The Dissertation Committee for Ajinkya Jain
certifies that this is the approved version of the following dissertation:

Learning and Leveraging Kinematics for Robot Motion Planning Under Uncertainty

Committee:

Scott David Niekum, Co-Supervisor

Ashish Deshpande, Co-Supervisor

Raul G. Longoria

Dongmei Chen

Farshid Alambeigi

Learning and Leveraging Kinematics for Robot Motion Planning Under Uncertainty

by

Ajinkya Jain

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
August 2021

Dedicated to my parents, Kiran and Arun,

on whose shoulders my dreams soared

Acknowledgments

This journey would not have been possible without the help and support from many people. First and foremost, I would like to thank my advisor Scott Niekum for his advice and support throughout my PhD. I am thankful to him for giving me the flexibility and encouragement to pursue my interests while providing much-needed help and guidance. I will forever be indebted to him for his support and patience during my initial years, when I struggled to find my research direction. I am highly grateful to him for his excellent mentorship that has helped me to grow as an independent thinker and taught me that it is always possible to communicate even the most complex, abstract ideas in simple words.

I would also like to thank my co-advisor Ashish Deshpande for his constant support, encouragement, and guidance throughout my PhD. He motivated me to look at the research questions through different lenses ranging from the control theory and mechanism design to machine learning, which helped me tremendously in growing as a researcher. I am also deeply grateful to my committee members, Raul Longoria, Dongmei Chen, and Farshid Alambeigi, for providing invaluable feedback on my dissertation.

During my time at UT, I had been fortunate to get to know and work with many other graduate students. I would like to particularly thank the members of the Personal Autonomous Robotics Lab, Daniel Brown, Caleb Chuck, Yuchen Cui, Wonjoon Goo, Prasoon Goyal, Akanksha Saran, Jordan Schneider, and Christina Yuan, for their helpful discussions and advice throughout this journey. I am especially grateful to Rudolf Lioutikov for bearing with me while I bounced random ideas off him, supporting and guiding me while refining my thought process, motivating me whenever I felt the need, and most of all, being there with his great smile to cheer me up whenever I was struggling. I am also thankful to Stephen Giguere for going above and beyond to help me with his much-needed support and advice, especially towards the end of my PhD. I am also grateful to the members of the robotics community at UT and Georgia Tech, especially Adam Allevato, Vivian Chu, Taylor Kessler Faulkner, Tesca Fitzgerald, Alex Gutierrez, Justin W Hart, Steven Jens Jorgensen, Donghyun Kim, Mincheol Kim, Ian Lenz, Elaine Short, Max Svetlik, Ye Zhao, and many others I am sure I am forgetting, for their friendship, support, and many great discussions.

I have been fortunate enough to have some of the most amazing people as friends who helped me with whatever I needed, whenever I needed it. Thanks, Rituparna Samanta, Sumit Sinha, Chola Bhargava Dandamudi, Surya Dhulipala, Sundar Varadan, Mihir Mehta, Subramanian Sankaranarayanan, and Amrutha Sridhar, for the countless fun moments we had together. I would also like to thank all my friends that I had been fortunate to gain during my undergraduate years, especially thanks to Mohit Agarwal, Rohit Saini, Arush Sinhal, Mohd. Dawood, Mridul Verma, Avinash Shrivastava, Amber Srivastava, Anuj Agrawal, and Yashesh Dhebar. Our shared fun has always been my bedrock throughout this journey. I also want to thank Tushti Shah, without whom I can't imagine what this process would have been. Meeting her was one of the most delightful things that happened to me during my PhD. In her, I found not only an amazing partner but my best friend on whom I can lean on whenever I stumbled. Thanks, Tushti, for always believing in me and being the reason for my smile. Finally and most importantly, I would like to thank my parents, Kiran and Arun Jain, my brother, Aakansh Jain, and my family for being there with me at every step of my life watching over me. Their constant support and motivation have been my north star, guiding me forward always even when the sky may be dark and cloudy.

AJINKYA JAIN

The University of Texas at Austin
July 2021

Learning and Leveraging Kinematics for Robot Motion Planning Under Uncertainty

by

Ajinkya Jain, Ph.D.
The University of Texas at Austin, 2021

Supervisors: Scott David Niekum
Ashish Deshpande

Service robots that can assist humans in performing day-to-day tasks will need to be general-purpose robots that can perform a wide array of tasks without much supervision from end-users. As they will be operating in unstructured and ever-changing human environments, they will need to be capable of adapting to their work environments quickly and learning to perform novel tasks within a few trials. However, current robots fall short of these requirements as they are generally highly specialized, can only perform fixed, predefined tasks reliably, and need to operate in controlled environments. One of the main reasons behind this big gap is that the current robots require complete and accurate information about their surroundings to function effectively, whereas, in human environments, robots will only have access to limited information about their tasks and environments. With incomplete information about its surroundings, a robot using pre-programmed or pre-learned motion policies will fail to adapt to the novel situations encountered during operation and fall

short in completing its tasks. Online motion generation methods that do not reason about the lack of information will not suffice either, as the developed policies may be unreliable under incomplete information. Reasoning about the lack of information becomes critical for manipulation tasks a service robot would have to perform. These tasks will often require interacting with multiple objects that make or break contacts during the task. A contact between objects can significantly alter their subsequent motion and lead to sudden transitions in their dynamics. Under these sudden transitions, even minor errors in estimating object poses can cause drastic deviations from the robot's initial motion plan for the task and lead the robot to failure in completing the tasks. Hence, service robots need methods that generate motion policies for manipulation tasks efficiently while accounting for the uncertainty due to incomplete or partial information.

Partially Observable Markov Decision Processes (POMDPs) is one such mathematical framework that can model and plan for tasks where the agent lacks complete information about the task. However, POMDPs incur exponentially increasing computational costs with planning time horizon, which restricts the current POMDP-based planning methods to problems having short time horizons. Another challenge for planning-based approaches is that they require a state transition function for the world they are operating in to develop motion plans, which may not always be available to the robot. In control theory terms, a state transition function for the world is analogous to its system plant. In this dissertation, we propose to address these challenges by developing methods that can learn state transition functions for robot manipulation tasks directly from observations and later use them to generate long-horizon motion plans to complete the task under uncertainty.

We first model the world state transition functions for robot manipulation tasks

involving sudden transitions, such as due to contacts, using hybrid models and develop a novel hierarchical POMDP-planner that leverages the representational power of hybrid models to develop motion plans for long-horizon tasks under uncertainty. Next, we address the requirement of planning-based methods to have access to world state transition functions. We introduce three novel methods for learning kinematic models for articulated objects directly from observations and present an algorithm to construct the state transition functions from the learned kinematics models for manipulating these objects. We focus on learning models for articulated objects as they form one of the biggest sets of household objects that service robots will frequently interact with. The first method, MICAH, focuses on learning kinematic models for articulated objects that exhibit configuration-dependent articulation properties, such as a refrigerator door that stays closed magnetically, from unsegmented sequences of observations of object part poses. Next, we introduce ScrewNet, which removes the requirement of object pose estimation of MICAH and learns articulation properties of objects directly from raw sensory data available to the robot (depth images) without knowing their articulation model category a priori. Extending it further, we introduce DUST-net, which learns distributions over articulation model parameters for objects indicating the network's confidence over the estimated parameters directly from raw depth images. Combining these methods, in this dissertation, we introduce a unified framework that can enable a robot to learn state transition functions for manipulation tasks from observations and later use them to develop long-horizon plans even under uncertainty.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the longstanding goals of robotics is to build service robots that can assist humans in performing day-to-day tasks. Service robots aim to be general-purpose robots that can perform a wide array of tasks without needing much supervision from the end-users. As service robots will be helping humans in homes, schools, hospitals, and workplaces, they will be required to adapt to their work environments quickly and learn to perform novel tasks within a few trials. However, current robots fall short of these requirements. They are generally highly specialized, can only perform fixed, predefined tasks reliably, and need to operate in controlled environments. One of the main reasons behind this big gap is that the current robots require complete and accurate information about their surroundings to function effectively, whereas, in unstructured human environments, robots will only have access to limited information about their tasks and work environments [46, 121]. Moreover, they will be required to perform their tasks with high reliability and safety guarantees, as safety is of utmost importance in human environments.

With the limited available information, a robot using pre-programmed or pre-learned motion policies might fail to adapt to the novel situations encountered during operation and fall short in completing its tasks. Online motion generation methods that do not reason about the lack of information will not suffice either, as the developed policies may be unreliable under incomplete information. Accounting for the lack of information becomes critical for the manipulation tasks service robots would have to perform. These tasks will often require interacting with multiple objects that make or break contacts during the task. Changes in the contact state of an object can significantly alter its subsequent motion and lead to sudden transitions in its dynamics. Under these sudden transitions, even minor errors in estimating the object pose can cause substantial deviations from the robot's initial motion plan for the task and lead the robot to failure in completing the task. For example, consider the robot manipulation task of placing a glass vase on a table. As soon as the vase and the tabletop make contact, a previously optimal downward force on the vase will suddenly result in a large reactive normal force with no downward displacement. If the robot overestimates the distance between the vase and the table even slightly, the chances are that the robot will fail to reduce its force on the vase in time and may end up breaking the vase instead. Hence, service robots need methods that generate motion policies for manipulation tasks efficiently while accounting for the uncertainty due to incomplete or partial information to operate safely in human environments.

Partially Observable Markov Decision Processes (POMDPs) [62] is a mathematical framework to model and plan for tasks in which the decision-making agent does not have complete information about its task or the environment. Using POMDP-based planning methods, a robot can develop motion plans to take information gathering actions while completing the task objectives. However, obtaining an exact solution of a POMDP is intractable as it results in solving a decision problem that is PSPACE-complete [62, 80, 97, 103, 145] and intractable to solve. Numerous methods have been presented in the literature to solve POMDPs approximately [2, 3, 48, 51, 68, 69, 94, 119, 122, 144]. However, the majority of existing approximate methods incur exponentially increasing computational costs with planning horizons, which restrict them to problems having short time horizons.

Another challenge that arises while using planning-based methods is that they need state transition functions for the world they are operating in to develop motion plans. The world state transition functions are analogous to the system plants from the control theory. While such models can be provided to the robot by an expert, it is challenging to pre-program models of all tasks that a service robot will need to perform in uncontrolled human environments. Hence, a service robot needs methods for learning such functions directly from observations. Household appliances constitute one of the biggest sets of objects that service robots will need to interact with frequently. Predominantly, household appliances, such as refrigerators, microwaves, and drawers, are articulated objects and consist of multiple functional parts connected by mechanical joints such as hinges and sliding joints. For articulated objects, the state transition functions are governed primarily by the articulation properties (or joint properties) of the objects, including the location of the joint, its axis of motion, type (hinge or sliding), and the available range of motion. A robot capable of learning the articulation properties for objects directly from observations can construct state transition functions for them without requiring expert supervision [56, 58], which it can later use to manipulate these objects. Hence, service robots need methods that can learn articulation properties for objects directly from observations.

Addressing these challenges, in this dissertation, we present methods for learning state transition functions for manipulation tasks from observations and performing POMDP-based motion planning for long-horizon robot manipulation tasks. Specifically, in this thesis, we wish to answer the question:

How can a robot learn state transition functions for complex manipulation tasks directly from observations and use them to perform long-horizon planning under uncertainty?

This dissertation addresses this question in four stages:

1. We first introduce a hierarchical POMDP planner, that under incomplete information about the world, can leverage object kinematics and interactions to perform long-horizon manipulation tasks robustly. However, the POMDP planner needs the state transition functions for tasks to generate such motion plans, which may not be available for a service robot performing novel tasks.

2. To address this requirement, we next provide a noise-robust method for learning planning-compatible state transition functions for articulated objects that exhibit configuration dependent articulation properties from observations. However, it requires 6D object poses as input, which may not be readily available for all scenarios.

3. Addressing this challenge, we next introduce a method for learning articulation properties for objects directly from raw sensory data available to the robot, such as depth images. Combining it with the previous methods, we obtain a framework that learns state transition functions for manipulating household (articulated) objects directly from raw observations.

4. Finally, we extend the framework to predict both the articulation model parameters for objects and the network's confidence over them directly from raw depth images. This results in a complete framework that given raw sensory information about the objects

that the robot is interacting with, first learns state-transition functions for them with model uncertainty estimates, and later uses them to generate uncertainty-aware motion plans to manipulate them.

In the remainder of this chapter, we will discuss these stages in more detail and highlight the specific contributions of this dissertation.

We first focus on investigating the question that if the world state transition function for a task is known, can a hierarchical POMDP-based planning approach leverage it to develop long-horizon motion plans for the task under incomplete information? In particular, we focus on the problem of finding motion policies for manipulation tasks involving contacts in the presence of partial or noisy feedback. Current methods to calculate manipulation policies for tasks involving contacts between objects either avoid them altogether by working in isolation or assume perfect knowledge about the world and plan explicitly for contacts [28, 96, 113, 114]. Alternatively, they choose to ignore these nonlinearities while planning motion policies and rely on impedance/admittance controllers to handle them during execution [43, 47, 52]. While these methods work well in controlled industrial environments with near-perfect sensors, they can fail in the presence of noisy feedback that plagues service robots [46]. Under changing dynamics, even minor state estimation errors can lead to significant deviations from the plan and cause the robot to fail in completing the task. Recently, data-driven methods have garnered much attention as they can learn manipulation policies that handle contacts implicitly [4, 11, 41, 67, 88, 99, 110, 116]. However, these methods often require a large amount of data to learn such policies, which may be difficult to obtain for real-world robot manipulation tasks.

In this work, we model the nonlinear dynamics of robot motion planning tasks involving contacts using hybrid models composed of a discrete set of simpler local dynamics models of which only one is active at any given time (e.g., a change in dynamics due to contact) [33, 79]. We introduce a novel hierarchical POMDP-based planner, the POMDP-HD planner [54, 55] that can leverage the representational power of hybrid models to reduce its uncertainty over states. The proposed planner solves the POMDP problem by dividing it into two levels: at the higher level, discrete state plans are generated to find a sequence of local models that should be visited during the task, and at the lower level, these discrete state plans are converted into cost-optimized continuous state belief-space plans.

The hierarchical structure of the POMDP-HD planner garners crucial advantages over direct planning-based approaches. The hierarchical structure helps the planner to decompose long-horizon POMDP planning problems into multiple smaller segments that can be sequenced to find a complete solution with significantly less effort. This decomposition alleviates the constraint that POMDP-based planners can find tractable solutions only when the planning problems have short time horizons. This approach is similar to how humans leverage contacts and transitions in dynamics to improvise and complete tasks under imperfect information. For example, while assembling an Ikea table, if direct visual feedback is not available for inserting a table leg on the underside of the table, a human can try to wiggle the leg around to locate the hole in order to complete the task.

Another significant benefit the POMDP-HD planner offers that it can choose to leverage a specific local dynamics model to enhance the effectiveness of the generated plans. For example, if the k-th local dynamics model restricts the allowed motion to be within a subspace of the robot's configuration space, visiting the states governed by the model can help the robot to reduce its uncertainty over system state along the dimensions orthogonal to the allowed motion vector. E.g., due to the presence of a wall, the k-th local model allows displacements only in directions that point either away or along the wall. If the robot deliberately plans to make contact with the wall and move along it, the robot can localize itself better in the direction orthogonal to the wall plane by observing its subsequent motion. This indirect feedback for uncertainty reduction is critical for tasks in which observations are highly noisy or even entirely unavailable (for example, due to occlusions).

Next, we address learning state transition functions for tasks that can be used to perform long-horizon planning under uncertainty. We focus on learning state transition functions for tasks involving sudden changes in the dynamics, such as those involving making/breaking contacts with objects and manipulating objects exhibiting conditional dynamics, such as a stapler that intrinsically changes its articulation state (e.g. rigid vs. rotational) based on the relative angle between its arms. Such transitions in the dynamics are often viewed as inconvenient discontinuities that make manipulation difficult. However, when these transitions are well understood, they can be leveraged to reduce uncertainty or aid manipulation, as noted above. Current model-free reinforcement learning approaches [4, 11, 41, 67, 88, 99, 110, 116] can learn to cope with hybrid dynamics implicitly but require large amounts of data to do so and may still face representational issues near discontinuities. They scale poorly as the problem complexity grows and do not transfer well to significantly different problems. On the other hand, hierarchical POMDP planning-based methods [13, 55, 67, 142] can represent and reason about hybrid dynamics and uncertainty directly. They scale well by decomposing the planning problems into smaller subproblems and work well on novel tasks. However, they typically rely on precise hand-specified models and task decompositions. To bridge this gap, we introduce Model Inference Conditioned on Actions for Hierarchical Planning (MICAH), which enables hierarchical POMDP planning-based methods, such as the POMDP-HD planner, to perform novel manipulation tasks given noisy observations. MICAH infers hybrid automata for articulated objects with configuration-dependent dynamics from unsegmented sequences of observed poses of object parts. These automata can then be used for developing state transition functions for novel manipulation tasks involving these objects, which, in turn, can be used to perform motion planning under uncertainty for the tasks.

MICAH consists of two parts: (1) a novel action-conditional inference algorithm called Act-CHAMP for kinematic model estimation and changepoint detection from unsegmented data, and (2) an algorithm to construct hybrid automata for objects using the detected changepoints and estimated local models from Act-CHAMP. Due to action-conditional inference, MICAH is more robust to noise and less vulnerable to several modes of failure than prior model inference approaches [63, 64, 85–87, 100, 111, 130, 131]. These prior approaches assume that the visual pose observations alone provide sufficient information for model estimation, which does not hold for many scenarios and can lead to poor performance. For example, an observation-only approach cannot distinguish between observations obtained by applying force against a rigid object and taking no action at all on a free body, estimating that the model is rigid in both cases.

