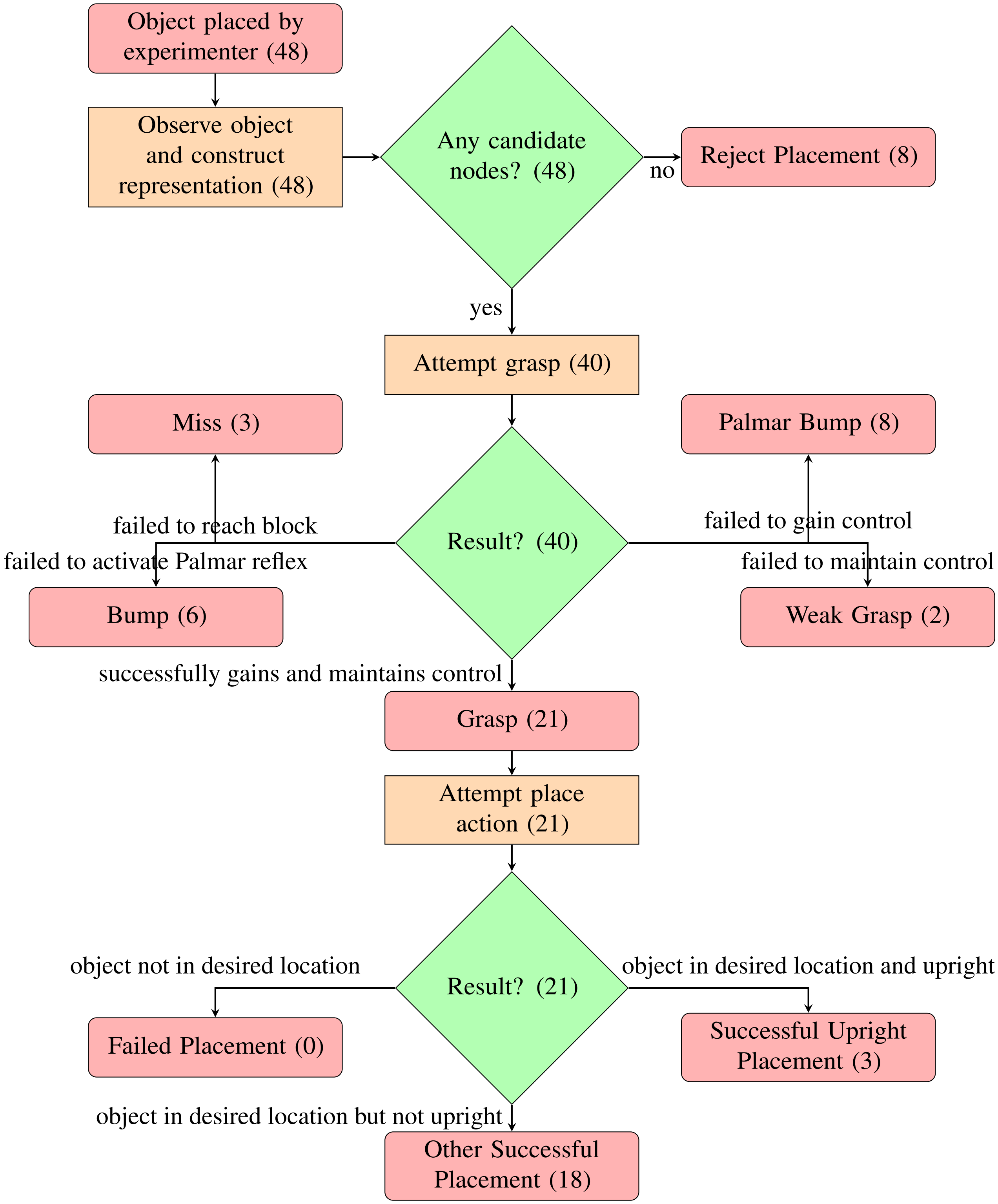

Object placed by experimenter (48)
Observe object and construct representation (48)
Any candidate nodes? (48)
no
Reject Placement (8)
yes
Attempt grasp (40)
Miss (3)
Palmar Bump (8)
failed to reach block
failed to gain control
Result? (40)
failed to activate Palmar reflex
failed to maintain control
Bump (6)
Weak Grasp (2)
successfully gains and maintains control
Grasp (21)
Attempt place action (21)
object not in desired location
object in desired location and upright
Result? (21)
Failed Placement (0)
Successful Upright Placement (3)
object in desired location but not upright
Other Successful Placement (18)

Figure 7.23: A flowchart detailing the 21 successful pick-and-place trials and a breakdown of the errors that caused the 19 failed trials, beginning with experimenter placement in the top-left node. Nodes are annotated with the number of trials that reached them.