MICAH provides a noise-robust method for constructing state transition functions for manipulating articulated objects from visual observations with high accuracy. However, it requires time series of observations of relative poses between the object parts as input, which is often difficult to obtain in unstructured human environments. Recent work on learning articulation parameters for objects from visual observations has explored estimating these parameters directly from raw data, such as depth images [1, 72, 75] or PointClouds [147, 153], using deep neural networks. However, current methods [1, 72, 147, 153] require knowing the object articulation model type a priori, which may not be available to a service robot interacting with novel objects. Alleviating this requirement, we introduce ScrewNet [58], a method that estimates an object's articulation model directly from depth images without requiring a priori knowledge of the object articulation model category. It uses screw theory to unify the representation of different articulation types and perform category-independent articulation model estimation. A unified representation also helps it to be more data-efficient than prior methods [1, 72]. Compared to other approaches, ScrewNet can successfully estimate the articulation models and their parameters for novel objects across articulation model categories with better-on-average accuracy using only half the training data. The estimated articulation model for an object can later be used with the hybrid automata construction algorithm of MICAH to develop a hybrid automaton representing the state transition fuction for manipulating the object.

One limitation of ScrewNet [58] and related methods [1, 72, 75, 147, 153] is that they can only predict point estimates for the articulation model parameters for objects. For constructing POMDP planning-compatible state transition functions for articulated objects, uncertainty over the estimated articulation model parameters also needs to be accounted for. Reasoning about the uncertainty in the estimated model parameters offers additional benefits such as using active learning methods for improving model parameter estimates [23] and directly learning behavior policies that provide safety assurances [137]. Motivated by these advantages, we introduce a method, DUST-net [57], that efficiently learns distributions over articulation models directly from depth images without requiring to know articulation model categories a priori.

We first introduce a novel representation for distributions over rigid body transformations and articulation model parameters based on screw theory [121], von Mises-Fisher distributions [84] and Stiefel manifolds [22]. Combining these concepts allows for an efficient, mathematically sound representation that inherently satisfies several constraints that rigid body transformations and articulations must adhere to. This distributional representation helps DUST-net [57] to perform articulation model estimation for novel objects across articulation model categories with better accuracy than state-of-the-art methods [1, 58] while also providing model uncertainties. The representation also helps DUST-net to be more computationally and data-efficient than other state-of-the-art methods [1, 72], as it uses a single network to estimate model parameters for all common articulation models, unlike other methods that require a separate network for each articulation model category [1, 72, 147, 153]. Empirically, DUST-net outperforms other methods even when trained using only half the training data in comparison. Additionally, the distributional learning setting also yields DUST-net more robustness to outliers and noise, making it an ideal candidate for learning articulation models for objects directly from raw depth images. The learned distributions using DUST-net can later be combined with the hybrid automata construction algorithm of MICAH to construct a state transition function with model uncertainty estimates for manipulating the object.

With the presented methods in this dissertation, namely the POMDP-HD planner [55], MICAH [56], ScrewNet [58], and DUST-net [57], we introduce a unified framework that addresses the research question initially posed in this dissertation. Using the presented framework, a robot can learn state transition functions for complex articulated objects involved in a manipulation task directly from raw observations and use them to perform long-horizon planning to complete the task even under uncertainty.

1.1 Contributions

In particular, this dissertation presents the following contributions:

1. A hierarchical POMDP planner that can leverage hybrid task dynamics to perform long-horizon motion planning under uncertainty (Chapter 4).
2. An action-conditional model inference algorithm, Act-CHAMP, for estimating kinematic models and possible transitions points in the governing model for articulated objects from unsegmented data (Chapter 5).
3. An algorithm to construct hybrid automata for articulated objects using the detected changepoints and estimated local models from Act-CHAMP (Chapter 5).
4. A method for learning articulation models for objects directly from raw depth images without requiring to know their articulation model category a priori (Chapter 6).
5. A novel representation for distributions over rigid body transformations and articulation model parameters based on screw theory, von Mises-Fisher distributions, and Stiefel manifolds (Chapter 7).
6. An approach that efficiently learns distributions over articulation model parameters for objects directly from depth images without needing to know their articulation model category a priori (Chapter 7).

Chapter 2

Background and Related Work

This chapter surveys work related to the main threads of this thesis, intending to provide a background for the presented work among the existing literature on robotics research. Section 2.1 focuses on the POMDP-based planning methods for robot motion planning under uncertainty. Section 2.2 surveys methods for learning kinematic models of articulated bodies from observations.

2.1 POMDP-based Planning for Robot Motion Planning Under Uncertainty

Robot motion planning methods aim to find a cost-optimal path connecting an initial state of the robot to a target state while avoiding obstacles given a complete and correct model of the world dynamics and a cost function specifying preferences over paths [70]. However, more often than not, the world dynamics model is either incomplete or contains inaccuracies, resulting in uncertainty over the effectiveness of the generated plan from a motion planner that assumes otherwise. Partially observable Markov decision processes (POMDPs) [50, 62, 94] is a framework that reasons about possible incompleteness or inaccuracies in the world dynamics model and develops cost-optimal motion plans that are effective even under uncertainty. In Section 3.2 we discuss them in detail. In this section, we briefly describe different POMDP-based motion planning methods for robot motion planning under uncertainty.

Broadly, POMDP based methods can be divided into two categories based on whether their state, action and observation spaces are *discrete* or *continuous*. Discrete space POMDP solvers, in general, either approximate the value function using point-based methods [69, 112, 119, 123, 124] or use Monte-Carlo sampling in the belief space [6, 65, 68, 74, 101, 122, 125] to make the POMDP problem tractable. Continuous space POMDP solvers often approximate the belief over states as a distribution having finite parameters (typically Gaussian) and either solve the problem analytically using gradients [51, 81, 144] or use random sampling in the belief space [2, 48]. Other approaches have also extended point-based methods to continuous domains [119].

Discrete space POMDP solvers have been shown to be able to successfully plan for large discrete space domains. However, continuous space domains are infinite-dimensional, and discrete space solvers often fail to find feasible solutions for planning horizons longer than a few steps [68]. Among continuous space POMDP solvers, Agha-Mohammadi et al. [2] and Hollinger and Sukhatme [48] have proposed sampling based methods that can find effective solutions even in complex domains. However, most sampling based methods suffer from the problem of obtaining sub-optimal solutions and can only be probabilistically optimal at best [27]. Gradient-based POMDP solvers [51, 81, 144] form another class of very powerful POMDP solvers which can find locally optimal solutions, but in the context of manipulation planning, sudden changes in dynamics due to contacts result in non-finite gradients at the transition points and restrict the applicability of such methods.

2.1.1 POMDP planning in domains with hybrid dynamics

POMDP solvers for domains governed by hybrid dynamics models, such as the one proposed in this work, have been previously discussed by Brunskill et al. [13], Sreenath et al. [127] and Agha-mohammadi et al. [3]. Recall that a hybrid dynamics model consists of multiple local dynamics models of which only one is active at any given time. In the most closely related work to ours, Brunskill et al. [13] proposed a point-based POMDP planning algorithm, SM-POMDP planner, for solving continuous-state POMDPs based on the hybrid system dynamics. They approximated the complex nonlinear system dynamics using a hybrid multi-modal dynamics model with continuous state-dependent discrete mode switching conditions. However, unlike our POMDP planner (Chapter 4), SM-POMDP planner plans only in the continuous domain and the discrete states are obtained "passively" using the switching conditions. While this approach can be used to find feasible motion plans, it is not leveraging some of the major natural advantages of the hybrid dynamics representation such as shorter planning horizons and a structured way to leverage dynamics for state uncertainty reduction.

Sreenath et al. [127] discussed the problem of bipedal walking on a varying terrain by formulating it as a POMDP problem defined on a continuous-time hybrid system. They proposed a bi-level POMDP controller to track the transitions in the terrain as a set of discrete states and were able to show stable bipedal walking in simulated domains. However, this is a passive approach as well, as it uses hybrid dynamics only to capture the transitions in the terrain and not to simplify the POMDP problem. Agha-mohammadi et al. [3] discussed a POMDP solver with hybrid states to solve health-aware stochastic motion planning problem for quadrotors, however, the proposed solution is restricted only to the domains in which the discrete and continuous states evolve independently.

Pineau et al. [112] and Toussaint et al. [142] have also previously proposed hierarchical POMDP planners. The planner developed by Pineau et al. [112] leverages a human-designed task hierarchy to reduce problem complexity, while Toussaint et al. [142] emphasizes automatic discovery of hierarchy in state space using a dynamic Bayesian network. Although such approaches can work well for some robot control tasks, we believe that a more natural hierarchy of subtasks emerges automatically if a hybrid dynamics model is used to represent tasks with nonlinear dynamics, such as robot manipulation tasks involving contacts.

2.1.2 Further approaches

As hybrid dynamics models are very effective in modeling nonlinearities that are due to sudden transitions in the dynamics, a natural application for the proposed POMDP solver is contact-rich robotic manipulation. One of the current approaches for solving the robot manipulation planning problem is to search for an optimal sequence of parameterized manipulation actions or primitives to perform the task [26, 66]. Kroemer et al. [66] have proposed to represent primitives for different phases (modes) of a multi-phase manipulation task using *dynamic movement primitives (DMPs)* and learn a library of such manipulation skills which can be sequenced to perform a task. Unfortunately, a lack of a task dynamics model prevents these methods from generalizing to novel manipulation tasks, e.g. having different cost functions, even if it involves the same objects.

More recently, researchers [31, 41, 67, 71, 83, 99] have explored using deep learning techniques to develop end-to-end control policies directly from vision; however, under sparse availability of training data (especially in robotics), these approaches tend to fail to develop generalized control policies for all system states or initial conditions.

2.2 Learning Object Kinematics from Observations

POMDP-based planning methods require world dynamics models to generate motion plans. Service robots would need to learn such models directly from observations. In this section, we survey existing approaches for learning such models for articulated objects from observations as articulated objects constitute one of the largest sets of household objects service robots will interact with frequently.

2.2.1 Articulation Model Estimation from Visual Observations

Learning articulation properties for objects directly from visual data has been studied via different approaches in the literature [1, 7, 44, 63, 64, 72, 85, 86, 100, 109, 111, 131, 132]. Sturm et al. [131] proposed a probabilistic framework to learn motion models of articulation bodies from human demonstrations. Pillai et al. [111] extended the framework to estimate the articulation model for textured objects directly from raw RGB images by extracting SURF features from the images and tracking them robustly. However, this framework [131] assumes that the objects are governed by a single articulation model, which may not hold true for all objects. For example, a stapler intrinsically changes its articulation state (e.g., rigid vs. rotational) based on the relative angle between its arms. To address this, Niekum et al. [100] proposed an online changepoint detection algorithm, *CHAMP*, to detect both the governing articulation model and the temporal changepoints in the articulation relationships of objects. However, all these approaches are observation-only and may fail to correctly infer the object articulation model under noisy demonstrations or in cases when actions are critical for inference. In this dissertation, we present MICAH (Chapter 5) that uses an action-conditional approach to learn articulation properties for objects and hence is more robust to noise compared to prior methods.

More recently, Abbatematteo et al. [1] and Li et al. [72] proposed methods to learn articulation properties for objects from raw depth images given articulation model category. In a related body of work on object parts mobility estimation, Wang et al. [147] and Yan et al. [153] proposed approaches to segment parts of the object in an input point cloud and estimate their mobility relationships, given a known articulation model category. Alleviating the requirement of having a known articulation model category, we introduce ScrewNet in Chapter 6 that performs category-independent articulation model estimation from depth images. However, ScrewNet [58] and other existing methods [1, 72, 75, 147, 153] for learning articulation models for objects can only predict point estimates for the articulation model parameters. Addressing this, in Chapter 7 we present a novel approach, DUST-net, that predicts a distribution over the articulation model parameters.

2.2.2 Interactive Perception

Another closely related body of work is of *interactive perception* approaches that aim at leveraging the robot's actions to better perceive objects and build accurate kinematic models [12, 63, 64, 85, 86]. Katz and Brock [63] first used this approach to learn articulated motion models for planar objects [63] and later extended it to use RGB-D data to learn 3D kinematics of articulated objects [64]. Martín-Martín et al. [87] and Martín-Martín and Brock [86] further extended the approach and used hierarchical recursive Bayesian filters to develop online algorithms from articulation model estimation from RGB images. Though these approaches use a robot's actions to generate perceptual signals for model estimation, they require the robot's interaction behavior to be pre-scripted by an expert, unlike MICAH (Chapter 5), that can estimate models even from noisy demonstrations given by non-expert humans. Another limitation of current IP approaches is that they still require textured objects for estimating the object articulation model, whereas the methods introduced in this dissertation, ScrewNet (Chapter 6) and DUST-net (Chapter 7) impose no such requirement on the objects.

Recently, Gadre et al. [32] introduced Act the Part (AtP), an approach for learning to interact with articulated objects for discovering and segmenting their parts in the input images. However, AtP primarily focuses only on articulated objects having revolute joints, whereas, ScrewNet and DUST-net can estimate articulation properties of objects belonging to all common articulation model categories, namely revolute, prismatic, rigid, and helical.

2.2.3 Active Learning

In some cases, the provided demonstrations might not be informative enough to estimate the articulation motion model with high confidence. Hausman et al. [44] developed an active learning algorithm to disambiguate the active articulation model of objects by taking informative actions. In another related work, Barragän et al. [7] proposed a decision-theoretic framework that uses Bayesian filtering to take actions for reducing entropy over the governing model type and its parameters for a mechanism. However, both methods assume that the governing model and its parameters do not change with time. Through MICAH [56], we aim to alleviate this assumption by presenting an approach to perform model inference even for objects whose articulation properties changes depending upon their configuration.

2.2.4 Rigid Body Pose Estimation

Articulation model estimation for objects can be viewed as a subset of the body of work on rigid body pose estimation [15, 16, 21, 73, 104, 107, 120, 126, 136, 146, 148, 150]. Byravan et al. developed SE3-nets [15] and SE-3Pose-Nets [16] to learn predictive dynamics models of object motion in a scene from input point-cloud data and applied action vectors which can be used to directly perform robot visuomotor control from input point cloud data. Some selected recent work on estimating point estimates for rigid body poses are [21, 73, 104, 107, 126, 146, 148, 150]. Various methods for estimating distributions over orientation or rigid body transformations have also been proposed in the literature [5, 34, 35, 98, 108, 115, 118, 120, 128, 135]. Gilitschenski et al. [34] proposed Kalman filters that leverage the Bingham distribution to estimate orientation vectors—that is, points on the unit hypersphere. Gilitschenski et al. [34], Arun Srivatsan et al. [5], Srivatsan et al. [128] and Rosen et al. [118] propose strategies that can be used to estimate the rigid body transformation of an object using a combination of Bingham and Guassian distributions, and the von Mises-Fisher distribution, respectively. Further approaches for estimating pose uncertainty have been proposed which are not based on estimating distributions over orientation or rigid body transformations, such as Shi et al. [120], which estimates uncertainty by measuring the disagreement of an ensemble of deep learning models trained on RGB image data. The mathematical model used by methods presented in this dissertation are inspired by these work, but 1) extends them to also represent uncertainty over the configuration of articulated object components about screw axes, and 2) integrates them into a deep learning model that is capable of learning these configurations from raw depth images. In addition, while these approaches use distributions over orientations and rigid body transformations to produce estimates, DUST-net directly outputs a distribution that can be used to facilitate further applications such as uncertainty-aware behavior planning.

2.2.5 Articulated object pose estimation

For known articulated objects, the problem of articulation model parameter estimation can also be treated as an articulated object pose estimation problem. Different approaches leveraging object CAD model information [25, 90, 106] and the knowledge of articulation model category [1, 72, 151, 155] have been proposed to estimate the 6D pose of the articulated object in the scene. These approaches can be combined with an object detection method, such as YOLOv4 [10], to develop a pipeline for estimating the articulation model parameters for objects from raw images. On the other hand, the methods presented in this dissertation, ScrewNet (Chapter 6) and DUST-net (Chapter 7), can directly estimate the articulation model for an object from depth images without requiring any prior knowledge about it.

2.2.6 Human Pose Estimation from images

Human pose estimation is another related field of work [14, 30, 39, 77, 95, 117, 154, 157, 158]. However, as the human body consists of only revolute joints, this body of work assumes that the joint model category for the articulated body, i.e., the human body, is known as a priori. Whereas, the methods we present do not assume the availability of this information and can estimate articulation properties of objects belonging to all common articulation model categories, namely revolute, prismatic, rigid, and helical.

2.2.7 Further approaches for articulation model estimation

Articulation motion models can be viewed as geometric constraints imposed on two or more rigid bodies. Pérez-D'Arpino and Shah [109] have proposed a method, *C-LEARN*, to learn geometric constraints encountered in a manipulation task from non-expert human demonstrations. Subramani et al. [132, 133] developed an approach to learn geometric constraints governing relative motion between objects from human demonstrations. Their proposed approach can successfully learn geometric constraints even from noisy demonstrations. Other approaches for learning articulation properties for objects from human demonstrations have explored leveraging different sensing modalities [24, 75, 76]. Recently, Daniele et al. [24] proposed a multimodal learning framework that incorporates both vision and natural language information for articulation model estimation. Affordances provide yet another framework to represent the relationship between the applied actions on an object and the observed object motion. Sun et al. [134] have proposed an object-object affordance learning approach to model the interactive motions between paired objects, human actions, and object labels as human-object-object affordances. However, these approaches predict point estimates for the articulation model parameters, unlike DUST-net, which predicts a distribution over the articulation model parameters.

Chapter 3

Notation and Preliminaries

In this chapter we introduce the notation that we will use throughout most of this dissertation. For a more in depth treatment of partially observable Markov decision processes, hybrid dynamics, screw transformations, and statistics over Stiefel manifolds, we recommend Thrun et al. [138], Lygeros et al. [79], Siciliano and Khatib [121], and Chikuse [22], respectively.

3.1 Markov Decision Processes

A Markov decision process (MDP) is a model of an agent (a robot, in our case) interacting with a world. The agent takes as input the state of the world and generates output actions affecting the state of the world. Formally, an MDP is defined as a tuple $\langle \mathcal{X}, \mathcal{U}, T, R, \gamma, \mathcal{X}_0 \rangle$, where $\mathcal{X}$ is the set of states of the world, $\mathcal{U}$ is the set of actions, $T : \mathcal{X} \times \mathcal{U} \rightarrow \Pi(\mathcal{X})$ is the *state-transition function*, $R : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and $\mathcal{X}_0$ is the initial state distribution [62, 138]. In the MDP framework, it is assumed that, while there may be uncertainty about the effects of an agent's actions due to stochastic world dynamics or inaccurate world dynamics model, there is no uncertainty about the agent's current state — it is known completely and perfectly.

A policy π is a mapping from states $\mathcal{X}$ to a probability distribution over actions $\mathcal{U}$.

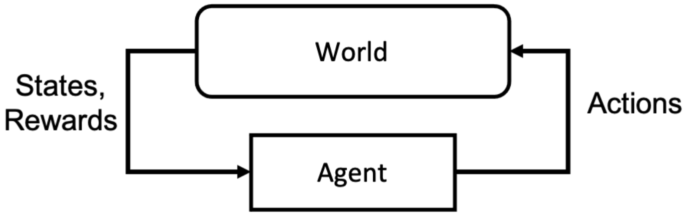


Figure 3.1: An MDP models the interactions between agent and world. An MDP agent aims to find an optimal policy π^* that maximizes the expected discounted sum of future rewards

The value of a policy π under reward function R is defined as the expected return of that policy, $V_R^\pi = \mathbb{E}_{x_0 \sim \mathcal{X}_0}[\sum_{t=0}^{\infty} \gamma^t R(x_t) \mid \pi]$. The value of executing policy π starting at state $x \in \mathcal{X}$ is given as $V_R^\pi(x) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(x_t) \mid \pi, x_0 = x]$. Given an MDP representation of a task, we would like our agents to find an optimal policy π^* under the given reward function R, such that starting in state x if it is executed, it would maximize the expected discounted sum of future rewards $V_R^* = \max_\pi V_R^\pi$.

3.2 Partially Observable Markov Decision Processes

Partially Observable Markov Decision Processes (POMDPs) extend the MDP framework to the cases when the agent is unable to observe the current state. Instead, the agent makes an observation based on the action and the resulting state. Formally, a POMDP is defined as tuple $\langle \mathcal{X}, \mathcal{U}, T, R, \mathcal{Z}, \mathcal{O}, \gamma \rangle$, where $\mathcal{X}, \mathcal{U}, T, R, \gamma$ describe an MDP, $\mathcal{Z}$ is the set of observations the agent can experience of its world, and $\mathcal{O} : \mathcal{X} \times \mathcal{U} \rightarrow \Pi(\mathcal{Z})$ is the observations function, which gives, for each action and resulting state, a probability distribution over possible observations [62]. The goal of the agent stays the same: to maximize expected discounted sum of future rewards.

To account for the uncertainty over its current state, the robot keeps an internal belief state $b[\mathbf{x}_t] \in \mathbb{B}$, that summarizes all past control inputs and sensor measurements,

$$b(\mathbf{x}_t) = p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{u}_{t-1}, \mathbf{z}_1, ..., \mathbf{z}_t) \tag{3.1}$$

where $\mathbf{x}_t \in \mathcal{X}$, $\mathbf{u}_t \in \mathcal{U}$ and $\mathbf{z}_t \in \mathcal{Z}$ are the robot's state, control input and received measurement at time step t, respectively and $\mathbb{B} \subset \{\mathcal{X} \to \mathbb{R}\}$ represent the space of all possible beliefs. If we choose belief states to be probability distributions over states of the world represented as an MDP, the decision process over belief states is also *Markov*, i.e., the belief state at time $t+1$ is dependent only on the belief state at time t and the action at time t [62]. A POMDP can be decomposed into two parts: *state estimation* and action generation. The first part is to update update the current belief state *posterior* based on the last action $\mathbf{u}_t$, the current observation $\mathbf{z}_{t+1}$, and the belief state *prior* b^-_{t+1},

$$b_{t+1}(\mathbf{u}_t, \mathbf{z}_{t+1}, \hat{b}_{t+1}) = \eta_1 \; p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1}) \; \hat{b}_{t+1} \tag{3.2}$$

where η_1 is a normalizing constant and the observation function is defined as $\mathcal{O} = p(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})$. The second part uses a policy π defined over the agent's belief states to generate an action u_{t+1} for the time step $t+1$ given the current belief state posterior b^+_{t+1}. The resulting belief state prior for the time step $t+1$ is

$$\hat{b}_{t+2}(\mathbf{x}_{t+2}, \mathbf{u}_{t+1}, b_{t+1}) = \eta_2 \int p(\mathbf{x}_{t+2}|\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) \; b_{t+1} \; d\mathbf{x}_{t+1} \tag{3.3}$$

where η_2 is a normalizing constant and the transition function for the underlying MDP is given as $T = p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$.

3.3 Hybrid Dynamics

A hybrid dynamics model of a system is a dynamics model in which the states of the system evolve with time over both continuous space $x \in X = \mathbb{R}^N$ and a finite set of discrete

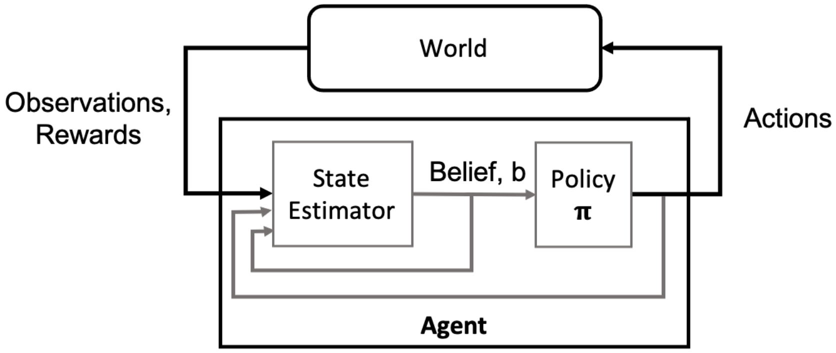


Figure 3.2: A POMDP extends the MDP framework to the case when the agent is unable to observe the current state. A POMDP agent can be decomposed into a state estimator and a policy π

states $q \in Q \subset \mathbb{W}$ [79]. Each discrete state of the system corresponds to a separate dynamics model that governs the evolution of continuous states. These types of dynamical models are sometimes referred to as *switched dynamical systems* in the literature [33].

In a hybrid model, discrete state transitions of the system can be represented as a directed graph with each possible discrete state q corresponding to a node and edges ($e \in E \subseteq Q \times Q$) marking possible transitions between the nodes. These discrete state transitions are conditioned on the continuous states. A transition from the discrete state q to another state q' happens if the continuous states $\mathbf{x}$ are in the *guard set* $G(q, q')$ of the edge $e_q^{q'}$ where $e_q^{q'} = \{q, q'\}$, $G(\cdot) : E \to P(X)$ and $P(X)$ is the power set of X. Thus, for each discrete state q, in a hybrid dynamics model we can define:

$$x_{t+1} = F^q(x_t, u_t), \qquad\qquad z_t = H^q(x_t) \tag{3.4}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $z \in \mathbb{R}^l$, $F^q(x, u)$ and $H^q(x)$ are the continuous state, control input, observation variables, state dynamics and observation functions respectively. Evolution of the discrete state of the system can be modeled by a finite state Markov chain. Defining the

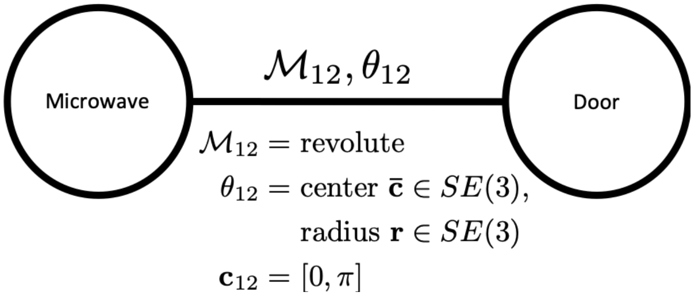


Figure 3.3: Kinematic graph for microwave which considers the kinematic model as only revolute.

state transition matrix as $\Pi = \{\pi_{ij}\}$, the discrete state evolution can be given as:

$$q_{t+1} = \Pi q_t \tag{3.5}$$

3.4 Kinematic Graphs

We represent the kinematic structure for articulated objects using *kinematic graphs* [131]. A kinematic graph $G = (V_G, E_G)$ consists of a set of vertices $V_G = 1, ..., p$, corresponding to the p parts of the articulated object, and a set of undirected edges $E_G \subset V_G \times V_G$, each describing the kinematic link between two object parts. Each joint (ij) is assigned a kinematic model $\mathcal{M}_{ij}$, a parameter vector $\boldsymbol{\theta}_{ij}$, and a set defining the range of the joint $\mathbf{c}_{ij}$. The graph of a microwave, for example, consists of two nodes (a body and a door), with a revolute kinematic model encoded in the edge between the body and door. An example kinematic graph for a microwave is shown in Figure 3.3. In this example, the parameter vector, $\boldsymbol{\theta}_{12}$, encodes the position and orientation of the axis of rotation between the body and the door as well as the radius of the door's rotation, and the joint range, $\mathbf{c}_{ij}$, defines the set of possible values the angle made by the door and the body can take.

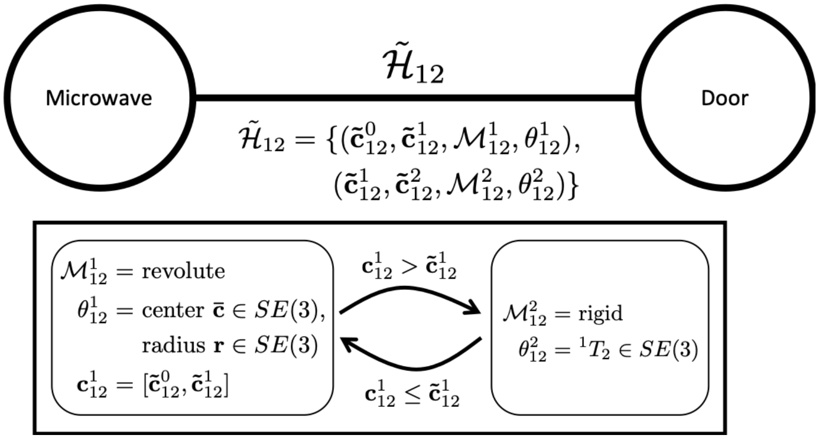


Figure 3.4: Extended kinematic graph for microwave which considers a hybrid model that can be revolute or rigid, depending on the configuration.

Sturm et al. [131] proposed to associate a single kinematic link model $\mathcal{M}_{ij}$ with model parameter vector θ_{ij} with each edge. However, there are many articulated objects with links that are not governed by a single kinematic link model. For example, in most configurations, a microwave door is a revolute joint with respect to the microwave; however, due to the presence of a latch, this relationship changes to a rigid one when the door is closed. In this dissertation, we extend kinematic graphs so that they can represent the hybrid kinematic structure of such objects (see Figure 3.4). Extended kinematic graph construction is discussed in more details in Chapter 5.

3.5 Screw Transformations and Plücker Coordinates

Chasles' theorem states that *"Any displacement of a body in space can be accomplished by means of a rotation of the body about a unique line in space accompanied by a translation of the body parallel to that line"* [121]. This line is called the screw axis of displacement,

S [59, 89]. We use Plücker coordinates to represent this line. The Plücker coordinates of the line $l = \mathbf{p} + x\mathbf{l}$ are defined as $(\mathbf{l}, \mathbf{m})$, with moment vector $\mathbf{m} = \mathbf{p} \times \mathbf{l}$ [59, 89]. The constraints $\|\mathbf{l}\| = 1$ and $\langle \mathbf{l}, \mathbf{m} \rangle = 0$ ensure that the degrees of freedom of the line in space are restricted to four. The rigid body displacement in $SE(3)$ is defined as $\sigma = (\mathbf{l}, \mathbf{m}, \theta, d)$. The linear displacement d and the rotation θ are connected through the pitch h of the screw axis, $d = h\theta$. The distance between $l_1 := (\mathbf{l}_1, \mathbf{m}_1)$ and $l_2 := (\mathbf{l}_2, \mathbf{m}_2)$ is defined as:

$$\mathsf{d}((\mathbf{l}_1, \mathbf{m}_1), (\mathbf{l}_2, \mathbf{m}_2)) = \begin{cases} 0, & \text{if } l_1 \text{ and } l_2 \text{ intersect} \\ \|\mathbf{l}_1 \times (\mathbf{m}_1 - \mathbf{m}_2)\|, & \text{elif } l_1 \parallel l_2 \\ \dfrac{|\mathbf{l}_1 \cdot \mathbf{m}_2 + \mathbf{l}_2 \cdot \mathbf{m}_1|}{\|\mathbf{l}_1 \times \mathbf{l}_2\|}, & \text{else} \end{cases} \tag{3.6}$$

Frame transformations on Plücker lines: Given a rotation matrix R and a translation vector $\mathbf{t}$ between two frames $\mathcal{F}_A$ and $\mathcal{F}_B$, a 3D line displacement matrix $\tilde{D}$ can be defined between the two frames for transforming a line $l := (\mathbf{l}, \mathbf{m})$ from frame $\mathcal{F}_A$ to frame $\mathcal{F}_B$ as:

$$\begin{gathered} \begin{bmatrix} {}^{B}\mathbf{l} \\ {}^{B}\mathbf{m} \end{bmatrix} = {}^{B}\tilde{D}_A \begin{bmatrix} {}^{A}\mathbf{l} \\ {}^{A}\mathbf{m} \end{bmatrix}, \\ \text{where,} {}^{B}\tilde{D}_A = \begin{bmatrix} R & \mathbf{0} \\ [\mathbf{t}]_\times R & R \end{bmatrix}, [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \end{gathered} \tag{3.7}$$

where $[\mathbf{t}]_\times$ denotes the skew-symmetric matrix corresponding to the translation vector $\mathbf{t}$, and $({}^{A}\mathbf{l}, {}^{A}\mathbf{m})$ and $({}^{B}\mathbf{l}, {}^{B}\mathbf{m})$ represents the line l in frames $\mathcal{F}_A$ and $\mathcal{F}_B$, respectively [8].

3.6 Stiefel Manifold and statistics over it

Stiefel manifold: The *Stiefel manifold* $\mathrm{V}_{k,m}$ is the space whose points are sets of k orthonormal vectors in $\mathbb{R}^m$, called k-frames in $\mathbb{R}^m$ ($k \leq m$) [22]. Points on the Stiefel manifold $\mathrm{V}_{k,m}$ are represented by the set of $m \times k$ matrices X such that $X^T X = I_k$, where

I_k is the $k \times k$ identity matrix; thus $\mathrm{V}_{k,m} = \{X_{m,k}; X^T X = I_k\}$. Some special cases of the Stiefel manifold are the unit hypersphere $\mathrm{V}_{1,m}$ in $\mathbb{R}^m$ for $k = 1$, and the orthogonal group $O(m)$ of $m \times m$ orthonormal matrices for $m = k$.

Matrix von Mises-Fisher distribution: A random matrix X on $\mathrm{V}_{k,m}$ is said to have the matrix von Mises-Fisher distribution (or matrix Langevin distribution), if its density function is given by $\mathcal{F}(\mathbf{X}|m, \mathbf{F}) = \dfrac{1}{{}_0F_1(\frac{m}{2}, \frac{1}{4}\mathbf{F}^T\mathbf{F})} \exp(\mathrm{Tr}(\mathbf{F}^T\mathbf{X}))$, where $\mathbf{F}$ is any $m \times k$ matrix and ${}_0F_1$ is a hypergeometric function with matrix argument [22]. We can write the general (unique) singular value decomposition (SVD) of $\mathbf{F}$ as $\mathbf{F} = \Gamma\Lambda\Omega^T$, where $\Gamma \in \tilde{\mathrm{V}}_{k,m}$, $\Omega \in O(k)$, $\Lambda = diag(\lambda_1, ..., \lambda_k)$, $\lambda_1 \geq ... \geq \lambda_k \geq 0$, $\tilde{\mathrm{V}}_{k,m}$ denotes the set of matrices $\Gamma \in \mathrm{V}_{k,m}$ with the property that all the elements of the first row of the matrix Γ are positive, and $O(k)$ denoting the orthogonal group in k dimensions. It can be shown that ${}_0F_1(\frac{m}{2}, \frac{1}{4}\mathbf{F}^T\mathbf{F}) = {}_0F_1(\frac{m}{2}, \frac{1}{4}\Lambda^2)$. For more details, we refer to [22].