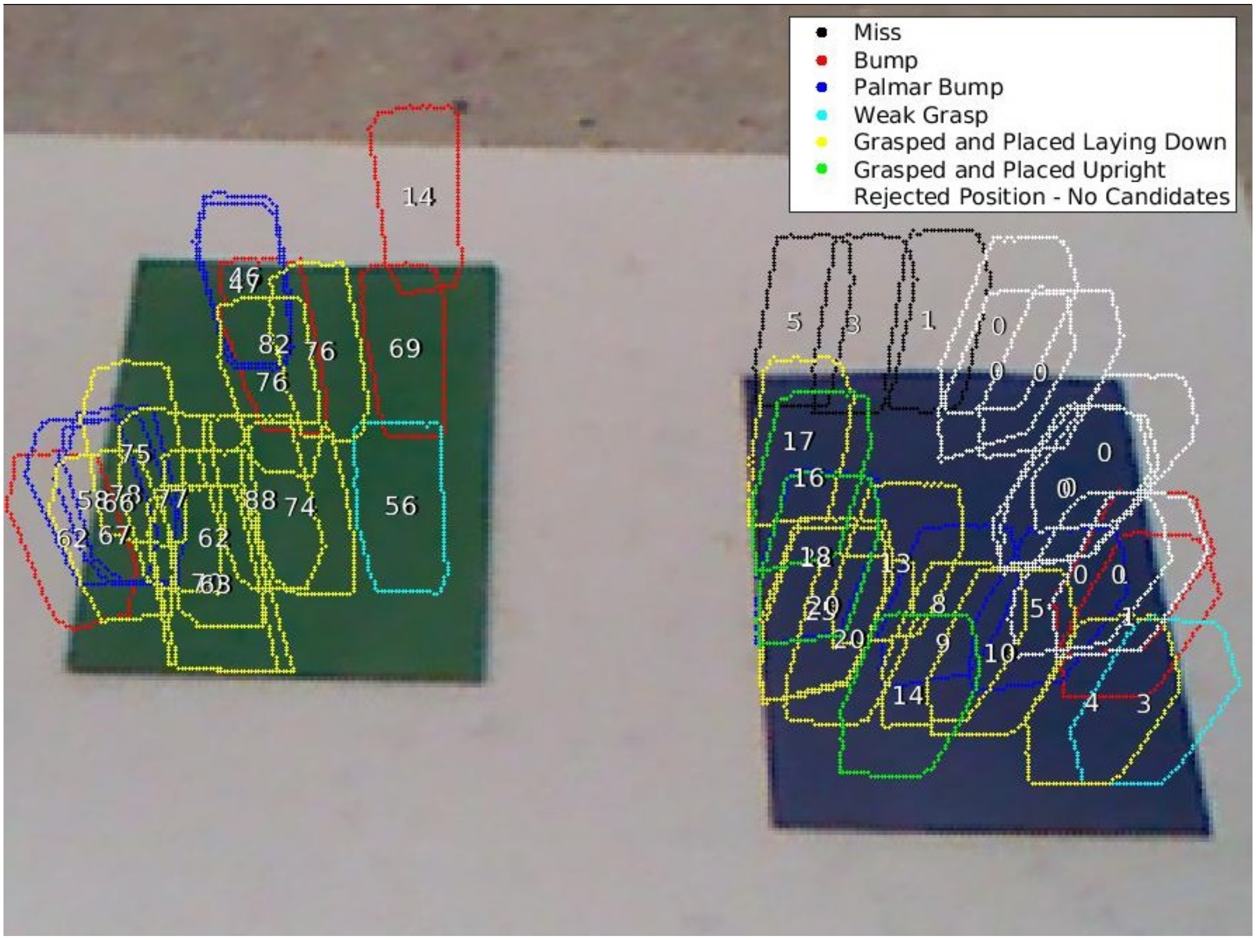


Figure 7.24: The 48 initial positions of the target object when testing the reliability of the agent's pick-and-place action. Each position is shown as a boundary of the target mask that corresponded to the block when placed upright at random coordinates within one of the locations by the experimenter. Each boundary is superimposed on a zoomed-in image of the locations without the block or arm present. Each position is annotated with the number of candidate nodes found when planning the grasp component of the pick-and-place. The relative sparsity of the graph in the blue location L_2 is evident in the lower numbers of candidates. 8 of 28 positions generated for L_2 were rejected by the agent for having 0 candidates. Of the 20 trials that were attempted from L_2, 11 succeeded, outperforming the 10 out of 20 successes when grasping from the green location L_1. No straightforward threshold or trend exists relating the number of candidates and the likelihood of success, other than failing each trial with less than 3 candidates. One trial succeeds with 4 candidates while another fails with 69. While it is necessary to have enough candidates for one to exist with reliable features, outside of localized areas where the images have less noise and the image processing is more accurate, having too many candidates appears to increase the chance of selecting a node with false positive desirable features, leading to a failure. This image resembles the visualizations of grasp results because the current method for pick-and-place only fails if the grasp fails, as the place action remains 100% reliable. Note that we require a successful place action to end the agent's control over the object that was maintained during a grasp, so this makes weak grasps a failed grasp, rather than a failed placement. Similarly, even if the agent would bump the block into the desired location without a grasp - a possible edge case that did not occur in these trials - the agent would not consider this a successful placement because it can determine it never had full control of the block with a grasp.

sequence was entirely dependent on the success or failure of the semi-reliable grasp action. The overall success rate of the pick-and-place sequence was 52.5% (21/40), but the success rate of pick-and-place sequence given that the grasp was successful was 100% (21/21). Importantly, this allows us to conclude that our methodology where the agent learned to ungrasp and place without performing a typical grasp first yielded a placing policy that is 100% reliable for both locations whether the block was grasped from the table or with experimenter assistance. The reliability only being limited as a result of the grasp action also makes it clear that to improve performance the agent should focus on improving the grasp action in future work.

7.6 Conclusion

In this chapter, the agent has learned to make two new actions reliable and has sequenced actions to construct a higher order action that gives a significant amount of control over the objects in the environment and will facilitate future learning, by continuing to identify unusual events and using the pattern in this work, or by using the experience gathered in another approach, such as reinforcement learning or deep reinforcement learning to advance to smooth, expert forms of the early actions.

In Section 7.2, the agent observed many examples of the typical result when changing a degree of freedom after a grasp. When changing any of the six major (proximal) joints of the arm to move the hand in space, the grasp is maintained, and the grasp is also maintained when rotating the wrist with the most distal joint w2 or attempting to close the grippers further by decreasing the aperture a. However, commanding an increase $\Delta a \geq 5.6$ does result in a physical opening of the grippers that loosens the grip on the grasped block, and in one case with $\Delta a = 16.75$ (Figure 7.1), the grippers open enough and the block is oriented in a way that it falls out of the hand, ending the grasped state. The agent can verify that this has occurred by moving the arm along any trajectory and noting that the block's current mask no longer intersects with each node in the trajectory's stored mask, so the property of following the motion of the hand is ended by this successful *ungrasp*.

In Section 7.3, the agent conducts ungrasping trials, and now only modifies a since no other degree of freedom ever produced an ungrasp. The agent also rejects any $\Delta a < 5.6$, as it has found these values to be below a minimum threshold of intended change to have the grippers physically move. Out of 50 trials, the agent observes 6 more successful ungrasps, and identifies a threshold $\Delta a^* = 21.5$ where no attempt with a larger Δa has failed. An additional 50 trials with these larger commanded openings of the grippers produce 48 successes and allow the agent to revise Δa^* to 41.94 so that its threshold property still holds. These results lead to a decision to extrapolate, and the agent is able to ungrasp with full reliability for all following experiments by setting the grippers to 100% open.

In Section 7.4, the agent considers the qualitative locations L_1 and L_2 on the tabletop for the first time, and discovers that for either location it is more rare for random ungrasps to send the object into the location than outside of the location. This becomes the next unusual event to focus on, and the agent begins to define a *place* action that ends a grasp and results in the new quasi-static location of the block being in a specific location. Since the agent already has learned to end grasps with a fully reliable ungrasp, the only necessary learning remaining to refine an ungrasp into a place is how to select the pose for the hand where the ungrasp will be performed. By examining the ungrasps from section 7.3 the agent identifies a clear trend that as the image-space center to center distance decreases, the reliability of a node for placing increases. The suitability of this feature is evaluated on the five nearest nodes to each location of interest, and the agent finds that nine of the 10 nodes are 100% reliable. In both cases, the single closest node is 100% reliable for placing, and the agent will perform all placements from the closest node to the intended location in this work. Learning to choose the closest node was first done in the original environment for our agent and the first set of images for the Explored PPS Graph, and a sample place action can be viewed at https://youtu.be/KBeuamdgDkE. As can be seen in the video, all trials for learning to ungrasp and to place were performed from a starting state with the block already in the hand, to avoid additional time requirements when trials with failed grasps would be rejected.

Before the experiment in Section 7.5 was performed, the agent moved to a new environment and collected a new set of images for the nodes of the Explored PPS Graph (as discussed in Section 6.5). With the changed positions of the robot, table, and camera relative to each other and the new set of images were taken, the closest node to the green location was n_{79} and the closest node to the blue location was n_{1251}. Although the closest nodes were different from the original image set, the agent used the same policy of using these closest nodes to place into the green and blue locations and found placing to still be 100% reliable.

In Section 7.5, the agent combines the newly learned place action with the most reliable grasp action from Chapter 6 to form a policy for the *pick-and-place* action, which takes the block from one qualitative location and moves it to a different qualitative location. The place component of the pick-and-place is 100% reliable, demonstrating that the ungrasping and placing techniques that were learned starting with a block in the hand do not receive any essential assistance from the experimenter - the placements work equally well when the state of the block in the hand is one that arises from a typical independent grasp. However, the overall reliability of the pick-and-place action is only 52.5% (succeeding in 21 out of 40 trials) because it cannot succeed without a successful grasp, and the grasp action is at best semi-reliable (57.5% was the best success rate seen in Chapter 6). An example of a pick-and-place action with the current best policy can be viewed at https://youtu.be/afiG2o8Rlgc. Identifying additional features and continuing practice for grasping will be the most efficient way to improve the reliability of the pick-and-place action.

Given the limited information provided to our agent throughout this work and the limited assumptions of prior knowledge and innate capabilities, the result of 52.5% successful pick-and-place actions is significant progress. As the agent moves from this early form of the action, it may continue to improve its reliability and the efficiency with which it can be executed. With the current learned semi-reliable policy, we have demonstrated that an agent can apply the pattern of learning from unusual events all the way from motor babbling with single joints to an action as complex as pick-and-place. Achieving semi-reliable pick-and-place is also significant for expanding the potential application of this work to other works that may use pick-and-place or its component actions as action primitives.