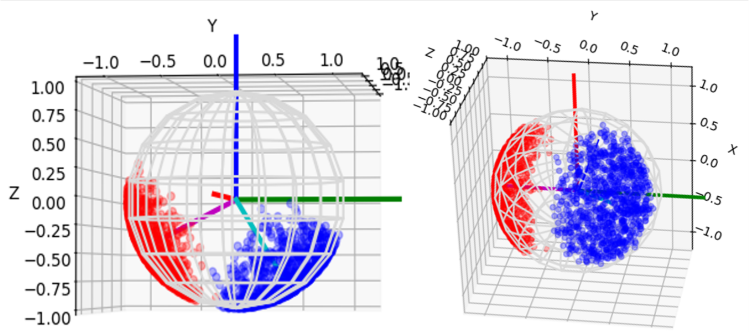


Figure 3.5: Matrix von Mises-Fisher distribution over $\mathrm{V}_{3,2}$, X, Y, Z axes are shown in red, blue and green colors, respectively. Magenta and cyan colors denote vectors corresponding to the first and second column of the matrix $M \in \mathrm{V}_{3,2}$ representing the mode of the distribution

Chapter 4

Robot Motion Planning Under Uncertainty and Hybrid Dynamics

One of the biggest challenges in robot motion planning is to develop feasible motion plans for systems having highly nonlinear dynamics in the presence of partial or noisy observations. Often, these nonlinearities are caused by sudden transitions or discontinuities in the dynamics (for example, due to contacts in a robot manipulation task). When task dynamics change suddenly in state space, even small state estimation errors can lead to large deviations and plan failure. Therefore, reasoning about uncertainty over states becomes crucial in order to develop robust motion plans. In this chapter, we introduce a novel hierarchical POMDP-based motion planner for tasks with hybrid dynamics, the POMDP-HD planner [55], as the first contribution of this thesis. We propose to leverage a natural, simplifying assumption that the nonlinear dynamics of robot motion planning tasks can be decomposed into a discrete set of simpler local dynamics models, of which only one is active at any given time (e.g. a change in dynamics due to contact). Note that these local dynamics models may be approximate, especially when they are learned from data or are a simplification of a complex underlying model. A complete dynamics model can then be defined as a *hybrid dynamics* model having hybrid states comprised of the continuous states of the system along with a discrete state denoting the active local dynamics model.

The POMDP-HD planner garners numerous benefits. Due to hierarchical structure, it breaks long-horizon planning problems into multiple smaller segments that can be sequenced to find a complete solution. Since POMDP planning becomes exponentially more difficult with longer horizons [62, 103], a hierarchical approach breaks the problem into chunks that can be solved with significantly less effort. Another major benefit of planning with hybrid states is that the planner can choose to leverage a specific local dynamics model in order to improve the effectiveness of the generated plans. For example, if it is known a priori that in the k-th local dynamics model, motion is allowed only along a particular vector (e.g. due to presence of a wall), it can be used to reduce the state uncertainty along the dimensions orthogonal to the allowed motion vector. This indirect feedback for uncertainty reduction is critical for tasks in which observations are highly noisy, or even entirely unavailable (for example, due to occlusions).

4.1 Hierarchical POMDP Planner

We propose to solve the problem of motion planning under uncertainty for tasks governed by highly nonlinear dynamics as a POMDP problem defined on a hybrid dynamics model. Different local dynamics models constituting the task dynamics are represented as distinct discrete states of the hybrid model. Under uncertainty over the robot state, a separate discrete distribution needs to be maintained to represent our confidence over the active local dynamics model at each time step. Jointly, a hybrid belief over the hybrid state of the system can be defined with a continuous part representing uncertainty over the robot state and a discrete part representing uncertainty in the active local dynamics model. In this work, we assume that the continuous part of hybrid belief is represented by a mixture of $\mathcal{L}$ Gaussian distributions, each having a mixing weight of α_l, given as:

$$b_t^x = \sum_{l=1}^{L} \alpha_l \mathcal{N}(\mu_l, \Sigma_l) \tag{4.1}$$

4.1.1 Belief Propagation under Hybrid Dynamics

A hybrid belief is defined as $B = \{b^x, b^q\}$, where b^x and b^q correspond to the belief over continuous robot state, $\mathbf{x}$, and discrete states, q, respectively. Propagation of hybrid beliefs using Bayesian filtering can be separated into two steps: making a prediction using the dynamics model to obtain a belief *prior* and updating it based on the received observation to compute the belief *posterior*.

4.1.1.1 Belief Prior

We extend the system dynamics, $F^q(x_t, u_t)$, for uncertainty propagation and represent it as $\mathcal{F}^q(b_t^x, u_t)$. At each time step t, we can propagate the current belief b_t^x through the system dynamics of each discrete state $\mathcal{F}^q(x_t, u_t)$ individually and then take a weighted sum of the propagated belief set to obtain a belief prior for the next time step $\hat{b}_{t+1}^x$, as:

$$\hat{b}_{t+1}^x = \sum_{q'} \mathcal{F}^{q'}(b_t^x, u_t)\ b_t^q[q'] \tag{4.2}$$

where $b_t^q[q'] = p(q_t = q'|x_t)$ is q'-th component of b_t^q, and x_t, q_t and u_t represent the continuous states, discrete state, and continuous control input to the system at time t, and $\hat{b}[\mathbf{x}_{t+1}]$ is denoted as $\hat{b}_{t+1}^x$. Note that $\mathcal{F}^{q'}(x_t, u_t)$ represents a general dynamics function and can be stochastic. Under stochastic continuous state dynamics, the definition of the discrete state transition matrix as given in Equation 3.5 needs to be extended. Assuming the transitions of discrete states are given by a directed graph with self-loops, we can define the extended discrete state transition matrix Π at time t as $\Pi_t = \{p(q^j_{t+1}|q^i_t, \hat{b}^x_{t+1})\} \;\; \forall q^i, q^j \in Q$ where

$$p(q^j_{t+1}|q^i_t, \hat{b}^x_{t+1}) = \begin{cases} \eta \int_{\mathbb{R}^N} \mathbb{1}^{q^j}_{q^i}(\mathbf{x})\hat{b}^x_{t+1}(\mathbf{x})d\mathbf{x}, & if \; \exists \; e^{q^i}_{q^j}, \\ \epsilon, & \text{otherwise} \end{cases} \tag{4.3}$$

where $\mathbb{1}^{q^j}_{q^i}(\mathbf{x})$ is an indicator function defined as:

$$\mathbb{1}^{q^j}_{q^i}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in G(q^i, q^j) \\ 0, & otherwise \end{cases} \tag{4.4}$$

where $\eta = \sum_{k=1}^{|Q|} \pi_t(i,k)$ is a normalization constant, and ϵ is a small probability to handle cases when received observations do not correspond to any legal discrete transition. Calculating the extended discrete state transition matrix Π_t at each time step using Eq. 4.3 can be computationally expensive. An approximation of Π_t can be obtained by sampling n random points from the belief over continuous states b^x_{t+1} and calculating ratio of points lying in the guard set $G(q^i, q^j)$ to the total number of sampled points for each discrete state q^j.

4.1.1.2 Belief Posterior

We use a hybrid estimation algorithm based on Bayesian filtering to reduce the uncertainty over states using noisy continuous state observations. The proposed algorithm consists of two layers of filters: the first estimates the continuous states of the system and the second estimates the discrete states of the system. Upon receiving observation z_{t+1}, the continuous state prior is updated by taking a weighted sum of a bank of extended Kalman filters running independently, with each discrete mode having an individual filter. The weights for the sum are determined using the prior for the discrete mode $\hat{b}^q_{t+1}$. The complete update step for continuous states can be written as:

$$b^x_{t+1} = \hat{b}^x_{t+1} + \sum_{q'} \Big(\mathbf{K}^{q'}_{t+1}(z_{t+1} - \mathcal{H}^{q'}_{t+1}(\hat{b}^x_{t+1}))\Big)\hat{b}^q_{t+1}[q'] \tag{4.5}$$

where $\mathbf{K}^{q'}_{t+1}$ is the Kalman Gain for discrete state q' at time t+1 and $\hat{b}^q_{t+1}[q']$ is q'-th component of $\hat{b}^q_{t+1}$. The update for the discrete state can be obtained by using a Bayesian filter update given as:

$$b^q_{t+1} = \gamma \mathbf{M}_{t+1} \circ \hat{b}^q_{t+1} \tag{4.6}$$

where $\mathbf{M}_{t+1} = [P(z_{t+1}|q_{t+1} = q')]^T \;\; \forall q' \in Q$, $\circ$ is the element-wise multiplication operator, $\gamma = \dfrac{1}{\sum_{q'} \mathbf{M}_{t+1} \circ \hat{b}^{q'}_{t+1}}$ is a normalization constant and

$$P(z_{t+1}|q_{t+1} = q') = z_{t+1} \sim \mathcal{H}^{q'}_{t+1}(b^x_{t+1}) \tag{4.7}$$

where $\mathcal{H}^{q'}_{t+1}(.)$ is the observation function for state q'. Mixing weights for the mixture of Gaussians are also updated based on the received observations as

$$\alpha^l_{t+1} = \mathcal{N}(z_{t+1} - \hat{z}^l_{t+1}|\mathbf{0}, \Sigma^l_{t+1}), \quad where, \quad \hat{z}^l_{t+1} = \sum_{q'} \hat{b}^q_{t+1}[q'\ ]\ (\mathcal{H}^{q'}_{t+1}\mu^l_{t+1}) \tag{4.8}$$

A new mixture of $\mathcal{L}$ Gaussians is then chosen to represent the continuous belief b^x_{t+1} at the next step.

4.1.2 Direct Planning

With the hybrid belief propagation equations defined, we can now use trajectory optimization technique to solve the POMDP. We assume *maximum likely observations (MLO)* obtained by propagating the current belief over continuous states through the system dynamics (Eqn. 4.2) as the true observations for developing locally optimal motion plans, as

Algorithm 1: High-Level Plan → Continuous State Goals

```
Function high_level_plan_to_countinuous_state_goals (high-level plan)
    for each $q^k$ in high-level plan do
        Define corresponding full-confidence vector, $W^k_{full\_conf} = \begin{cases} 1, & \text{if } q = q^k_{goal} \\ 0, & \text{else} \end{cases}$
        Sample $n$ random points: $X_{sample} = \{x_1, ..., x_n\} \sim \mathcal{X}$;
        for each $x_i \in X_{sample}$ do
            Find confidence distribution on discrete states $w_i \in W_{sample}$:
              Sample a random set $X' \sim \mathcal{X}$;
              for each $q' \in \mathcal{Q}$ do
                  $w_i(q') = \frac{|x' \in X' \cap G(q', q'') \; \forall q''|}{|X'|}$;
            Find cost of divergence $c_i \in C' \subset \mathbb{R}$: $c_i(x_i) = Hellinger(w_i, W^k_{full\_conf})$;
        Define cost map on complete domain $\mathcal{X}$: $C_{complete}(\mathbf{x})$ = Interpolate($C'$);
        Find best representative point in continuous state: $x^k_{best} = global\_optimization(\mathbf{x}, C_{complete})$;
        Append $x^k_{best}$ to $X_{cs\_goals}$;
    return $X_{cs\_goals}$;
```

introduced by Platt et al. [61]. In this work, the nonlinear optimization problem set up for trajectory optimization is posed as a sequential least squares programming (SQP) problem and solved using the SNOPT software package [36, 37]. We denote this approach as the *direct planning* approach.

4.1.3 Hierarchical Planner

Although the direct planning approach can be used to solve the POMDP, planning for longer horizons in complex tasks, such as contact-rich manipulation tasks, can result in infeasible computational costs [103]. To tackle this challenge, we propose a hierarchical planner that decomposes the POMDP problem into smaller subproblems which can be solved with significantly less effort.

The proposed hierarchical planner has two levels: a higher level to find the best sequence of local dynamics models that should be visited along the path (by visiting corresponding regions in continuous state space) and a lower level that is similar to the aforementioned direct planning approach. The higher level planner generates a set of candidate high-level plans consisting of all feasible permutations (without repetitions) of the discrete states of the task*. A transition between two discrete states is deemed to be an infeasible transition, if the regions of the continuous state space corresponding to the two discrete states form a pair of positively-separated sets.

We define the term *confidence* to denote the probability of a continuous state belief to be in a particular discrete state. Spatial distribution of confidence across the continuous domain for a particular discrete state is defined as the *confidence map* associated with that state. A confidence map for a particular discrete state can be converted into a cost map by calculating a cost of divergence between a full-confidence vector ($W^k_{full\_conf}$, one-hot vector with probability of being in that particular state equals to one) and the confidences at randomly sampled points across the domain. A high-level plan can then be converted into a sequence of continuous state goals by finding the global minimum of such cost maps associated with each discrete state in the plan (see Algorithm 1). The lower level planner is then called for each of these continuous state goals and a complete continuous state path for the high-level plan is generated by combining the outputs of lower level planner. An

---

*The feasibility check also helps in keeping the POMDP tractable. Gulyás et al. [40] have shown that the average path length for a connected graph decreases as its graph connectivity increases. If the graph of discrete states, from which the set of feasible high-level plans is derived, is not sparse enough to solve the POMDP tractably, a simple heuristic can be defined that penalizes plans with longer path lengths. Preferential choice of shorter plans results in fewer calls to the lower level planner and reduces computational time.

additional discrete state is added to each high-level plan which represents the desired goal of the task and is considered to be active within an $\epsilon-$neighbourhood of the actual task goal. High-level plans are then ranked by calculating a divergence cost on the distribution of planner's confidence on the active discrete state at the final point of the plan and the desired confidence distribution (all the probability mass within the $\epsilon-$neighbourhood of the goal). The continuous state plan corresponding to the high-level plan with the minimum cost is chosen to be executed.

In this work, we have used *Hellinger distance* [18] to calculate the divergence cost between the discrete distributions as it forms a symmetric bounded metric with a value between 0 and 1, and was found to be more numerically stable than the Bhattacharya distance, KL-divergence, and Jensen–Shannon divergence on the tested application domains. Radial basis functions were used to interpolate the divergence costs throughout the domain and the differential evolution method was used to find the approximately globally optimal solutions of the generated cost map [129].

4.1.4 Trajectory Stabilization

With the MLO assumption, it is very likely that during execution the belief over robot state will diverge from the *nominal* trajectory planned. To ensure that the execution phase belief follows the plan, a *belief space LQR* (B-LQR) controller can be defined around the nominal trajectory. B-LQR controllers were introduced by Platt et. al [61] and can be seen as belief-space extension of Linear-Quadratic Regulators (LQR). For systems modelled as linear-Gaussian processes, a B-LQR controller is optimal and equivalent to a linear-Quadratic Gaussian (LQG) controller. In B-LQR, each point in the nominal trajectory is defined as a set point and quadratic costs are defined for the distance from it and the control effort required to converge to it. Closed form solutions exist to ensure convergence to the set point within a finite time horizon. While stabilizing the trajectory, the most likely active discrete state is taken to define the governing dynamics of the system. However, it may happen that the B-LQR controller is unable to stabilize the execution phase (actual) belief around the nominal trajectory. If the planned belief for the next step deviates more than a δ-threshold from the actual belief after the observation update, a replanning call to the planner is triggered.

4.2 Experiments

The proposed POMDP solver for hybrid dynamics was tested on two tasks: autonomous navigation and localization in a walled-domain with extremely poor observations, and a real manipulation task of partially assembling a toy airplane [17] under noisy observations by leveraging contacts to reduce uncertainty.

4.2.1 Domain-I: Walled Domain

The first task is an autonomous navigation task in a 2D domain ($\{x, y\} \in [-2, 15]$) having extremely noisy observations ($w \sim \mathcal{N}(\cdot|0, 15\ units)$, zero-mean Gaussian noise). The domain consists of two perpendicular walls parallel to the x and y axis respectively. As the motion along a wall is constrained to be only parallel to the wall, the robot can use it to efficiently localize itself in a direction orthogonal to the wall. We compare the performance of the hierarchical planner with the direct planning approach. Note that the direct planning approach is similar in principle to the SM-POMDP planner proposed by Brunskill et al. [13]

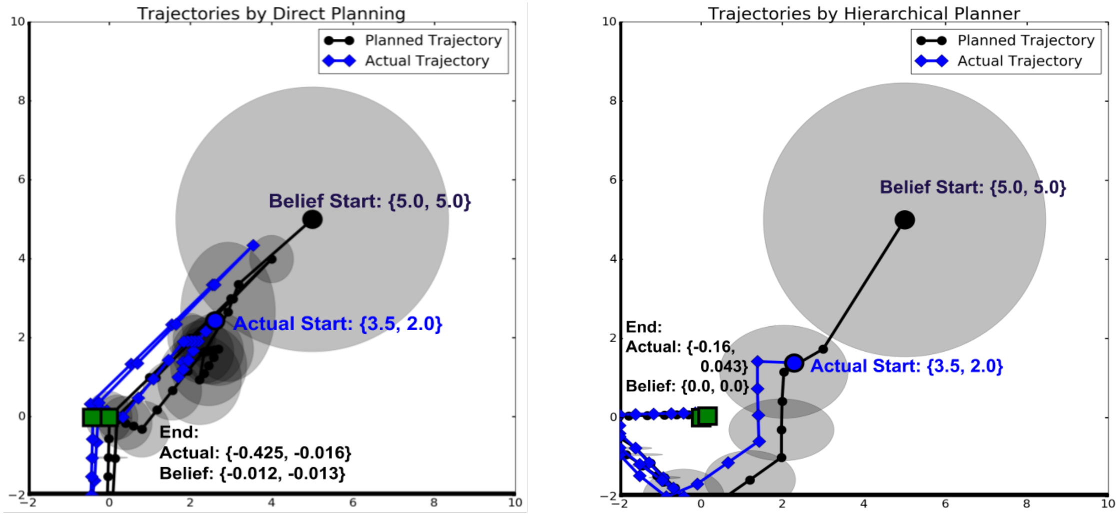
and hence, provides a comparison of the proposed hierarchical planner with a flat, single-level planning approach. Hybrid dynamics model can be given as

$$f(\mathbf{x}_t, \mathbf{u}) = \begin{cases} \mathbf{x}_t + \mathbf{u}, & \text{if } x > -2, y > -2 \\ \mathbf{x}_t + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u}, & \text{if } x < -2 \\ \mathbf{x}_t + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u}, & \text{if } x > -2, y < -2, \end{cases} \tag{4.9}$$

where $\mathbf{x}_t = \{x_t, y_t\}^T$. The observation function was defined as $h(x_t) = x_t + w$. Matrices defining the cost function over error in states, control input, additional cost for final state error and covariance were taken as $Q = diag(0.5, 0.5)$, $R = diag(10.0, 10.0)$, $Q_T = 1e4$ and $\Lambda = 1e7$ respectively. Number of Gaussians used to model continuous belief $L = 1$.