Along with improving the reliability of the pick-and-place action, a good candidate for a next step after this work is to allow the agent to practice through play, autonomously choosing targets and executing pick-and-place actions in continuous cycles. At this time, we can already show a proof of concept for this method, as seen in the video found at https://youtu.be/xnNfqUmlCKI. In this video, the agent is able to perform a grasp from the blue location and place into the green location, and then observe from the home node that the block is in a position where it is upright and can be grasped again. After taking 12 seconds for this observation and trajectory planning, the agent performs a second pick-and-place action to take the object from the green location back to the blue location. The agent could hypothetically continue this back and forth process indefinitely, each time gaining experience that could improve all of the component actions. The ability to practice the actions in a self-guided setting will also allow the agent to re-attempt failed component actions. If the agent observes that the grasp has failed, it may attempt it again with a new plan until it succeeds before moving on to attempt the placement, which cannot succeed if the grasp preceding it failed. And if a placement would fail, the agent could return to the grasp portion to regain control and attempt the placement again. The ability to retry difficult cases presents the agent with valuable experience and may increase the success rate from what can be achieved in one-shot actions.

In order to take the step from performing pick-and-place actions in isolated trials that each begin after the block is set up by the experimenter to repetitive pick-and-place actions in a self-guided learning by play setting (such as the one formalized in [42]), the only issue that must be resolved is reducing the number of trials that leave the block prohibitively difficult for the agent to grasp. Many failed grasp attempts and many successful placements knock over the block, leaving it with a very low profile near the table. The agent was not allowed to collect nodes this close to the table's surface to avoid collisions during motor babbling. As a result, few nodes appear to have intersecting depth ranges, and of those that do, many must be modified far enough from their stored center that the local Jacobian estimate will be inaccurate. It is also the case that almost no nodes in the graph have the grippers pointing downward to approach the block from above, with a crane-like method. This is likely due to the initial pose for n_0 being mostly horizontal, and that the random motor babbling changes to the joints would have to change all of the vectors by a significant amount and within a particular range all for the same node to have even one with the hand pointed downwards. This suggests one improvement would be to gather additional nodes in another round of motor babbling starting from a crane-like position. These new nodes would be connected to each other and any existing nodes by the same heuristic of edge length as before. An alternative to better handling of targets that are not upright blocks is to increase the frequency with which objects stay upright. This could involve succeeding with grasping so that the block is not knocked over by a bump, and could also involve using nodes for the place action that have been identified as likely to leave the block upright. The agent cannot currently distinguish upright blocks from other orientations, but has observed both orientations, so it should be possible to learn a classifier. Once the agent can identify upright placements, it can begin to use only nodes that produce them reliably, and then has much more potential to perform long loops of practicing actions through play.

In the following chapter, we will continue the discussion of future work that has been suggested here and throughout the dissertation, as well as the contributions and significance of the work in its current state.

CHAPTER 8

Discussion

8.1 Contributions and Importance

8.1.1 The Peripersonal Space Graph Model

We present the Peripersonal Space (PPS) Graph, a model of the space surrounding the agent that it can reach with its manipulators. A PPS Graph $\mathcal{P} = \langle \mathcal{N}, \mathcal{E} \rangle$ is composed of a set of nodes $\mathcal{N}$ and a set of edges $\mathcal{E}$. The structure of the PPS Graph resembles a probabilistic roadmap (PRM) [25] and the Visual Roadmap (VRM) proposed in [39], which also stored types of visual and configuration data. Unlike in [39], the PPS Graph is constructed through unguided exploration and the agent must autonomously learn to interpret and use the information stored in the graph. The formalization for the general PPS Graph model is given in Section 3.2.1, and the construction of a specific implementation, the Explored PPS Graph, is discussed in Section 4.3.

The exploration that the agent uses to build the PPS Graph is carried out by motor babbling with the arm. Beginning from an arbitrary pose, each subsequent pose is generated by moving to a configuration that differs from the current configuration by a small randomly sampled amount for each joint angle. At each pose, the agent records a node n_i, which stores the configuration $\mathbf{q}_i$ that was held to produce the pose and the visual percept $I_{\mathbf{q}_i}$ of the environment while the arm is in that pose. By storing this pair of corresponding percepts, each node maps from a configuration to the associated image features for the pose it produces. The set of all nodes then serves as a discrete mapping between the configuration space and the image space. Even though the nodes are produced with random motions, they provide coverage for most of the spatial region that is in view and in reach of the agent, such that most points in peripersonal space are near at least one node.

The motor babbling process also produces the first edges of the graph, connecting each n_i to n_{i+1} with an edge $e_{i,i+1}$ because the motion between the nodes has been demonstrated to be feasible when the arm was moved from n_i to n_{i+1}. In general, edges will represent affordances for safe motion between the nodes they connect. As it is not feasible to test the motion between all pairs of nodes n_i and n_j, additional edges $e_{i,j}$ are added according to a heuristic. If $||\mathbf{q}_i - \mathbf{q}_j||$ is less than the median length of all edges that were added during motor babbling, n_i and n_j are assumed to be sufficiently close for the move between them to be safe, and $e_{i,j}$ is added to the graph. All graph nodes are in a single connected component, so planning trajectories to move between any two nodes, however distant they are in configuration or image space, is possible by finding a graph path between them. As almost all points in peripersonal space are near at least one node, and such a node can be identified by visual similarity (overlapping masks or depth ranges, or nearby image-space centers), trajectories planned as paths along PPS Graph edges may move the hand near almost any point in peripersonal space.

We also present a novel application of estimated Jacobian models for the relationship between changes in configuration and changes in image space. As the global Jacobian is prohibitively complex for the infant-like agent to learn and use, the estimates are linear approximations calculated for the neighborhoods of individual nodes, and remain sufficiently accurate when these changes are small enough that the configuration and position in image space remain local to those of the node used for the calcuation. These estimates can be computed using the information stored in a node n_i and its neighborhood $N(n_i)$, the set of all nodes it is directly connected to by an edge. These estimates are calculated to extend the Explored PPS Graph in Section 4.4. Often more important than using a local Jacobian estimate to predict how a perturbation of the joint angles will affect the position of the hand in image space is the ability to use the pseudo-inverse of a local Jacobian estimate to predict the adjustment to the joint angles necessary to produce a desired change in image space (such as to move the center of the hand to the center of an observed foreground object). This allows a more precise move to a desired position since it is not limited by the closeness of the nearest stored pose. In addition to the capability to move to configurations that have not been previously visited, the extension to a locally continuous model allows the agent to more fully and more accurately represent peripersonal space.

It is also noteworthy that the PPS Graph model can be constructed and used without any forward or inverse kinematics models that relate the configuration of the robot to a 6D pose of the hand in standard coordinates for the workspace. Further, the agent does not have access to or make an attempt to recreate the workspace coordinates and orientation of the hand or other objects. The agent only represents the self with the joint angles of the arm and the hand's position and orientation in image space, and as the agent can only sense nonself objects with vision, they can only be represented by their position and orientation in image space. These simple representations of the hand and foreground objects are sufficient for the agent's actions in this work, and more detailed information, such as geometric models, point clouds, or grasp points are not provided to or needed by the agent. Instead of using inverse kinematics to plan a reach to an object, the agent will select the node that best matches the node visually according to the criteria for the action, and then map from that node's visual features to the configuration that is also stored in the node. The agent determines that this configuration, modified as necessary according to the node's local Jacobian estimate, will be the final position. Finding the shortest graph path to the selected node completes the trajectory and produces a reliable plan for completing the action.

The nodes, edges, and neighgborhoods of the PPS Graph allow the agent to make safe motions with predictable results throughout the environment. The agent may learn the typical results for these motions and then observe unusual results that qualitatively differ, and repeating these becomes the goal of new actions. The PPS Graph has further significance for action learning as it provides a mapping between the internal state of the arm (sensed as a vector of joint angles through proprioception) and the external state of the arm (sensed with a camera that provides an RGB image and a disparity image of the environment). This mapping allows the agent to model the consequences of changes to its configuration, and to determine the changes to its configuration necessary to interact with the environment in specific learned ways that reliably accomplish self-defined goals.

8.1.2 A Method for Autonomous Learning From Unusual Events

We present a method that applies the principles of intrinsic motivation to allow an embodied robotic agent to learn to perform self-defined actions reliably. See Sections 1.3-1.4 for a full description of this method, and the representation and features learned are discussed throughout Chapter 3. We also provide a high level abstraction of this procedure in Figure 8.1. This method begins with the agent performing a known action repeatedly to observe its typical results, as well as a small number of unusual events where the results were qualitatively different from the typical results. These unusual events will fall into one or more clusters corresponding to the same type of event, and the agent will focus on one cluster at a time. The agent will define a new action that has a goal of repeating this type of unusual event. The definition of actions according to the experience of the agent is a key feature of this work, and contributes to the justification that the agent learns from unguided exploration. In many robotics approaches, the agent is provided with a detailed definition and policy for the actions that will be performed as innate knowledge, so that the agent can immediately perform them successfully and the execution may be fine-tuned by expert designers. In alternative robotic learning approaches, the agent may still be explicitly provided with the goal of the action it is to learn. Methods such as reinforcement learning and deep reinforcement learning do not require an explicit statement of the action or goal, but provide a reward signal that implicitly conveys information about the goal to the agent. Our method avoids providing this information in any form, and the actions created are entirely dependent on what the agent has observed. The methodology we present supports the creation of any number and any type of actions and could allow the agent to pursue learning to make these actions reliable in any order or simultaneously, but to ensure timely evaluation in this work, the agent has been prevented from creating actions that are not discussed, even if the observations of unusual events are available, and continues with a single action until its evaluation is complete.

1. Begin with a set of known actions $\boldsymbol{A} = \{\text{move the arm to a selected node}\}$
2. Repeat these steps, producing a new reliable or semi-reliable action each iteration:
    - (a) Select an action a from $\boldsymbol{A}$ to perform
    - (b) Execute multiple trials of a and observe the results of each trial
    - (c) The typical result of a is learned by determining the most common type of qualitative result observed.
    - (d) Clusters of trials with results that qualitatively differ from the typical result are examples of unusual events.
    - (e) Define a new action b with a goal to repeat one of these newly observed types of unusual events. The initial policy for b is equivalent to the policy for a.
    - (f) While b is not yet fully reliable, and while the reliability of b is improving:
        - i. Generate a new feature, selection criterion, or sensorimotor technique
        - ii. Modify the policy of b to consider the newly generated aspect
        - iii. Evaluate the modified b action over a set of trials
        - iv. Keep the modification if b has an increased success rate, otherwise revert to the previous policy
    - (g) Add b to the set of known actions $\boldsymbol{A}$ using its most reliable policy

Figure 8.1: The agent's learning procedure, which produces a set of infant-like early actions from autonomous experience. The agent is intrinsically motivated to define new actions to repeat unusual events and learn to make them reliable. In this work, features, criteria, and techniques are sometimes provided instead of autonomously generated to constrain the search space and allow more efficient evaluation. This learning focuses on producing actions that reliably achieve their goals, but may be awkward or inefficient when performed. A second learning process in future work could transition from these to more skillful late actions with smooth, efficient executions.

Once an action has been defined by the agent, the agent will be intrinsically motivated to make the action reliable. Once the action is reliable, its outcome is predictable and the agent will be able to repeat the unusual event when desired. In order to increase the reliability of an action, the agent will practice the action as planned by the most reliable method available (which may initially be random motion or the policy for a previous action that sometimes produces the new unusual result). Examination of success and failure cases will allow the agent to identify features and contingencies that can be used to increase the likelihood of success. In general, the agent will be searching for candidate nodes, trajectories, or methods to use in future attempts that have similar properties to those used in prior successes. At this time, our implemented model differs from the proposed ideal methodology in that some individual features or feature sets to select from are provided to the agent, and the agent performs the selection of the best feature or features and is always responsible for finding the most reliable values of each feature autonomously and with domain-general statistical methods. This assistance consists of providing exhaustive feature sets or instruction to perform grid searches over a full range of values when possible, but without the scope restrictions of this work the agent can increase the amount of learning that it does autonomously by always generating features without input.

The process of learning from unusual events is demonstrated well by the reach and grasp actions, which are also important to this work given their fundamental roles in an agent's ability to influence its environment. Prior to learning the reach action, the agent has learned to move the arm and has constructed the PPS Graph, which allows a wide range of motions throughout peripersonal space using safe graph path trajectories. As the agent practices the move action in the presence of a foreground object (a distinctive yellow rectangular prism block in this work), it observes the typical result of these moves, where the hand moves back to the position where it was observed when first visiting the randomly chosen destination node and the appearance of the background is unchanged. However, given the wide distribution of the nodes being moved to, with enough trials the agent also observes the unusual result of a change in the appearance of the object in addition to the movement of the hand. This change is observed as a significant change in the set of pixels in the binary image mask of the object when segmented before and after the move. This method only relies on a comparison of 2D image projections of the object, and allows change-detection without detailed knowledge of the nature or structure of the object. Importantly, the change persists even once the hand has been moved away or moved back to the home node to observe the result, ruling out that the change was a temporary occlusion. This significant persistent change characterizes the bump type of unusual event, and a reach is defined as an action that causes a bump.

Given that the bumps were observed while performing move arm trajectories to random nodes, the agent will continue to use move arm trajectories to perform the reach, and will make the reach action more reliable by learning better criteria for the selection of the final configuration. The first improvement comes from finding which of a set of intersection criteria have been most reliable for past reach attempts. The agent finds that the conditional probability of a bump is highest when the final node of the trajectory has a stored binary image mask and depth range that have nonempty intersections with the binary image mask and depth range of the target object in the current visual percept. It also determines that the intersection of masks is most predictive of success when the mask for the smaller grasping region of the hand, the "palm", is used rather than the mask corresponding to the full hand or the mask containing all pixels the hand passes through during the final motion. For each placement of the target object, the agent can identify a set of candidate final nodes that have these most reliable nonempty intersection features, and successfully causes a bump more often by selecting one of these candidates instead of a random final node. To improve reliability further, the agent learns to select the most reliable candidate node instead of a random candidate. The agent considers four distance measures, and by each measure observes that the conditional probability of a bump decreases as the distance between the image-space center of the palm and the center of the target increases. The Euclidean distance between image-space centers is identified as the measure that best differentiates reliable and unreliable candidates, and the agent begins planning reaches as a trajectory to the final node candidate with a minimum distance by this measure. The final improvement to the reach action policy that makes it 100% reliable is the use of the pseudo-inverse of the local Jacobian estimate for the selected closest final node to modify the image-space center of the palm to be equal to the target's image-space center. Within the margin of error of the estimate, which will be smaller when using the nearest node, this produces a distance of zero between the centers, which the agent correctly predicts will be highly reliable.