Sample trajectories planned by the direct planning and the hierarchical planner are shown in Figure 4.1. It is evident from the figures that the hierarchical planner plans to selectively visit the two discrete states representing the walls, in contrast to the direct method. Also, the hierarchical planner is able to converge to the goal faster and with a much lower uncertainty than the direct planning approach. As the direct planner does not leverage the knowledge of local dynamics models in a structured way, it needs to plan longer trajectories to gather more information. However, due to high noise in the observations, it still fails to converge to the goal with high accuracy.

Additional statistical analysis to compare the two approaches in terms of total planning time, final error and final belief uncertainty are presented in Table 4.1. It can be seen from Table 4.1 that, for comparable final error and final belief uncertainty, the hierarchical planner is able to find a solution approximately 5 times faster than the direct planning approach.



(a) Direct Planning

(b) Hierarchical Planning

Figure 4.1: A comparison of planned and actual trajectories using the direct planning and hierarchical planning approaches on the walled domain. For both cases, Initial belief mean $\mu = \{5, 5\}$, $cov = diag(11.5, 11.5)$, True start position:=$\{3.5,\ 2.0\}$. Gray circles represent belief covariance.

4.2.2 Domain-II: Airplane assembly

We experimentally demonstrate that the hierarchical POMDP planner can be used to tractably solve a real world manipulation task — the partial assembly of a toy airplane from the YCB dataset [17]. We considered the first step of inserting the landing gear into the wing as a test case for our planner. The task requires a high precision with maximum tolerance of $\pm 0.2\ cm$. Feedback on the location of the airplane in world was noisy and had an average estimation error of $\pm 2.0\ cm$. This experiment demonstrates two important features of the proposed planner: first, the planner can be scaled to solve real-world manipulation planning under uncertainty problems and second, due to the hierarchical planning approach, the planner essentially enables the robot to plan and "feel around" to localize itself when observations are noisy, similar to what a human might do.

In a robot manipulation task involving contacts, based on the type of contact between the bodies, the number of state-dependent local dynamics models can be large, or even infinite. We simplify the problem by assuming an approximate hybrid dynamics model, in which the local dynamics models correspond to possible motion constraints that the robot can encounter while executing the task. For example, the task of placing a cup on a table can be considered to be approximately made of two local dynamics models: one when the two objects are not in contact and the other when the cup is in contact with the table plane. The second dynamics model represents the motion constraint placed on the cup by the table by restricting its motion to be only along its plane and not penetrating it. This approximation helps in having a succinct and effective representation of the task dynamics; under this approximation, for a specific set of inputs, the relative motion between the two objects in contact will always be the same, independent of the type of contact between them. In this case, the specific set of inputs would be the set of all inputs which do not result in moving the cup away from the table plane, resulting in breaking the contact between them.

In this experiment, we consider the domain to be made up of four distinct local

Metric	**Direct**	**Hierarchical**
Average Total time (in seconds)	51.908	**10.695**
Average Final Error	$[-0.168, 0.172]^T$	$[\mathbf{0.086}, \mathbf{0.198}]^T$
Average Final Maximum Belief Uncertainty	0.696	**0.625**

Table 4.1: Comparison of direct and hierarchical planning. Values are averaged over 5 runs. Planning horizon: 20 steps. Belief start: $[5, 5]^T$. actual start: $[3.5, 2.0]^T$. Termination condition: Maximum likelihood estimate of belief converged within a ball of 0.2 *unit* radius around the goal ($[0, 0]^T$) with max covariance of 1 *unit*.

dynamics models: two corresponding to the linear motions along the wing plane edges, one corresponding to the corner of the plane and one to represent free-body motion elsewhere in the domain. At the highest level, the planning problem can be broken down into two steps: first, to localize the gear at a point in a plane parallel to the wing and second, to insert the gear into the hole. A hybrid dynamics model in a plane parallel to the wing can be given as

$$f(\mathbf{x}_t, \mathbf{u}) = \begin{cases} \mathbf{x}_t + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} + \mathbf{v}, & \text{if } x \in [4, 4.5], y > -13.5 \\ \mathbf{x}_t + \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \mathbf{u} + \mathbf{v}, & \text{if } x < 4, y \in [-14, -13] \\ \mathbf{x}_t + \mathbf{0} * \mathbf{u} + \mathbf{v}, & \text{if } x \in [4, 4.5], y \in [-14, -13.5] \\ \mathbf{x}_t + \mathbf{u} + \mathbf{v}, & \text{otherwise} \end{cases} \tag{4.10}$$

where $\mathbf{v}$ is process noise, modeled as $v \sim \mathcal{N}(\cdot|0, \mathbf{I}_2)$ with 1 unit = 1 cm. The observation function $h(x_t) = x_t + w$ with zero-mean Gaussian observation noise $w \sim \mathcal{N}(\cdot|0, 2\mathbf{I}_2)$. The planner took 14.682 seconds for planning on an Intel$^{\circledR}$ Core$^{\text{TM}}$ i7-6700 CPU @3.40GHz, 16Gb RAM. Feedback was obtained on the location of the airplane in the world frame by doing an online color-based object cluster extraction, using multi-plane segmentation from the Point Cloud Library (PCL) on the point cloud data of a Microsoft Kinect v2 sensor. Matrices defining the cost function over error in states, control input, additional cost for final state error and covariance were taken as $Q = diag(0.5, 0.5)$, $R = diag(0.1, 0.1)$, $Q_T = 5000$ and $\Lambda = 1e7$ respectively. Number of Gaussians used to model continuous belief $L = 1$.

Figure 4.3 shows snapshots of the trajectory executed by the robot during the task from two perpendicular angles. The right Panel of Figure 4.2 shows the trajectory planned by the hierarchical planner and the actual trajectory taken by the robot in a plane parallel to the wing. It can be see from Fig. 4.2 that the planner plans to activate the motion constraint

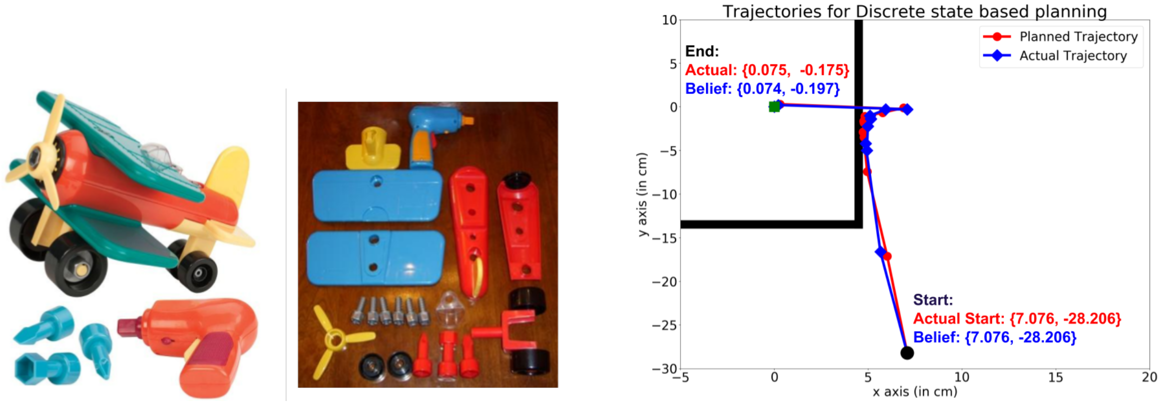


Figure 4.2: *Left Panel*: Toy Airplane from YCB Dateset. *Right Panel*:Planned and Actual trajectories for the airplane assembly task in a plane parallel to wing plane. Bold black lines represents the edges of the airplane wing. 1 unit = 10 cm.

parallel to the wing in order to reduce its uncertainty. Once localized in the plane parallel to the wing, the robot changes planes to move to a point directly above the hole and then proceeds to insert the landing gear into the wing.

4.3 Conclusion

We present a hierarchical POMDP planner for hybrid dynamics which can develop locally optimal motion plans for tasks involving nonlinear dynamics under noisy observations. We proposed to model nonlinear task dynamics, especially due to sudden changes in dynamics, using a hybrid dynamics model. The POMDP-HD planner generates hierarchical motion plans at two levels: first, a high-level motion plan that sequences the local dynamics models to be visited and second, based on the best high-level plan, a detailed continuous state motion plan to be followed by the robot. The hierarchical planning approach breaks the large POMDP problem into multiple smaller segments with shorter planning horizons,

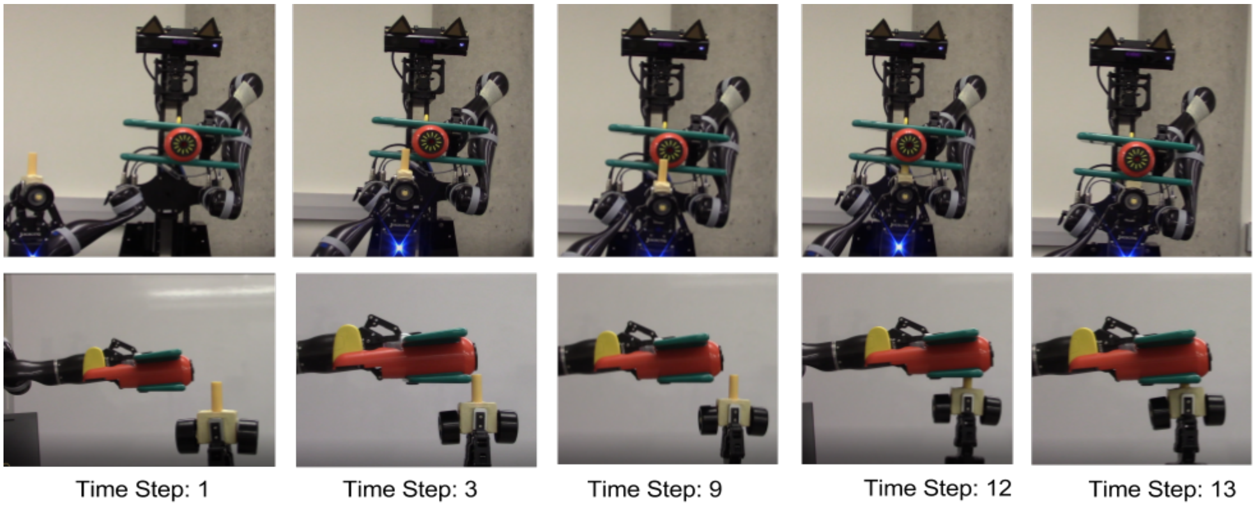


Figure 4.3: Snapshots of the robot assembling the toy airplane

which significantly increases the computational efficiency of the planner. High-level planning also enables the robot to leverage task dynamics to improve its performance—for example, reducing uncertainty using the task motion constraints in order to develop motion plans which are more robust to state uncertainty. However, the hierarchical planner requires access to a task dynamics model for developing motion plans. In the next chapter, we partially alleviate this requirement by presenting an algorithm to learn planning-compatible hybrid kinematic models for articulated objects directly from visual observations.

Chapter 5

Learning Hybrid Object Kinematics for Efficient Hierarchical Planning Under Uncertainty

Robots working in human environments need to perform dexterous manipulation on a wide variety of objects. Such tasks typically involve making or breaking contacts with other objects, leading to sudden discontinuities in the task dynamics. Furthermore, many objects exhibit configuration-dependent dynamics, such as a refrigerator door that stays closed magnetically. While the presence of such nonlinearities in task dynamics can make it challenging to represent good manipulation policies and models, if well-understood, these nonlinearities can also be leveraged to improve task performance and reduce uncertainty. For example, when inserting a screw into the underside of a table, if direct visual feedback is not available, indirect feedback from wiggling the screw (a semi-rigid connection between the screw and the table) can be leveraged to ascertain whether the screw is inserted or not. In other words, the sensed change in dynamics (from free-body motion to rigid contact) serves as a landmark, partially informing the robot about the state of the system and reducing uncertainty. Such dynamics can be naturally represented as hybrid dynamics models or *hybrid automata* [79], in which a discrete state represents which continuous dynamics model is active at any given time.

Current model-free reinforcement learning approaches [41, 67, 99, 152] can learn to cope with hybrid dynamics implicitly, but require large amounts of data to do so, scale poorly as the problem complexity grows, face representational issues near discontinuities, and do not transfer well to significantly different problems. Conversely, hierarchical POMDP planning-based methods [13, 55, 67, 142] can represent and reason about hybrid dynamics directly, scale well via plan decomposition, work well on novel problems, and reason about uncertainty, but typically rely on precise hand-specified models and task decompositions. In this chapter, we introduce **M**odel **I**nference **C**onditioned on **A**ctions for **H**ierarchical Planning (MICAH) that bridges this gap and enables hierarchical POMDP planning-based methods, such as the POMDP-HD planner from the previous chapter, to perform novel manipulation tasks given noisy observations. MICAH infers hybrid automata for objects with configuration-dependent dynamics from unsegmented sequences of observed poses of object parts. These automata can then be used to perform motion planning under uncertainty for novel manipulation tasks involving these objects.

MICAH consists of two parts, corresponding to our two main contributions: (1) an novel action-conditional inference algorithm called Act-CHAMP for kinematic model estimation and changepoint detection from unsegmented data, and (2) an algorithm to construct hybrid automata for objects using the detected changepoints and estimated local models from Act-CHAMP. Due to action-conditional inference, MICAH is more robust to noise and less vulnerable to several modes of failure than existing model inference approaches [86, 100, 111]. These prior approaches assume that the visual pose observations alone provide sufficient information for model estimation, which does not hold for many scenarios and can lead to poor performance. For example, an observation-only approach cannot distinguish between observations obtained by applying force against a rigid object and taking no action at all on a free body, estimating that the model is rigid in both the cases.

To evaluate our method, we first show that for articulated objects, MICAH can correctly infer changepoints and the associated local models with higher fidelity and less data than a state-of-the-art observation-only algorithm, CHAMP [100]. We also consider four classes of noisy data to demonstrate its robustness to noise. Next, to test the planning-compatibility of the learned models, we learn hybrid automata for a microwave and a drawer from human demonstrations and use them with a recently proposed hierarchical POMDP Planner, POMDP-HD [55], to successfully manipulate them in new situations. Lastly, we show that the learned models through MICAH are rich-enough to be leveraged creatively by a hierarchical planner for completing novel tasks efficiently—we learn a hybrid automaton for a stapler and use it to dexterously place the stapler at a target point that is reachable only through a narrow corridor in the configuration space.

5.1 Preliminaries

5.1.1 Changepoint Detection

Given a time series of observations $\mathbf{y}_{1:n}$, a changepoint model introduces a number of temporal *changepoints* $\tau_1, ..., \tau_m$ that split the data into a set of disjoint segments, with each segment assumed to be governed by a single model (though different models can govern different segments). We build on the online MAP (maximum *a posteriori*) changepoint detection model proposed by Fearnhead and Liu [29], which was specialized for detecting motion models for articulated objects by Niekum et al. [100]. Given a time series of observations $\mathbf{y}_{1:n}$ and a set of parametric candidate models M, the changepoint model infers the MAP set of changepoint times $\boldsymbol{\tau} = \{\tau_0, \tau_1, ..., \tau_m, \tau_{m+1}\}$ where $\tau_0 = 0$ and $\tau_{m+1} = n$, giving us $m+1$ segments. Thus, the k^{th} segment consists of observations $\mathbf{y}_{\tau_k+1:\tau_{k+1}}$, and has an associated model $\mathcal{M}^k \in \mathbb{M}$ with parameters θ^k.

Assuming that the data after a changepoint is independent of the data prior to that changepoint, we model the position of changepoints in the time series as a Markov chain in which the transition probabilities are defined by the time since the last changepoint,

$$p(\tau_{i+1} = t | \tau_i = j) = \beta(t - j) \tag{5.1}$$

where $\beta(\cdot)$ is a probability distribution over time. For a segment from time s to t, the model evidence for the governing model being $\mathcal{M}$, is defined as:

$$L(s, t, \mathcal{M}) = p(\mathbf{y}_{s+1:t} | \mathcal{M}) = \int p(\mathbf{y}_{s+1:t} | \mathcal{M}, \theta) p(\theta) d\theta \tag{5.2}$$

The distribution over the position of the most recent changepoint prior to time t, C_t, can be efficiently estimated using the standard Bayesian filtering recursions and an online Viterbi algorithm [29]. We define E_s as the event that given a changepoint at time s, the MAP choice of changepoints has occurred prior to time s. Then, the probability of having a changepoint at time t, P_t, is defined as:

$$\begin{aligned} P_t &= p(C_t = s, \mathcal{M}, E_s, \mathbf{y_{1:t}}) \\ P_t^{MAP} &= p(\text{Changepoint at t}, E_s, \mathbf{y_{1:t}}) \end{aligned} \tag{5.3}$$

which results in

$$\begin{aligned} P_t(s, \mathcal{M}) &= (1 - B(t - s - 1))\ L(s, t, \mathcal{M})\ p(\mathcal{M})\ P_s^{MAP} \\ P_t^{MAP} &= \max_{s, \mathcal{M}} \left[\frac{\beta(t - s)}{1 - B(t - s - 1)} P_t(s, \mathcal{M}) \right] \end{aligned} \tag{5.4}$$

where $B(\cdot)$ is the cumulative distribution function of $\beta(\cdot)$. By finding the values of $(s, \mathcal{M})$ that maximize P_t^{MAP}, the Viterbi path can be recovered at any point. This process can be repeated until the time $t = 0$ is reached to estimate all changepoints that occurred in the given time series $\mathbf{y}_{1:T}$.

The algorithm is fully online, but requires $O(n)$ computations at each time step, since $P_t(s, \mathcal{M})$ values must be calculated for all $s < t$. The computation time is reduced to a constant by using a particle filter that keeps a constant number of particles, M, at each time step, each of which represents a support point in the approximate density $p(C_t = s, \mathbf{y}_{1:t})$. If at any time step, the number of particles exceeds M, stratified optimal resampling [29] is used to choose which particles to keep such that the Kolmogorov-Smirnov distance from the true distribution is minimized in expectation.