A strength of our learning from unusual events method is its ability to be applied iteratively to learn progressively more complex actions. Once the reach action is reliable, the agent observes interactions with the target object regularly as it continues to practice reaching, and these will typically be simple bumps. However, if the gripper aperture is also changed during these trajectories, it is possible for the agent to observe a new unusual event where the reach produces a persistent change that is not a typical bump, but instead an accidental grasp. The agent can identify that a grasp has occurred by the unusual behavior of the block after the grasp, as it moves along with the hand. However, if the agent relies only on random reach trajectories to explore the interactions between the hand and the object, then accidental grasps will be very rare indeed. An accidental grasp would require first a reach with the hand, with proper orientation and the grippers open, to a pose surrounding an object, followed immediately by a random motion to close the gripper securely on the object. The product of these two low probabilities would be very low. By contrast, in human infants, an object touching the palm triggers the Palmar reflex that closes the fingers, eliminating one of the two low-probability factors. We add a break-beam sensor to our robot's hand to act as a simulated Palmar reflex, sending a signal that commands the grippers to close when an object passes between them, breaking the beam. The presence of the simulated Palmar reflex removes the additional condition of timing the closing of the grippers correctly, allowing some accidental grasps to be observed, though they are still rare, as most reaches do not approach the target in the correct manner to allow a grasp. Given the important role of the simulated Palmar reflex for our agent, one might speculate that the Palmar reflex is present in humans as an evolved mechanism for gathering enough relevant information to allow grasping to be learned. Searches over potential feature values allow the agent to learn to approach the target with the grippers fully and to approach in a direction that aligns several vectors describing the motion, and that is approximately perpendicular to the major axis of the target. The orientation of the hand with the wrist is also a factor in grasp reliability, and the agent can reuse angles for the wrist joint w2 that have been observed to succeed before. Planning the final motion according to these criteria allows grasp trajectories to be 57.5% reliable.

Grasping (and the pick-and-place action of which it is a component) stand out from the other actions that have been made 100% reliabile within the scope of this work. But why is it the case that grasping remains only semi-reliable? The task of learning the grasp action is more difficult than learning any of the other actions, given the number of factors that must all be accounted for simultaneously. This difficulty is even more pronounced for our embodied robotic agent with a parallel gripper hand, which is structurally inferior to a human's hand and must grasp in a much more specific manner. Compared to the natural Palmar reflex, the simulated Palmar reflex also limits the types of grasp approaches that are viable, as objects must enter the space between the grippers through the front of the hand, past the tips of the gripper fingers, rather than the top or bottom. The mechanism of closing the grippers to produce a grasp also requires a more immediate, one-shot success, whereas the flexibility and dexterity of the human hand allow adjustments over time to create a secure grasp after a subpar first contact. Grasping is also a more demanding sensorimotor task. Unlike reaching where the final configuration is sufficient if it produces a significant collision between any parts of the hand and object, the preshaping and final configurations for grasping must be precise enough to satisfy all conditions for a grasp with the position and orientation of each component of the hand. We speculate that highly reliable grasping may require improved perception or a more informative representation of the workspace than the distorted image space. With these difficulties considered, the agent's significant progress to semi-reliable grasping is a valuable contribution and strengthens the demonstration of our method of learning from unusual events.

Observing and learning from unusual events also allows the agent to learn to ungrasp and place in this work. The method may be applied additional times to learn additional manipulation actions. For example, the agent may learn to perform a more specific type of grasp from unusual grasp action results, or the agent may identify rare types of placements, such as when the block lands upright or is stacked on another object, and learn actions to repeat these reliably. It would also be possible to return to the move arm action and observe other types unusual events than bumps and grasps.

An advantage of the learning method we present is its generality, which extends beyond the learning of manipulation actions. Given an agent physically capable of other types of actions, unusual events could be observed that prompt the definition of actions for navigating a large scale space, communicating with other agents, processing images, or numerous other domains. Our method also allows a single learning agent to learn multiple actions, and to utilize significant transfer of knowledge from an existing action policy to the policy to a new action so that the same features need not be relearned each time.

8.1.3 Learned Manipulation Actions

While high success rates for specific actions were not a key goal of this work, and we do not intend to compete with other methods in terms of success rates, the actions that can be performed are significant evidence for the strength of the learning method. While the grasp action, and the pick-and-place action that the grasp is a component of, remain only semi-reliable at the conclusion of this work, their reliability is a considerable improvement over the performance of random action baselines and is impressive in light of the limited information provided to the agent. Each learned action policy is also significant as one that may be implemented for other embodied robotic agents in other works, whether those works intend to continue to use the infant-like, but reliable (or semi-reliable) early actions or to develop more skilled late action policies after using the early actions for initialization. The actions learned in this work are:

- **Move Joint –** The agent is able to use the robot's built-in Joint Position Control mode to consistently bring any joint angle within 0.0087 radians of the desired set point. This is the primary control mode used throughout this work. The agent is also capable of using the robot's built-in Joint Velocity Control mode and a simple control law that commands a velocity of $-k_p(x - x_{set})$ to bring any joint angle within 0.01 radians of the desired set point while the velocity was less than 0.01 radians/second. When $k_p = 1.25$, the motion appeared to have a desirable property of being nearly critically damped, with changes in k_p allowing the production of overdamped, underdamped, and divergent motions. Further detail about the move joint action can be found in Section 4.2.

- **Move Arm –** The agent is assumed to be capable of combining move joint actions in parallel to produce the move arm action. With this move arm action the agent uses Joint Position Control mode to bring all joints to the desired set points simultaneously and without increasing the threshold for error. This action can be combined with the mapping provided by the PPS Graph to move the arm in a way that reproduces a previously observed visual state reliably, by returning to the configuration stored in the same node. By estimating a local Jacobian in the neighborhood of an observed node, the move arm action may also be used to bring the hand to positions in image space that were not previously visited, with accuracy dependent on the number of nearby nodes and their information. The edges of the PPS Graph also provide the agent with a feasible, safe trajectory from the initial configuration to a desired configuration, an improvement over the capabilities of the Joint Position control mode alone, which moves along a linear interpolation between the configurations and may be unsafe for long moves. From a developmental psychology perspective, the trajectories used by this move arm action along the edges of the graph may be desirable for attempts to model the movements of infants that also tend to contain jerky submotions, as observed in [49]. Further detail about the move arm action can be found in Sections 4.3-4.4.

- **Reach (100% reliable) –** The agent defines the reach action after discovering the unusual event of a bump. The agent's reach action was defined with a goal of repeating the bump type of event, and is 100% reliable by using a trajectory that ends at a configuration near a well-selected final node, and that has been modified according to that node's local Jacobian estimate and its displacement from the target. All of these reaches were successful at the intended final node, rather than accidental successes earlier in the trajectory. This is a significant increase from the baseline of causing bumps in 20% of moves to randomly selected nodes, where half of those successes were at early, unintentional points in the trajectory. Further detail about the reach action can be found in Chapter 5.

- **Grasp (57.5% reliable) –** The agent defines the grasp action after discovering that some reach trajectories produce accidental grasps while the Palmar reflex is present. The agent learns to more reliably repeat the grasp event by choosing a well-aligned final motion, orienting the wrist, and fully opening the grippers before the grasp approach. The most reliable grasping method in this work is 57.5% reliable, which is a significant increase over the baselines of 2.5% when performing moves to random nodes and 12.5% when performing reach trajectories. Both baselines were determined in trials where the Palmar reflex was active, as the reflex is necessary to make observing accidental grasps tractable. Because the grasping method changes the final node selection to prioritize alignment during the final motion, the success rate of the reaching component of the grasp (how many attempted grasps at least caused bumps) is reduced to 92.5%, but reaches planned with the learned reach action policy can still be used to bump the object with 100% reliability. Further detail about the grasp action can be found in Chapter 6.

- **Ungrasp (100% reliable) –** The typical behavior of an object that has been grasped is to move along with the hand. The agent defines an ungrasp action to release an object and repeat the unusual event where the previously grasped object stops moving along with the hand. A technique of fully opening the grippers is 100% reliable, a significant improvement over an initial motor babbling method of adding a small delta to the gripper aperture, which ungrasped in 5% of trials, and all methods that adjusted other degrees of freedom, which never ungrasped the object. Further detail about the ungrasp action can be found in Sections 7.2-7.3.

- **Place (100% reliable) –** The agent observes that typical ungrasps do not result in the object coming to rest in a qualitative location of interest, but the object does land in the location of interest as an unusual event. The agent defines the place action to repeat this event, and learns to accomplish it reliably by moving to the node nearest to the location in image space before opening the grippers to perform the ungrasp. This method is 100% reliable for placing into either of two locations, one of which where the PPS Graph is sparse and another where it is dense. This is a significant improvement over the baseline observed when ungrasps were successfully performed with the hand at random nodes. To use the ungrasp action evaluation results as as a baseline, 34.5% of successful ungrasp trials ended with the object in the location where the graph is dense, 25.5% ended with the object in the location where the graph is sparse, and 40% ended with the object outside of both locations of interest. When the agent instead uses the learned place action policy of moving to the nearest node before performing an ungrasp, the object is placed into the intended location in 100% of trials. The agent also observes that placing from any node near the center of the location is highly reliable, which could be useful if movement to the nearest node is blocked. Further detail about the place action can be found in Section 7.4.

- **Pick-and-Place (52.5% reliable) –** The pick-and-place action is created using the assumption that the agent can combine previously learned actions in sequence. For evaluation in this work, a pick-and-place action always consists of only a grasp action followed by a place action, but the agent could optionally perform move and reach actions at some points during the sequence without violating the preconditions of any of the component actions in the sequence. The reliability of the pick-and-place action is only limited by the reliability of the grasp component, which was 52.5% during the pick-and-place evaluation. In all trials with a successful grasp, the place component was successful. The pick-and-place action may also be used to move an object back and forth between two locations of interest - a useful way of independently generating additional opportunities to practice - but only as long as the target object continues to be placed in an upright orientation. Grasping is not yet reliable for other orientations of the object, so a placement that results in one of these orientations ends the cycle. Further detail about the pick-and-place action can be found in Section 7.5.

8.2 Directions for Future Work

Throughout this dissertation, we have included discussion of opportunities for future work. These discussions have included cases where it may be possible to improve upon results with the consideration of new features, where it may be possible to increase the autonomy of the agent with changes to the methods and assumptions, or for new directions that can be investigated as a continuation of this work. While these cases are identified in context, we highlight some key directions here.

In order to guarantee the timely evaluation of the agent's learned actions within this work, we have needed to provide some guidance that is not present in the ideal unguided method we propose. The removal of this guidance and an evaluation over the agent's autonomous behavior is an important step for future work to build on and strengthen the contributions in this work. As discussed above, the procedure for generating actions autonomously may produce a wide variety of actions. For the sake of timely evaluation in this work, we artificially restrict the agent's focus to only certain actions. In future work, the agent may be allowed to create actions for any event that it observes, and may practice executing the actions to make them reliable in any order, or in intermingled trials. This extension could be performed by applying the framework for open-ended action learning suggested by Kumar et al. (2021) [26], which includes a formalization for the agent's selection of an action to attempt given the current environmental state and a target for that action. A flexible learning order will also be significant for future work, as the experimenter-determined learning stages are a source of guidance that could be removed to make the agent more autonomous and more natural. Removing the learning stages will also allow for intrinsic motivation to be more explicitly modeled and used by the agent to choose the action it is currently most ready to practice and learn, rather than assuming the actions in each stage correspond to the actions with the highest intrinsic reward. Another step toward making the agent fully unguided is to give the agent sufficient capabilities for generating its own features and defining its own search problems, to replace portions of the current algorithm that assume the results of the generate and test process by providing a useful feature or set of features that is analogous to what would be found by a successful search. We claim that the resulting action qualities and success rates would be the same with each facet of guidance removed, but the proposed unguided method must be tested to support this claim.

There are also specific results that may be improved upon in future work. The grasp action is only semi-reliable at the conclusion of this work, and may be improved. The result of 98% successful grasping in Luo et al. (2018) [32] demonstrates that reliable grasping is attainable using sensorimotor learning models that can be trained from agent experience. As an alternative to their assumed kinematics knowledge and 3D object pose representations, we have proposed the Improved Explored PPS Graph that allows more detailed segmentation of the parts of the hand and that adds an assumption that the most distal joint w2, which rotates the hand, and the gripper aperture are degrees of freedom of the hand, rather than degrees of freedom of the arm as the other six arm joints are. This separation allows nodes to be defined with only the degrees of freedom of the arm, and at each node images may be recorded with multiple settings for the degrees of freedom of the hand, providing more data on how the orientation, size, and appearance of the hand is changed. The additional data reduces the reliance on estimating or ignoring the effects of the rotation of the wrist and the aperture of the grippers. We have not yet performed grasping experiments that use the data available in the Improved Explored PPS Graph, but expect that the general reliability of the grasp action can be improved using the additional features and more accurate feature estimates that can be extracted from the new implementation of the graph model.

The current grasp action is not capable of grasping blocks that are not upright. This appears to be caused by a lack of nodes close enough to the table to have intersections with a block laying down on its surface, as well as a lack of nodes that hold the grippers pointed down at the table for a well-aligned and perpendicular approach. While additional features may be identified to mitigate these issues, it may be the case that nodes would need to be added to the graph to resolve them completely. In order to produce more nodes near the surface of the table, the agent may need to perform motor babbling with less strict rejection criteria. As the graph tends to be sparse near edges of the explored space that are created either by the rejection criteria or the maximum extent of the arm's reach, it may be a better solution to perform motor babbling without the table present so that its surface need not be an edge. Once the table (or any other workspace surface) was introduced to the environment, nodes that fall beneath or inside its surface could be removed from consideration and have all edges to them disconnected. The lack of nodes with an orientation that would allow the hand to grasp the object in a crane-like maneuver can also be addressed by changes to the motor babbling method for building the graph. We hypothesize that since the initial pose for n_0 was approximately parallel to the table's surface and pointed to the right, it is unlikely that the arm will reach a configuration that points the hand downward by adding small random deltas to each joint. The configurations required to point the hand downward are quite distant from those of n_0, and also require each joint angle to be in a relatively specific range. The rejection criteria that keep all node configurations within reduced ranges may further limit the possible paths to arrive at one of these nodes while motor babbling. Instead of constructing a single chain of nodes, we hypothesize that the agent may be better served by constructing a few chains that each begin with a qualitatively different pose (such as pointed to the right for swinging motions or or pointed down for crane-like motions), and then connecting a number of nearest pairs of nodes between each chain in addition to adding edges that satisfy the configuration-space distance heuristic for feasibility. Experiments could be performed to determine if the additional types of nodes would require a larger total to be collected to cover the space equally well, and to determine if grasping or other actions become more reliable.

These improvements to the grasp action may also be complemented by improvements to the place action, so that the object may be placed upright and be in an orientation that allows it to be more easily grasped. Steps toward more precise and stable placements may also advance the agent's capabilities so that it can place in smaller qualitative locations, near specific coordinates, or on another object to create a stack. With enough improvements to either the grasp or place action or both, the agent may practice pick-and-place tasks in long cycles without requiring experimenter intervention, which is currently necessary whenever the object transitions to a state that is too difficult to grasp with the current method.