5.2 MICAH

Given a sequence of object part pose observations $\mathcal{D}_\mathbf{y}$ (e.g. from a visual tracking algorithm) and a sequence of applied actions $\mathcal{D}_\mathbf{a}$ on an articulated object, MICAH creates a planning-compatible hybrid automaton for the object. It does so in two steps: (1) it estimates the kinematic graph $\hat{G}$ representing the kinematic structure of the object given the sequence of pose observations $\mathcal{D}_\mathbf{y}$ and the applied actions $\mathcal{D}_\mathbf{a}$, and then (2) constructs a hybrid automaton H representing the motion model for the object given $\hat{G}$.

For the first step, we extend the framework proposed by Sturm et al. [131] in two important ways to better learn the kinematic structure of articulated objects. First, we include reasoning about the applied actions along with the observed motion of the object while estimating its kinematic structure. Second, we extend the framework to be able to learn the kinematic structure of more complex articulated objects that may exhibit configuration-dependent kinematics, e.g., a microwave. The original framework [131] assumes that each link of an articulated body is governed by a single kinematic model. For complex articulated objects that exhibit configuration-dependent kinematics, the transitions points in the kinematic model along with the set of governing local models and their parameters need to be estimated to learn the complete kinematic structure of the object.

To facilitate these extensions, we introduce a novel action-conditional changepoint detection algorithm, **Act**ion conditional **Ch**angepoint detection using **A**pproximate **M**odel **P**arameters (Act-CHAMP), that can detect the changepoints in the relative motion between two rigid objects (or two object parts), given a time series of observations of the relative motion between the objects and the corresponding applied actions. The algorithm is described in section 5.2.1.

Kinematic trees have the property that their edges are independent of each other. As a result, when learning the kinematic relationship between object parts i and j of an articulated object, only their relative transformations are relevant for estimating the edge model. MICAH first uses the Act-CHAMP algorithm to learn the kinematic relationships between different parts of the articulated object separately, and then combines them to estimate the complete kinematic graph $\hat{G}$ for the object. Once the kinematic graph $\hat{G}$ for an articulated object is known, MICAH constructs a hybrid automaton H to represent its motion model. We choose hybrid automata as they present a natural choice to model the motion of objects that may exhibit different motion models based on their configuration. Steps to construct a hybrid automation from the learned kinematic graph $\hat{G}$ is described in section 5.2.2.

5.2.1 Action-conditional Model Inference

Following Sturm et al. [131], we define the relative transform between two objects with poses x_i and $x_j \in SE(3)$ at time t as: $\boldsymbol{\Delta}_{ij,t} = \mathbf{x}_{i,t} \ominus \mathbf{x}_{j,t}$*. Additionally, we define an action $\mathbf{a}_t$ taken by the demonstrator at a time t as the intended displacement to be applied to the relative transform between two objects from time t to $t+1$ as: $\mathbf{a}_t = \boldsymbol{\Delta}_{ij,t} \ominus \boldsymbol{\Delta}_{ij,t+1}$. Given the time-series of observations $\mathcal{D}_{\mathbf{y}_{ij}} = \mathbf{y}_{1:T}$ of relative motion between the two object parts i and j of an articulated object and the corresponding applied actions $\mathcal{D}_{\mathbf{a}_{ij}} = \mathbf{a}_{1:T}$, we wish to find the set $\tilde{\mathcal{H}}_{ij}$ defining the kinematic relationship between the two object parts. The set $\tilde{\mathcal{H}}_{ij}$ consists of tuples $(\tilde{\mathbf{c}}_{ij}^{k-1}, \tilde{\mathbf{c}}_{ij}^{k}, \mathcal{M}_{ij}^{k}, \theta_{ij}^{k})$, where $\tilde{\mathbf{c}}_{ij}^{k-1}$ and $\tilde{\mathbf{c}}_{ij}^{k}$ denote the starting and the end configurations for the model $\mathcal{M}_{ij}^{k}$ to be the governing local model with parameters θ_{ij}^{k}. For the sake of clarity, we drop the subscript $\{ij\}$ in the following discussion in this section.

We propose a novel algorithm, Act-CHAMP, that performs action-conditional change-point detection to estimate the set $\tilde{\mathcal{H}}_{ij}$ given input time series of observations $\mathbf{y}_{1:T}$ and the corresponding applied actions $\mathbf{a}_{1:T}$. Act-CHAMP builds upon the CHAMP algorithm proposed by Niekum et al. [100]. The CHAMP algorithm reasons only about the observed relative motion between the objects for estimating the kinematic relationship between the objects. However, an observation-only approach can easily lead to false detection of change-points and result in an inaccurate system model. Consider an example case of deducing the motion model for a drawer from a noisy demonstration in which the majority of applied ac-

---

*The operators $\oplus$ and $\ominus$ represent motion composition operations. For example, if poses $\mathbf{x}_i$, $\mathbf{x}_j \in \mathcal{R}^{4\times4}$ are represented as homogeneous matrices, then these operators correspond to matrix multiplications $\mathbf{x}_i \oplus \mathbf{x}_j = \mathbf{x}_i\mathbf{x}_j$ and its inverse multiplication, $\mathbf{x}_i \ominus \mathbf{x}_j = \mathbf{x}_i^{-1}\mathbf{x}_j$, respectively.

tions are orthogonal to the axis of motion of the drawer. Due to intermittent displacements, an observation-only approach might model the motion of the drawer to be comprised of a sequence of multiple rigid joints. On the other hand, an action-conditional inference can maintain an equal likelihood of observing either a rigid or a prismatic model under off-axis actions, leading to a more accurate model.

Given the two time series inputs $\mathbf{y}_{1:T}$ and $\mathbf{a}_{1:T}$, we define the model evidence for model $\mathcal{M}$ being the governing model for the time segment between times s and t as:

$$L(s,t,\mathcal{M}) = p(\mathbf{y}_{s+1:t}|\mathcal{M},\mathbf{a}_{s:t-1}) = \int p(\mathbf{y}_{s+1:t}|\mathcal{M},\theta,\mathbf{a}_{s:t-1})p(\theta)d\theta \tag{5.5}$$

Each model $\mathcal{M}$ admits two functions: a forward kinematics function, $f_{\mathcal{M},\theta}$, and an inverse kinematics function, $f^{-1}_{\mathcal{M},\theta}$, which maps the relative pose between the objects $\mathbf{\Delta}_{ij}$ to a unique configuration $\mathbf{c}$ for the model (e.g. a position along the prismatic axis, or an angle with respect to the axis of rotation) as:

$$f_{\mathcal{M},\theta}(\mathbf{c}_{\mathcal{M}}) = \mathbf{\Delta} \qquad \text{(forward kinematics)}$$

$$f^{-1}_{\mathcal{M},\theta}(\mathbf{\Delta}) = \mathbf{c}_{\mathcal{M}} \qquad \text{(inverse kinematics)}$$

We consider three candidate models $\mathcal{M}^{\text{rigid}}$, $\mathcal{M}^{\text{revolute}}$, and $\mathcal{M}^{\text{prismatic}}$ to define the kinematic relationship between two objects. Complete definitions of forward and inverse kinematics models for these models are beyond the scope of this work; for more details, see Sturm et al. [131].

Additionally, we define the Jacobian and inverse Jacobian functions for the model $\mathcal{M}$

as

$$J_{\mathcal{M},\theta}(\delta\mathbf{c}_{\mathcal{M}}) = \delta\mathbf{\Delta} \qquad \text{(Jacobian)}$$

$$J^{-1}_{\mathcal{M},\theta}(\delta\mathbf{\Delta}) = \delta\mathbf{c}_{\mathcal{M}} \qquad \text{(inverse Jacobian)}$$

where $\delta\mathbf{\Delta}$ and $\delta\mathbf{c}_{\mathcal{M}}$ represent small perturbations applied to the relative pose and the configuration, respectively.

Using these functions, we can define the likelihood of obtaining observations $\mathbf{y}_{1:T}$ upon applying action $\mathbf{a}_{1:T}$ under model $\mathcal{M}$ as:

$$p(\mathbf{y}_{2:T}|\mathcal{M}, \theta, \mathbf{a}_{1:T-1}) = \prod_{t=2}^{T} p(\mathbf{y}_t \mid \hat{\mathbf{\Delta}}_t) \tag{5.6}$$

where $\hat{\mathbf{\Delta}}_t$ is the predicted relative pose under the model $\mathcal{M}$ at time t, and can be calculated using the observation $\mathbf{y}_{t-1}$ and applied action $\mathbf{a}_{t-1}$ at time $t-1$ as:

$$\hat{\mathbf{\Delta}}_t = f_{\mathcal{M},\theta}(\ f^{-1}_{\mathcal{M},\theta}(\mathbf{y}_{t-1}) + J^{-1}_{\mathcal{M},\theta}\ \mathbf{a}_{t-1}) \tag{5.7}$$

The probability $p(\mathbf{y}_t \mid \hat{\mathbf{\Delta}}_t)$ can be calculated by defining an observation model, given an observation error covariance Σ_y for the perception system as:

$$\mathbf{y}_t \sim \begin{cases} \mathbf{\Delta}_t + \mathcal{N}(0, \Sigma_y) & \text{if } \nu = 1 \\ \mathcal{U} & \text{if } \nu = 0 \end{cases} \tag{5.8}$$

where the probability of observation being an outlier is $p(\nu = 0) = \gamma$, in which case it is drawn from a uniform distribution $\mathcal{U}$. The data likelihood is then defined as:

$$p(\mathbf{y}_t|\mathbf{\Delta}_t) = p(\mathbf{y}_t|\mathbf{\Delta}_t, \gamma)p(\gamma), \quad \text{where,} \tag{5.9}$$

$$p(\mathbf{y}_t|\mathbf{\Delta}_t, \gamma) = (1-\gamma)p(\mathbf{y}_t|\nu = 1) + \gamma p(\mathbf{y}_t|\nu = 0), \tag{5.10}$$

$$p(\gamma) \propto e^{-w\gamma}, \tag{5.11}$$

and w is a weighting constant.

Finally, similar to Niekum et al. [100], we can define our BIC-penalized likelihood function as:

$$\ln L(s, t, \mathcal{M}) \approx \ln p(\mathbf{y}_{s+1:t} | \mathcal{M}, \hat{\theta}, \mathbf{a}_{s:t-1}) - \frac{1}{2} k_q \ln(t - s) \tag{5.12}$$

where estimated parameters $\hat{\theta}$ are inferred using MLESAC (Maximum Likelihood Estimation Sample Consensus) [140]. This likelihood function can be used in conjugation with the changepoint detection algorithm described in section 5.1.1 to infer the MAP set of changepoint times $\boldsymbol{\tau}$ along with the associated local models $\mathcal{M}^k \in \mathbb{M}$ with parameters θ^k. The detected changepoints $\boldsymbol{\tau}$ and the local models can be later combined appropriately to obtain a set $\mathcal{H}_{ij}$ consisting of tuples $(\tau_{ij}^{k-1}, \tau_{ij}^k, \mathcal{M}_{ij}^k, \theta_{ij}^k)$, where τ_{ij}^{k-1} and τ_{ij}^k denote the starting and the end changepoints for the time segment k in the input time series $\mathbf{y}_{1:T}$.

The transition conditions between the local models can be made independent of the changepoint times, $\boldsymbol{\tau}$, by making use of the observations corresponding to the changepoint times $\mathbf{y}_\tau \subseteq \mathbf{y}_{1:T}$. If an observation $\mathbf{y}_{\tau^k}$ corresponds to the changepoint τ^k denoting the transition from local model $\mathcal{M}^k$ to the model $\mathcal{M}^{k+1}$, then the inverse kinematics function $f^{-1}_{\mathcal{M}^k, \theta^k}$ can be used to find an equivalent configurational changepoint $\tilde{\mathbf{c}}^k$, a fixed configuration for model $\mathcal{M}^k$, that marks the transition from model $\mathcal{M}^k$ to the next model $\mathcal{M}^{k+1}$. We can thus convert the set $\mathcal{H}_{ij}$ to the set $\tilde{\mathcal{H}}_{ij}$, consisting of tuples $(\tilde{\mathbf{c}}_{ij}^{k-1}, \tilde{\mathbf{c}}_{ij}^k, \mathcal{M}_{ij}^k, \theta_{ij}^k)$, that is independent of the input time series.

The complete kinematic structure of the articulated object can then be estimated by finding the set of edges E_G, denoting the kinematic connections between its parts, that maximizes the posterior probability of observing $\mathcal{D}_z$ under applied actions $\mathcal{D}_a$ [131]. However,

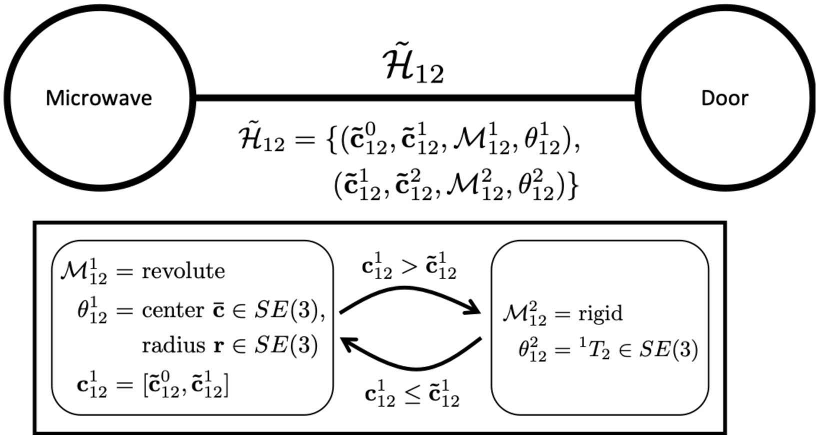


Figure 5.1: Extended kinematic graph for microwave which considers a hybrid model that can be revolute or rigid, depending on the configuration.

to account for complex articulated objects that exhibit configuration-dependent kinematics, now each edge $\tilde{e}_{ij} \in E_G$ of the kinematic graph G can correspond to multiple kinematic link models $\mathcal{M}^k_{ij}$, unlike the original framework [131], in which each edge corresponds to only one kinematic link model $\mathcal{M}_{ij} \in \mathbb{M}_{ij}$. To denote the change, we call such kinematic graphs, *extended kinematic graphs*. An example extended kinematic graph for a microwave is shown in Figure 5.1.

5.2.2 Hybrid Automaton Construction

Hybrid automata present a natural choice for representing an articulated object that can have a discrete number of configuration-dependent kinematics models. A hybrid automaton can model a system that evolves over both discrete and continuous states with time effectively, which facilitates robot manipulation planning for tasks involving that object. We define the hybrid automaton $\mathsf{H} = (Q, X, U, Init, f, I, \mathcal{E}, \mathcal{G}, R, \phi)$ for the articulated object as:

- $Q = \prod_{\tilde{e}_{ij} \in E_G} \mathbb{M}_{ij}$, i.e. the Cartesian product of the sets of local models defining kinematic relationship between two object parts;

- $X = \prod_{\tilde{e}_{ij} \in E_G} \bar{c}_{ij}$, where we use a single variable $\bar{c}_{ij} \in \mathbb{R}$ to represent the configuration value $\mathbf{c}_{\mathcal{M}^k}$ under all models $\mathcal{M}_{ij}^k \in \mathbb{M}_{ij}$, as each of the candidate articulation models admits a single-dimensional configuration variable $\mathbf{c}_{\mathcal{M}} \in \mathbb{R}$;

- $U = U_C = \prod_{\tilde{e}_{ij} \in E_G} u_{ij}$, where $u_{ij} \in \mathbb{R}$ is the input delta to be applied to the continuous state $\bar{c}_{ij}$ and the set of discrete input variables is the null set $U_D = \emptyset$ as we cannot control the discrete states directly;

- $Init$ is defined as per the task definition;

- The vector field f governing the evolution of the continuous state vector x with time is defined as $f(q, x, u) = (\mathbf{x}_t - \mathbf{x}^q) + \mathbf{u}_t$, where $q \in Q$, $\mathbf{x}_t, \mathbf{x}^q \in X$, and $\mathbf{u} \in U$. The vector $\mathbf{x}^q \in X$ is so defined that its l-th element $\mathbf{x}^q[l] = \sum_{r=0}^{k-1} \tilde{\mathbf{c}}_{ij}^r$, where l-th dimension of X corresponds to the kinematic relationship between object parts i and j with $\tilde{e}_{ij} \in E_G$, and $q[l] = \mathcal{M}_{ij}^k$;

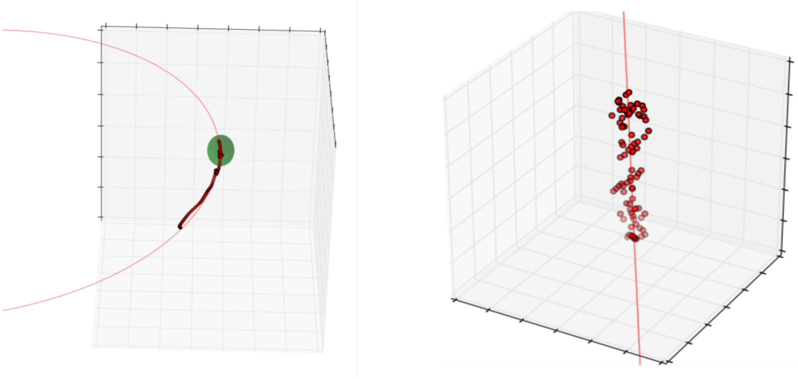
- For each discrete state $q \in \mathbf{Q}$, an invariant set $I(q)$ is defined such that within it the time evolution of the continuous states is governed by the vector field $f(q, x, u)$ $\forall x \in I(q) \subseteq X, u \in U$. We define I as $I = \prod_{\tilde{e}_{ij} \in E_G} Dom(\mathbb{M}_{ij})$, where $Dom(\mathbb{M}_{ij}) = \{Dom(\mathcal{M}_{ij}^k) \forall k \in |\mathbb{M}|\}$ with $Dom(\mathcal{M}_{ij}^k)$ defined as $Dom(\mathcal{M}_{ij}^k) = [0, \tilde{\mathbf{c}}_{ij}^{k+1})$;

- The set of edges defines the set of feasible transitions between the discrete states, $\mathcal{E} = \{(q, q') \mid q[l] = \mathcal{M}_{ij}^k \Rightarrow q'[l] = \mathcal{M}_{ij}^r, r \in \{k-1, k, k+1\}\}$;

- Guards $\mathcal{G}$ can be constructed using the configurational changepoints estimated for the object. If an edge $e = (q, q') \in \mathcal{E}$ corresponds to a transition from a local model $\mathcal{M}_{ij}^{k}$ to model $\mathcal{M}_{ij}^{k+1}$, then the guard for the edge e can be defined as $\mathcal{G}(e) = \{\bar{c}_{ij} \geq \tilde{\mathbf{c}}_{ij}^{k+1}\}$. Analogously, the guard for the reverse transition $\mathcal{G}(e' = (q', q)) = \{\bar{c}_{ij} < 0\}$. To handle the corner cases when $\bar{c}_{ij} < 0$ for $\mathcal{M}_{ij}^{1}$ or $\bar{c}_{ij} > \hat{\mathbf{c}}_{ij}^{m+1}$ for model $\mathcal{M}_{ij}^{m}$ (assuming $|\mathbb{M}_{ij}| = m$), we define two additional edges e_{ij}^{0} and e_{ij}^{-1} which corresponds to the self transitions to the same discrete states such that $\bar{c}_{ij}$ is lower-bounded at 0 for $\mathcal{M}_{ij}^{1}$ and upper-bounded at $\hat{\mathbf{c}}_{ij}^{m+1}$ for model $\mathcal{M}_{ij}^{m}$;
- The reset map R is an identity map;
- The set of admissible inputs $\phi(q, \mathbf{x}) = \mathbf{U}$.