The graph representation may also be improved by giving the agent the capability to add nodes to the graph after the initial motor babbling is complete. Assuming the image processing system used is capable of accurately identifying the features of the hand in an image with an object present (and potentially adjacent to the hand), one approach would be to add a node whenever the arm is moved to a configuration that has been adjusted according to a local Jacobian estimate. This node should record the actual configuration reached (not the intended configuration, which may differ slightly due to the error tolerance in the built-in Joint Position Control mode) and derive the visual features from new images taken of the hand at this pose, not simply the intended image-space center (which may not be produced due to errors from the local Jacobian estimate). Nodes added in this way would be near the position of a block that was the target of a reach or grasp, which ensures that the new nodes could be near future target positions and may become useful final node candidates. Since objects are at the depth of the table's surface, these nodes are also likely to be added where the graph is sparse and improve density in that region. We also suggest an approach where the agent identifies the sparse regions of the graph and creates one or more new nodes in each. "Holes" in the graph may be identified by examining heat maps of the coverage of the existing nodes' palm masks or depth ranges, or by identifying a position (u, v, d) in image space that has the most distance to the nearest node's palm center. Once the hole or sparse region is identified, the agent can perform a move into it using a local Jacobian adjustment from a nearby node, and record the resulting pose. Once the nodes have been added, the agent will have a more complete model of peripersonal space. The added density in regions that were sparse is also likely to improve the reliability and generality of the agent's actions, as this depends on the availability of a candidate node with reliable features for the current target.

Another opportunity for future work is to extend the usage of the local Jacobian estimates to modify aspects of the full trajectory, rather than to adjust only the final configuration or final move. We have carried out a preliminary reaching experiment where instead of using a graph path trajectory, the agent extracts the center of the target c^t from the current images and estimates the configuration required to reach it based on its displacement from the current nearest node, which is initially the home node n_h, so $\hat{q}^t = (c^t - c^p_h) \cdot \hat{J}^+(n_h)$. Given that the target is generally distant and well outside the neighborhood of n_h, the estimate $\hat{q}^t$ will be inaccurate. However, the estimate is still accurate enough that moving towards $\hat{q}^t$ tends to decrease the distance from the target. Each time during this reach that the arm configuration becomes nearest to that of a different node n_n, the estimated configuration for reaching the target is updated to $\hat{q}^t = (c^t - c^p_n) \cdot \hat{J}^+(n_n)$. If the estimate is ever particularly poor, the motion can still be corrected once a different node is nearest and gives a different estimate. Since the distance $||c^t - c^p_n||$ tends to decrease over time, eventually the estimates are accurate and the reach succeeds with a bump of the target object. When this is recognized, the agent stops the motion of the arm and the reaching trial ends. A demonstration of this approach can be viewed at: https://www.youtube.com/watch?v=MSbbPwWCgd8. While this motion still has some peculiarities, it looks significantly smoother, faster, and more efficient than a reach along a graph path trajectory to a target at the same position, which can be viewed at: https://www.youtube.com/watch?v=3yiEK99bKNE. We have not yet pursued this reaching method further as it is unclear how to retain some advantages of graph path trajectories. In particular, this preliminary new reaching method does not have a mechanism for avoiding obstacles, and it also cannot guarantee a specific well-aligned direction of approach that would be necessary to use the same method for planning a successful grasp trajectory.

We also intend for our work to contribute to conversations in other fields, where work may be continued by experiments in those fields or through collaboration. There are clear points where approaches from reinforcement learning and deep reinforcement learning could be combined with our approach for mutual benefit. Reinforcement learning and deep reinforcement learning approaches could be applied to extend this work in multiple places, and we suggest two here. First, these methods could be used by the agent to select the best control law to use with Joint Velocity Control. At this time, we have demonstrated different behaviors through a grid search of coefficient values carried out at experimenter direction, and found that $k_p = 1.25$ produces motions that appear to be nearly critically damped. A reinforcement learning approach with appropriate rewards for reaching the set point as well as penalties for additional time or total rotation would allow a more precise optimal control law to be determined, and to be determined in a more automated method. Second, (deep) reinforcement learning methods can use the information in the PPS Graph and the learned action policies as an initialization rather than a random initialization, and train more efficiently by searching in this neighborhood rather than the space of all possible motions. Of particular interest would be the use of a reward signal to make the motion trajectories smoother and more efficient, allowing for a transition from early action forms learned in this work to more skillful late action forms. The agent could also become more skilled through a form of apprenticeship learning[1] where the required demonstration could be performed using the agent's previously learned early action policies. While these learned policies are unlikely to produce a fine-tuned "expert" demonstration, they are capable of serving an analogous role of directing the attention of the reinforcement learner to a neighborhood of the solution space that reliably achieves the goal. Smoother and more controlled motions may also be important to making actions that require high precision, such as grasping, significantly more reliable.

Our learning model and implementation on an embodied robotic agent also allow the expression and testing of hypotheses of interest to developmental psychologists. We have already shown that our model produces behaviors that are consistent with those observed in typical infants, and future collaboration could help to determine if the cause of the behavior in our model is a plausible cause in infants. Additional biases, constraints, and capabilities could also be added to the PPS Graph Representation to support the testing of additional theories, and experiments with the robot can provide faster and more transparent results than those that can be obtained directly from the observation of infants. After obtaining results with our model, this will suggest a focus for experiments with infants, which would provide important support for any theories generated beyond the consistencies in behavior that can be documented with the robot.

8.3 Conclusion

We have presented a learning algorithm where the observation of unusual events leads to the definition of actions to repeat them, and where exploration reveals features that can be used to make reliable plans for these actions that can be executed by an embodied robotic agent. In an amazing phenomenon, human infants and other natural agents can be observed to learn to successfully perform a multitude of actions that they are not innately capable of performing. This provokes a fundamental question of how this learning could be possible. Our implementation of a computational model illuminates a sufficient set of knowledge that can be gained to make these foundational actions reliable and a plausible way to represent this information. Our model is consistent with natural agents as it initially produces infant-like early actions and allows for these actions to be refined with additional practice and learning. For the set of actions considered in this work, our agent follows a learning process that demonstrates learning these actions from experience, as infants do, is feasible.

BIBLIOGRAPHY

[1] P. Abbeel, A. Coates, and A. Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[2] K. E. Adolph and S. E. Berger. *Physical and motor development*. Erlbaum, 5th edition, 2005.

[3] G. Baldassarre and M. Mirolli, editors. *Intrinsically Motivated Learning in Natural and Artificial Systems*. Springer, 2013.

[4] Bertenthal and Bennett. Origins and early development of perception, action, and representation. *Annual review of psychology*, 47:431–459, 02 1996.

[5] N. E. Berthier. The syntax of human infant reaching. In *Unifying Themes in Complex Systems: Proc. 8th Int. Conf. on Complex Systems*, volume VIII of *New England Complex Systems Institute Series on Complexity*, pages 1477–1487. NECSI Knowledge Press, 2011.

[6] N. E. Berthier and R. Keen. Development of reaching in infancy. *Experimental Brain Research*, 169:507–518, 2006. doi:10.1007/s00221-005-0169-9.

[7] N. E. Berthier, M. T. Rosenstein, and A. G. Barto. Approximate optimal control as a model for motor learning. *Psychological Review*, 112(2):329–346, 2005.

[8] A. J. Bremner, N. P. Holmes, and C. Spence. Infants lost in (peripersonal) space. *Trends in Cognitive Science*, 12(8):298–305, 2008. doi: 10.1016/j.tics.2008.05.003.

[9] D. Caligiore, D. Parisi, and G. Baldassarre. Integrating reinforcement learning, equilibrium points, and minimum variance to understand the development of reaching: A computational model. *Psychological Review*, 121(3):389–421, 2014. DOI:10.1037/a0037016.

[10] E. Chinellato, M. Antonelli, B. J. Grzyb, and A. P. del Pobil. Implicit sensorimotor mapping of the peripersonal space by gazing and reaching. *IEEE Trans. on Autonomous Mental Development*, 3(1):43–53, 2011.

[11] R. K. Clifton, D. W. Muir, D. H. Ashmead, and M. G. Clarkson. Is visually guided reaching in early infancy a myth? *Child Development*, 64(4):1099–1110, 1993.