5.3 Experiments and Discussions

In the first set of experiments, we compare the performance of Act-CHAMP with the CHAMP algorithm [100] to estimate changepoints and local motion models for a microwave and a drawer. Next, we test the complete method, MICAH, to construct planning-compatible hybrid automata for the microwave and drawer and discuss the results of manipulation experiments to open and close the microwave door and the drawer using the learned models. Finally, we show that MICAH can be combined with a recent hierarchical POMDP planner, POMDP-HD [55], to develop a complete pipeline that can learn a hybrid automaton from demonstrations and leverage it to perform a novel manipulation task—in this case, with a stapler. A video showcasing the experiments is available at: `https://youtu.be/f35gMoOoOy8`.

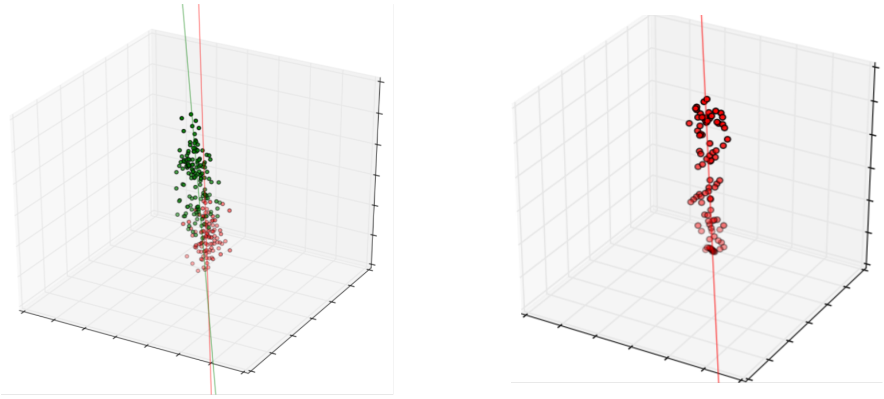


(a) Microwave (b) Drawer

Figure 5.2: Inferred motion models for the microwave and the drawer using Act-CHAMP. Points denote the recorded relative poses of object parts from one demonstration. The green circle represents the detected rigid model, the circular arc represents the detected revolute model, and the line represents the detected prismatic model.

5.3.1 Learning Kinematics Models for Objects

We collected six sets of demonstrations to estimate motion models for the microwave and the drawer. We provided kinesthetic demonstrations to a two-armed robot, in which the human expert physically moved the right arm of the robot, while the left arm shadowed the motion of the right arm to interact with objects while collecting unobstructed visual data. The first two sets provide low-noise data, by manipulating the door handle or drawer knob via a solid grasp. The next two sets provide data in which random periods of no actions on the objects were deliberately included while giving demonstrations. The last two sets consist of high-noise cases, in which the actions were applied by pushing with the end-effector without a grasp. Relative poses of object parts were recorded as time-series observations with an RGB-D sensor using the SimTrack object tracker [105]. For each time step t, the demonstrator's action $\mathbf{a}_t$ on the object was defined as the difference between the



(a) CHAMP

(b) Act-CHAMP

Figure 5.3: Act-CHAMP correctly infers the drawer motion model, while CHAMP (baseline) falsely detects a changepoint under noisy demonstrations.

position of the right end-effector at times t and $t + 1$.

With grasp: Both algorithms (CHAMP and Act-CHAMP) detected a single changepoint in the articulated motion of the microwave door and determined the trajectory to be composed of two motion models, namely rigid and revolute. For the drawer, both algorithms were able to successfully determine its motion to be composed of a single prismatic motion model(see Table 5.1). This demonstrates that for clean, information-rich demonstrations, Act-CHAMP can perform on par with the baseline.

No-Actions: When no action is applied to an object, due to the lack of motion, an observation-only model inference algorithm can infer the object motion model to be rigid. Moreover, if the agent stops applying actions after interacting with the object for some time, an observation-only approach can falsely detect a changepoint in the motion model. We hypothesize that an action-conditional inference algorithm such as Act-CHAMP won't suffer from these shortcomings as it can reason that no motion is expected if no actions are applied. To test it, we conducted experiments in which the demonstrator stopped applying actions on the object midway during a demonstration for an extended time randomly at two distinct locations. As expected, the observation-only CHAMP algorithm falsely detected changepoints in the object motion model and performed poorly (see Table 5.1). However, as Act-CHAMP reasons about the applied actions as well, it performed much better (see Table 5.1).

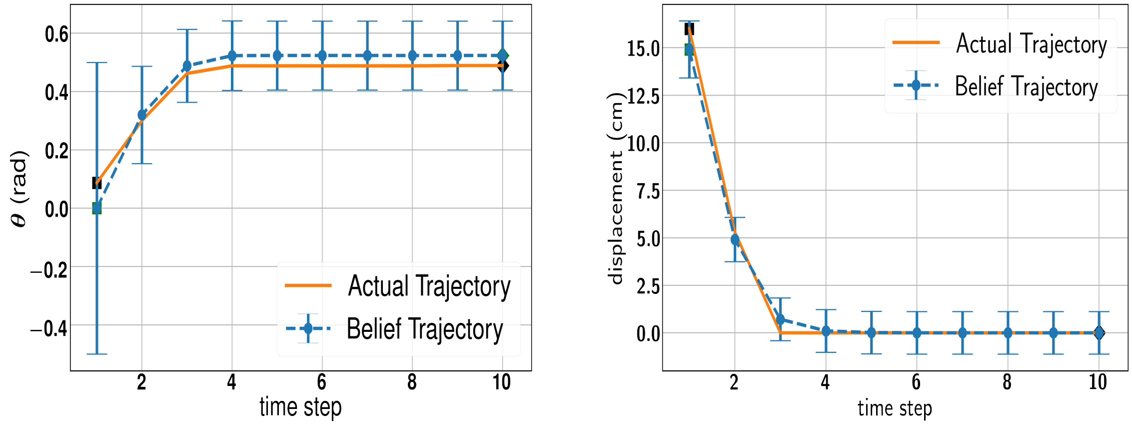
Without grasp: When actions are applied directly on the object (microwave door and the drawer, respectively), the majority of the applied actions are orthogonal to the axis of motion leading to low-information demonstrations. In such a case, while CHAMP almost completely failed to detect correct motion models for the microwave (5% success), Act-CHAMP was able to correctly detect models in almost one-third of the trials (see Table 5.1). For the drawer, CHAMP falsely detected a changepoint and determined that the articulation motion model is composed of two separate prismatic articulation models with different model parameters (Figure 5.3). However, due to action-conditional inference, Act-CHAMP correctly classified the motion to be composed of only one articulation model (Figure 5.3, see Table 5.1).

5.3.2 Object Manipulation Using Learned Models

To test the effectiveness of the learned hybrid automata using MICAH, we used them to perform the tasks of opening and closing a microwave door and a drawer using a robot manipulator. We use the POMDP-HD planner [55] to develop manipulation plans. Figure 5.4 shows the belief space and actual trajectories for the microwave and drawer manipulation tasks. For both the objects, low final errors were reported: 0.05 ± 0.01 rad for the microwave

Case	Object	Algorithms	Changepoint Detection		Error in Model Parameters
			Correct No.	Position Error	
With grasp	**Microwave**	CHAMP	**20/20 (100%)**	0.001 ± 0.001	**Center:** 0.046 ± 0.024 m **Axis:** 0.02 ± 0.01 rad **Radius:** 0.027 ± 0.024 m
		ActCHAMP	**20/20 (100%)**	0.001 ± 0.001	**Center:** 0.066 ± 0.034 m **Axis:** 0.03 ± 0.02 rad **Radius:** 0.013 ± 0.059 m
	Drawer	CHAMP	**20/20 (100%)**	—	**Axis:** 0.04 ± 0.01 rad
		ActCHAMP	**20/20 (100%)**	—	**Axis:** 0.04 ± 0.01 rad
No Actions	**Microwave**	CHAMP	11/20 (55%)	0.001 ± 0.001	**Center:** 0.015 ± 0.006 m **Axis:** 0.01 ± 0.01 rad **Radius:** 0.011 ± 0.006 m
		ActCHAMP	**14/20 (70%)**	0.001 ± 0.001	**Center:** 0.019 ± 0.009 m **Axis:** 0.01 ± 0.01 rad **Radius:** 0.002 ± 0.014 m
	Drawer	CHAMP	4/20 (20%)	—	**Axis:** 0.03 ± 0.01 rad
		ActCHAMP	**12/20 (60%)**	—	**Axis:** 0.03 ± 0.01 rad
Without grasp	**Microwave**	CHAMP	1/20 (5%)	0.001	**Center:** 0.51 m **Axis:** 0.42 rad **Radius:** 0.056 m
		ActCHAMP	**6/20 (30%)**	0.001 ± 0.001	**Center:** 0.328 ± 0.125 m **Axis:** 0.58 ± 0.20 rad **Radius:** 0.164 ± 0.103 m
	Drawer	CHAMP	9/20 (45%)	—	**Axis:** 0.18 ± 0.01 rad
		ActCHAMP	**15/20 (75%)**	—	**Axis:** 0.14 ± 0.04 rad

Table 5.1: Model detection comparison



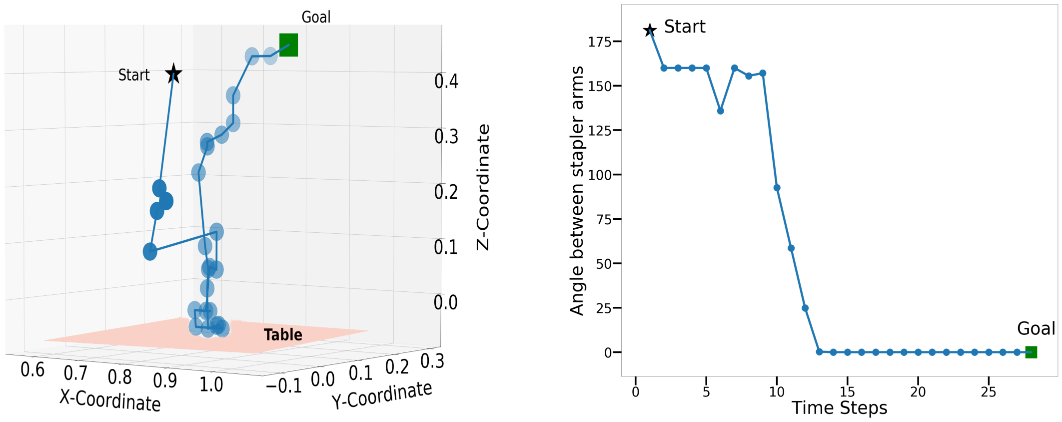
(a) Microwave (b) Drawer

Figure 5.4: Plots showing belief space [blue] and actual trajectories [orange] for microwave and drawer manipulation tasks using learned models. Error bars represent belief uncertainty.

and 0.005 ± 0.003 m for the drawer (average of 5 different tasks), validating the effectiveness of the learned automata.

5.3.3 Leveraging Learned Models for Novel Manipulations

Finally, we show that our learned models and planner are rich enough to be used to complete novel tasks under uncertainty that require intelligent use of object kinematics. To do so, we combine MICAH with the POMDP-HD planner for performing a manipulation task of placing a desk stapler at a target point on top of a tall stack of books. Due to the height of the stack, it is challenging to plan a collision-free path to deliver the stapler to the target location through a narrow corridor in the free configuration space of the robot; if the robot attempts to place the stapler at the target point while its governing kinematic model is revolute, the lower arm of the stapler will swing freely and collide with the obstacle. However, a feasible collision-free motion plan can be obtained if the robot first closes and



(a) Stapler: $\{x, y, z\}$ (b) Stapler: θ vs time

Figure 5.5: Planned trajectories for the stapler placement experiment. (Left) in $\{x, y, z\}$ (Right) Relative angle of the stapler arms over time.

Figure 5.6: Snapshots showing the executed trajectory for the stapler placement task. The red region denotes the unreachable workspace for the robot's end-effector.

locks the stapler (i.e. rigid articulation), and then proceeds towards the goal. To change the state of the stapler from revolute to rigid, the robot can plan to make contact with the table surface to press down and lock the stapler in a non-prehensile fashion.

As the task involves making and breaking contacts with the environment, we need to extend the learned hybrid motion model of the stapler to include local models due to contacts. We approximately define the contact state between the stapler and the table as to be either a line contact (an edge of the lower arm of the stapler in contact with the table), a surface contact (the lower arm lying flat on the table) or no contact. The set of possible local models for the hybrid task kinematics can be obtained by taking a Cartesian product of the set of possible discrete states for the stapler's hybrid automaton and the set of possible contact states between the stapler and the table. However, if the stapler is in the rigid mode, its motion would be the same under all contact states. Hence, a compact task kinematics model would consist of four local models—the stapler in revolute mode with no contact with the table, the stapler in revolute mode with a line contact with the table, the stapler in revolute mode with a surface contact with the table, and the stapler in rigid mode.

Given a human demonstration of robot's interaction with the stapler as input, MICAH first learns a hybrid automaton for the stapler and then extends it to the hybrid task model using the provided task-specific parameters. Next, the POMDP-HD planner uses the learned task model to develop motion plans to complete the task with minimum final state uncertainty. Note that only the final Cartesian position for the stapler was specified as the target for the task and not the articulation state of the stapler (rigid/revolute). Motion plans generated by the planner are shown in Figure 5.5. As can be seen from the plots, the planner plans to make contacts with the table to reduce the relative angle between the stapler arms and change the articulation model of the stapler. The plan drags the stapler along the surface of the table, indicating that it waits until it is highly confident that the stapler has become rigid before breaking contact. Making contacts with the table along the path also helps in funneling down the uncertainty in the stapler's location relative to the table in a direction parallel to the table plane normal, thereby increasing the probability of reaching the goal successfully. Figure 5.6 shows snapshots of the motion plan and actual execution of the robot performing the task.

5.4 Conclusion

Robots working in human environments require a fast and data-efficient way to learn motion models of objects around them to interact with them dexterously. We present a novel method MICAH, that performs action-conditional model inference from unsegmented human demonstrations via a novel algorithm, Act-CHAMP, and then uses the resulting models to construct hybrid automata for articulated objects. Action-conditional inference enables articulation motion models to be learned with higher accuracy than the prior methods in the presence of noise and leads to the development of models that can be used directly for manipulation planning. Furthermore, we demonstrate that the learned models are rich enough to be used for performing novel tasks with such objects in a manner that has not been previously observed.

One limitation of Act-CHAMP is that it requires time series of observations of relative poses between the object parts as input, which are often not available in unstructured human environments. In the next chapter, we address this limitation and present ScrewNet, a method that estimates articulation properties for objects directly from raw sensory input.

Chapter 6

Category-Independent Articulation Model Estimation From Depth Images Using Screw Theory

Human environments are populated with objects that contain functional parts, such as refrigerators, drawers, and staplers. These objects are known as articulated objects and consist of multiple rigid bodies connected via mechanical joints such as hinge joints or slider joints. Service robots will need to interact with these objects frequently. For manipulating such objects safely, a robot must reason about the articulation properties of the object. Safe manipulation policies for these interactions can be obtained directly either by using expert-defined control policies [9, 53] or by learning them through interactions with the objects [41, 67]. However, this approach may fail to provide good manipulation policies for all articulated objects that the robot might interact with, due to the vast diversity of articulated objects in human environments and the limited availability of interaction time. An alternative is to estimate the articulation models through observations, and then use a planning [55] or model-based RL method [67] to manipulate them effectively.