[12] R. K. Clifton, E. E. Perris, and D. D. McCall. Does reaching in the dark for unseen objects reflect representation in infants? *Infant Behavior & Development*, 22(3):297–302, 1999.

[13] R. K. Clifton, P. Rochat, D. Robin, and N. E. Bertheir. Multimodal perception in the control of infant reaching. *Journal of Experimental Psychology: Human Perception and Performance*, 20(4):876–886, 1994.

[14] D. Corbetta, S. L. Thurman, R. F. Wiener, Y. Guan, and J. L. Williams. Mapping the feel of the arm with the sight of the object: on the embodied origins of infant reaching. *Frontiers in Psychology*, 5(576), June 2014. http://dx.doi.org/10.3389/fpsyg.2014.00576.

[15] Y. Futagi, Y. Toribe, and Y. Suzuki. The grasp reflex and Moro reflex in infants: Hierarchy of primitive reflex responses. *International Journal of Pediatrics*, 2012(191562), 2012. doi:10.1155/2012/191562.

[16] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[17] M. Hersch, E. Sauser, and A. Billard. Online learning of the body schema. *Int. J. Humanoid Robotics*, 5(2):161–181, 2008.

[18] M. Hoffmann, L. K. Chinn, E. Somogyi, T. Heed, J. Fagard, J. J. Lockman, and J. K. O'Regan. Development of reaching to the body in early infancy: From experiments to robotic models. In *IEEE Int. Conf. Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2017.

[19] M. Hülse, S. McBride, J. Law, and M. Lee. Integration of active vision and reaching from a developmental robotics perspective. *IEEE Transactions on Autonomous Mental Development*, 2(4):355–367, 2010.

[20] L. Jamone, M. Bradao, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi. Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching. *Robotics and Autonomous Systems*, 62(4):556–567, 2014.

[21] L. Jamone, L. Natale, F. Nori, G. Metta, and G. Sandini. Autonomous online learning of reaching behavior in a humanoid robot. *Int. J. Humanoid Robotics*, 9(3):1250017, 2012.

[22] J. Juett and B. Kuipers. Learning to reach by building a representation of peri-personal space. In *IEEE/RSJ Int. Conf. Humanoid Robots*, 2016.

[23] J. Juett and B. Kuipers. Learning to grasp by extending the peri-personal space graph. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[24] J. Juett and B. Kuipers. Learning and acting in peripersonal space: moving, reaching, and grasping. *Frontiers in Neurorobotics*, 13, 2019.

[25] L. E. Kavraki, P. Svestka, J. . Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[26] Suresh Kumar, Alexandros Giagkos, Patricia Shaw, Raphaël Braud, Mark Lee, and Qiang Shen. Discovering schema-based action sequences through play in situated humanoid robots. *IEEE Transactions on Cognitive and Developmental Systems*, 2021.

[27] Suresh Kumar, Patricia Shaw, Alexandros Giagkos, Raphäel Braud, Mark Lee, and Qiang Shen. Developing hierarchical schemas and building schema chains through practice play behavior. *Frontiers in Neurorobotics*, 12:33, 2018.

[28] J. Law, P. Shaw, K. Earland, M. Sheldon, and M. Lee. A psychology based approach for longitudinal development in cognitive robotics. *Frontiers in Neurorobotics*, 8(1), 2014. doi: 10.3389/fnbot2014.00001.

[29] J. Law, P. Shaw, M. Lee, and M. Sheldon. From saccades to grasping: A model of coordinated reaching through simulated development on a humanoid robot. *IEEE Trans. on Autonomous Mental Development*, 6(2):93–109, 2014.

[30] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *International Journal of Robotics Research*, 37(4–5):xx–yy, 2018.

[31] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Technical Report arXiv:1603.02199v4 [cs.LG], arXiv, 2016.

[32] Dingsheng Luo, Fan Hu, Tao Zhang, Yian Deng, and Xihong Wu. How does a robot develop its reaching ability like human infants do? *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):795–809, 2018.

[33] J. Modayil and B. Kuipers. The initial development of object knowledge by a learning robot. *Robotics and Autonomous Systems*, 56:879–890, 2008.

[34] J. Mugan and B. Kuipers. Autonomous learning of high-level states and actions in continuous environments. *IEEE Trans. Autonomous Mental Development*, 4(1):70–86, 2012.

[35] E. Oztop, N. S. Bradley, and M. A. Arbib. Infant grasp learning: a computational model. *Experimental Brain Research*, 158(4):480–503, 2004.

[36] Jean Piaget. *The Origins of Intelligence in Children*. Norton, 1952.

[37] D. M. Pierce and B. J. Kuipers. Map learning with uninterpreted sensors and effectors. *Artificial Intelligence*, 92:169–227, 1997.

[38] L. Pinto and A. Gupta. Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2016. arXiv:1509.06825v1 [cs.LG].

[39] M. Seetha Ramaiah, Amitabha Mukerjee, Arindam Cakraborty, and Sadbodh Sharma. Visual generalized coordinates. *arXiv preprint arXiv:1509.05636*, 2015.

[40] A. Roncone, M. Hoffmann, U. Pattacini, L. Fadiga, and G. Metta. Peripersonal space and margin of safety around the body: Learning visuo-tactile associations in a humanoid robot with artificial skin. *PLoS ONE*, 11(10):ee0163713, 2016. https://doi.org/10.1371/journal.pone.0163713.

[41] P. Savastano and S. Nolfi. A robotic model of reaching and grasping development. *IEEE Trans. on Autonomous Mental Development*, 5(4):326–336, 2013.

[42] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990-2010). *IEEE Trans on Autonomous Mental Development*, 2(3):230–247, 2011.

[43] A. Serino. Peripersonal space (pps) as a multisensory interface between the individual and the environment, defining the space of the self. *Neuroscience and Behavioral Reviews*, 99:138–159, 2019.

[44] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglu, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 262(6419):1140–1144, 2018.

[45] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[46] M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2006.

[47] J. Sturm, C. Plagemann, and W. Burgard. Unsupervised body scheme learning through self-perception. In *IEEE Int. Conf. Robotics and Automation (ICRA)*, 2008.

[48] I. A. Sucan and S. Chitta. Moveit! *[Online] Available http://moveit.ros.org*, Accessed 3-25-2016.

[49] E. Thelen, D. Corbetta, K. Kamm, J. P. Spencer, K. Schneider, and R. F. Zernicke. The transition to reaching: mapping intention and intrinsic dynamics. *Child Development*, 64(4):1058–1098, 1993.

[50] B. L. Thomas, J. M. Karl, and I. Q. Whishaw. Independent development of the Reach and the Grasp in spontaneous self-touching by human infants in the first 6 months. *Frontiers in Psychology*, 5(1526):1–11, 2015. doi:10.3399/fpsyg.2014.01526.

[51] E. Ugur, Y. Nagai, E. Sahin, and E. Oztop. Staged development of robot skills: behavior formation, affordance learning and imitation with motionese. *IEEE Trans. on Autonomous Mental Development*, 7(2):119–139, 2015.

[52] A. L. van der Meer. Keeping the arm in the limelight: Advanced visual control of arm movements in neonates. *Eur. J. Paediatric Neurology*, 4:103–108, 1997.

[53] A. L. H. van der Meer, F. R. van der Weel, and D. N. Lee. The functional significance of arm movements in neonates. *Science*, 267:693–695, 1995.

[54] C. von Hofsten. Eye-hand coordination in the newborn. *Developmental Psychology*, 18(3):450–461, 1982.

[55] C. von Hofsten. Developmental changes in the organization of pre-reaching movements. *Developmental Psychology*, 20(3):378–388, 1984.

[56] C. von Hofsten. Structuring of early reaching movements: A longitudinal study. *J. Motor Behavior*, 23(4):280–292, 1991.