Existing methods for estimating articulation models of objects from visual data either use fiducial markers to track the relative movement between the object parts [76, 100, 131] or require textured objects so that feature tracking techniques can be used to observe this motion [56, 86, 111]. These requirements severely restrict the class of objects on which these methods can be used. Alternatively deep networks can extract relevant features from raw images automatically for model estimation [1, 72]. However, these methods assume prior knowledge of the articulation model category (revolute or prismatic) to estimate the category-specific model parameters, which may not be readily available for novel objects encountered by robots in human environments. Addressing this limitation, in this chapter, we present a novel approach, ScrewNet, which uses screw theory to perform articulation model estimation directly from depth images without requiring prior knowledge of the articulation model category. ScrewNet unifies the representation of different articulation categories by leveraging the fact that the common articulation model categories (namely revolute, prismatic, and rigid) can be seen as specific instantiations of a general constrained relative motion between two objects about a fixed screw axis. This unified representation enables ScrewNet to estimate the object articulation models independent of the model category.

ScrewNet garners numerous benefits over existing approaches. First, it can estimate articulation models directly from raw depth images without requiring a priori knowledge of the articulation model category. Second, due to the screw theory priors, a single network suffices for estimating models for all common articulation model categories unlike prior methods [1, 72]. Third, ScrewNet can also estimate an additional articulation model category, the helical model (motion of a screw), without making any changes in the network architecture or the training procedure.

To evaluate ScrewNet, we conduct a series of experiments on two benchmarking datasets: the simulated articulated objects dataset provided by Abbatematteo et al. [1], and the PartNet-Mobility dataset [20, 93, 149], and three real-world objects: a microwave, a drawer, and a toaster oven. We first test the performance of ScrewNet in estimating the

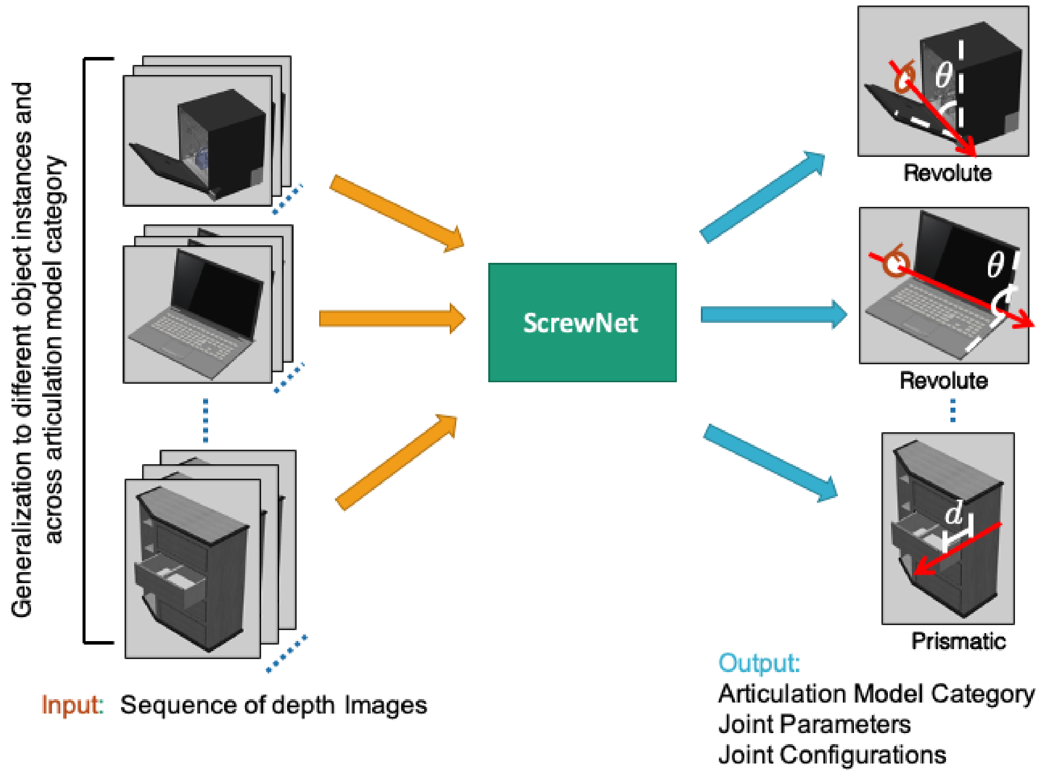


Figure 6.1: ScrewNet estimates the articulation model for objects directly from depth images and can generalize to novel objects within and across articulation model categories

articulation model parameters for unseen object instances from depth images that belong to the same object classes as seen during the training. Next, we evaluate its performance in estimating the parameters for unseen object instances belonging to novel object classes from a single articulation model category. Finally, we test its performance in estimating the articulation model parameters for unseen object instances of known object classes, but across articulation model categories. We compare the performance of ScrewNet with a current state-of-the-art method proposed by Abbatematteo et al. [1] and three ablated versions of ScrewNet and show that it outperforms all baselines with a significant margin.

6.1 Problem Formulation

Given a sequence of n depth images $\mathcal{I}_{1:n}$ of motion between two parts of an articulated object, we wish to estimate the articulation model $\mathcal{M}$ and its parameters ϕ governing the motion between the two parts without knowing the articulation model category a priori. Additionally, we wish to estimate the configurations $q_{1:n}$ that uniquely identify different relative spatial displacements between the two parts in the given sequence of images $\mathcal{I}_{1:n}$ under model $\mathcal{M}$ with parameters ϕ. We consider articulation models with at most one degree-of-freedom (DoF), i.e. $\mathcal{M} \in \{\mathcal{M}_{\text{rigid}}, \mathcal{M}_{\text{revolute}}, \mathcal{M}_{\text{prismatic}}, \mathcal{M}_{\text{helical}}\}$. Model parameters ϕ are defined as the parameters of the screw axis of motion, i.e. $\mathsf{S} = (\mathbf{l}, \mathbf{m})$, where both $\mathbf{l}$ and $\mathbf{m}$ are three-dimensional real vectors. Each configuration q_i corresponds to a tuple of two scalars, $q_i = (\theta_i, d_i)$, defining a rotation around and a displacement along the screw axis S. We assume that the relative motion between the two object parts is governed only by a single articulation model.

6.2 ScrewNet

We propose ScrewNet, a novel approach that given a sequence of segmented depth images $\mathcal{I}_{1:n}$ of the relative motion between two rigid objects estimates the articulation model $\mathcal{M}$ between the objects, its parameters ϕ, and the corresponding configurations $q_{1:n}$. In contrast to state-of-the-art approaches, ScrewNet does not require a priori knowledge of the articulation model category for the objects to estimate their models. ScrewNet achieves category independent articulation model estimation by representing different articulation models through a unified representation based on the screw theory [121]. ScrewNet represents the 1-DoF articulation relationships between rigid objects (rigid, revolute, prismatic, and helical) as a sequence of screw displacements along a common screw axis. A rigid model is now defined as a sequence of identity transformations, i.e., $\theta_{1:n} = 0 \wedge d_{1:n} = 0$, a revolute model as a sequence of pure rotations around a common axis, i.e., $\theta_{1:n} \neq 0 \wedge d_{1:n} = 0$, a prismatic model as a sequence of pure displacements along the same axis, i.e., $\theta_{1:n} = 0 \wedge d_{1:n} \neq 0$, and, a helical model as a sequence of correlated rotations and displacements along a shared axis, i.e., $\theta_{1:n} \neq 0 \wedge d_{1:n} \neq 0$).

Under this unified representation, all 1-DoF articulation models can be represented using the same number of parameters, i.e., 6 parameters for the common screw axis S and $2n = |\{(\theta_i, d_i) \forall i \in \{1...n\}\}|$ parameters for configurations, which enables ScrewNet to perform category independent articulation model estimation. ScrewNet not only estimates the articulation model parameters without requiring the model category $\mathcal{M}$, but is also capable of estimating the category itself. This ability can potentially reduce the number of control parameters required for manipulating the object [56]. A unified representation also allows ScrewNet to use a single network to estimate the articulation motion models across cate-

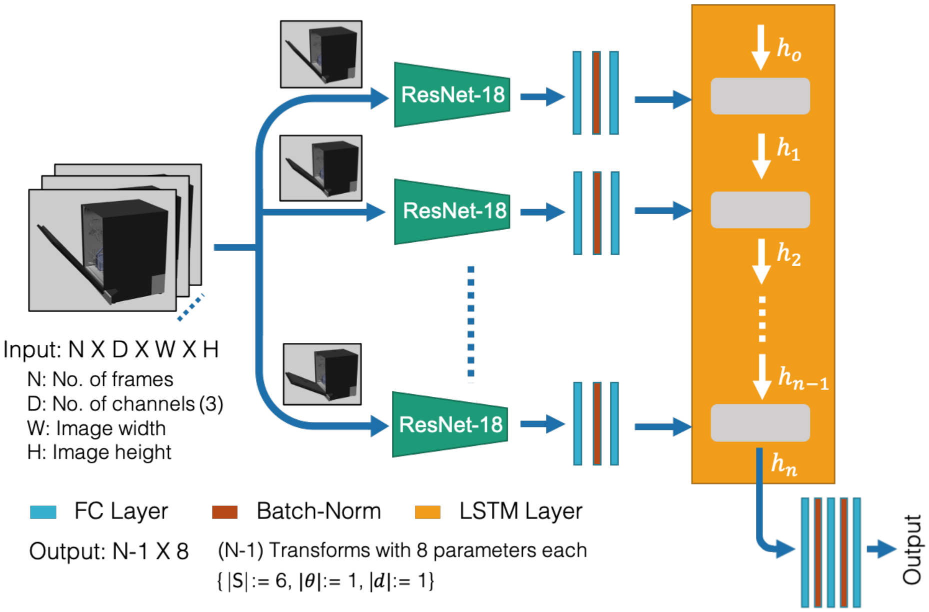


Figure 6.2: Taking a sequence of depth images as input, ScrewNet first extracts features from the depth images using ResNet, passes them through an LSTM layer to encode their sequential information, and then uses MLP to predict a sequence of screw displacements having a shared screw axis

gories, unlike prior approaches that required separate networks, one for each articulation model category [1, 72]. Having a single network grants ScrewNet two major benefits: first, it needs to train fewer total parameters, and second, it allows for a greater sharing of training data across articulation categories, resulting in a significant increase in the number of training examples that the network can use. Additionally, in theory, ScrewNet can also estimate an additional articulation model category, the helical model, which was not addressed in earlier work [1, 86, 131].

6.2.1 Architecture

ScrewNet sequentially connects a ResNet-18 CNN [45], an LSTM with one hidden layer, and a 3-Layer MLP. ResNet-18 extracts features from the depth images that are fed into the LSTM which encodes the sequential information from the extracted features into a latent representation. Using this representation, the MLP then predicts a sequence of screw displacements having a common screw axis. The network is trained end-to-end, with ReLU activations for the fully-connected layers. Fig. C.1 shows the network architecture. The model category $\mathcal{M}$ is deduced from the predicted screw displacements using a decision-tree based on the displacements properties of each model class.

6.2.2 Loss function

Screw displacements are composed of two major components: the screw axis S, and the corresponding configurations q_i about it. Hence, we pose ScrewNet training as a multi-objective optimization problem with loss

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathsf{S}_{\text{ori}}} + \lambda_2 \mathcal{L}_{\mathsf{S}_{\text{dist}}} + \lambda_3 \mathcal{L}_{\mathsf{S}_{\text{cons}}} + \lambda_4 \mathcal{L}_{\text{q}}, \tag{6.1}$$

where λ_i weights the respective component. $\mathcal{L}_{\mathsf{S}_{\text{ori}}}$ penalizes the screw axis orientation mismatch as the angular difference between the target and the predicted orientations. $\mathcal{L}_{\mathsf{S}_{\text{dist}}}$ penalizes the spatial distance between the target and predicted screw axes as defined in Eqn. 3.6. $\mathcal{L}_{\mathsf{S}_{\text{cons}}}$ enforces the constraints $\langle \mathbf{l}, \mathbf{m} \rangle = 0$ and $\|\mathbf{l}\| = 1$. $\mathcal{L}_{\text{q}} := \alpha_1 \mathcal{L}_{\theta} + \alpha_2 \mathcal{L}_{d}$ penalizes errors in the configurations, where $\mathcal{L}_{\theta}$ and $\mathcal{L}_{d}$ represent the rotational and translational error respectively:

$$\begin{aligned} \mathcal{L}_{\theta} &= \mathbf{I}_{3,3} - R(\theta_{\text{tar}}; \mathbf{l}_{\text{tar}}) \; R(\theta_{\text{pred}}; \mathbf{l}_{\text{pred}})^T, \\ \mathcal{L}_{d} &= \| d_{\text{tar}} \cdot \mathbf{l}_{\text{tar}} - d_{\text{pred}} \cdot \mathbf{l}_{\text{pred}} \| \end{aligned} \tag{6.2}$$

with $R(\theta; \mathbf{l})$ denoting the rotation matrix corresponding to a rotation of angle θ about the axis $\mathbf{l}$. We choose this particular form of the loss function for $\mathcal{L}_{\mathrm{q}}$, rather than a standard loss function such as an $L2$ loss, as it ensures that the network predictions are grounded in their physical meaning. By imposing a loss based on the orthonormal property of the $3D$ rotations, the proposed loss function ensures that the learned angle-axis pair $(\mathbf{l}_{\mathrm{pred}}, \theta_{\mathrm{pred}})$ corresponds to a rotation $R(\theta_{\mathrm{tar}}; \mathbf{l}_{\mathrm{tar}}) \in SO(3)$. Similarly, the loss function $\mathcal{L}_d$ calculates the difference between the two displacements along two different axes $\mathbf{l}_{\mathrm{tar}}$ and $\mathbf{l}_{\mathrm{pred}}$, rather than calculating the difference between the two configurations, d_{tar} and d_{pred}, which assumes that they represent displacements along the same axis. Hence, this choice of loss function ensures that the network predictions conform to the definition of a screw displacement. We empirically choose weights to be $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 1, \lambda_4 = 1, \alpha_1 = 1$, and $\alpha_2 = 1$.

6.2.3 Training data generation

The training data consists of sequences of depth images of objects moving relative to each other and the corresponding screw displacements. The objects and depth images are rendered in Mujoco [139]. We apply random frame skipping and pixel dropping to simulate noise encountered in real world sensor data. We use the cabinet, drawer, microwave, the toaster-oven object classes from the simulated articulated object dataset[1]. The cabinet, microwave, and toaster object classes contain a revolute joint each, while the drawer class contains a prismatic joint. We consider both left-opening cabinets and right-opening cabinets. From the PartNet-Mobility dataset [20, 93, 149], we consider the dishwasher, oven, and microwave object classes for the revolute articulation model category, and the storage furniture object class consisting of either a single column of drawers or multiple columns of drawers, for the prismatic articulation model category.

To generate the labels for screw displacements, we consider one of the objects, o_i, as the base object, and calculate the screw displacements between temporally displaced poses of the second object o_j with respect to it, as illustrated in Fig. 6.3. Given a sequence of n images $\mathcal{I}_{1:n}$, we calculate a sequence of $n-1$ screw displacements ${}^1\boldsymbol{\sigma}_{o_j} = \{{}^1\sigma_2, \ldots {}^1\sigma_n\}$, where each ${}^1\sigma_k$ corresponds to the relative spatial displacement between the pose of the object o_j in the first image $\mathcal{I}_1$ and the images $\mathcal{I}_{k,\ k\in\{2\ldots n\}}$. Note ${}^1\boldsymbol{\sigma}_{o_j}$ is defined in the frame $\mathcal{F}_{o_j^1}$ attached to the pose of the object o_j in the first image $\mathcal{I}_1$. We can transform ${}^1\boldsymbol{\sigma}_{o_j}$ to a frame attached to the base object $\mathcal{F}_{o_i}$ by defining the 3D line motion matrix $\tilde{D}$ (Eqn. 3.7) between the frames $\mathcal{F}_{o_j^1}$ and $\mathcal{F}_{o_i}$ [8], and transforming the common screw axis ${}^1\mathsf{S}$ to the target frame $\mathcal{F}_{o_i}$. The configurations ${}^1q_k$ remain the same during frame transformations.

6.3 Experiments

We evaluated ScrewNet's performance in estimating the articulation models for objects by conducting three sets of experiments on two benchmarking datasets: the simulated articulated objects dataset provided by Abbatematteo et al. [1] and the recently proposed PartNet-Mobility dataset [20, 93, 149]. The first set of experiments evaluated ScrewNet's performance in estimating the articulation models for unseen object instances that belong to the object classes used for training the network. Next, we tested ScrewNet's performance in estimating the model parameters for novel articulated objects that belong to the same articulation model category as seen during training. In the third set of experiments, we trained a single ScrewNet on object instances belonging to different object classes and articulation model categories and evaluated its performance in cross-category articulation model

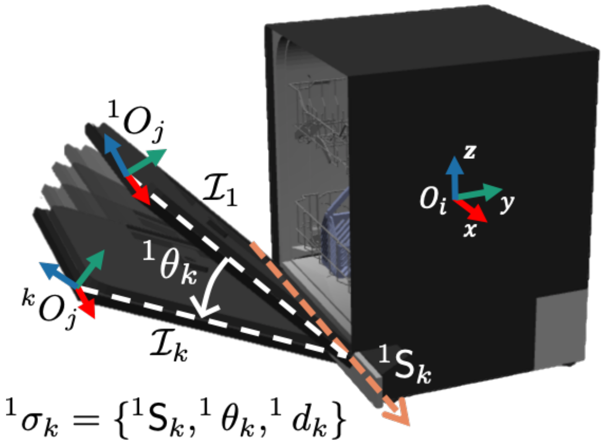


Figure 6.3: The training labels are generated by calculating the screw displacements between the temporally displaced poses of the object o_j, and expressing them in a frame of reference attached to the base object o_i

estimation. We compared ScrewNet with a state-of-the-art articulation model estimation method proposed by Abbatematteo et al. [1]. Lastly, to evaluate how effectively ScrewNet transfers from simulation to real-world setting, we trained ScrewNet solely using simulated images and tested it to estimate articulation models for three real-world objects.

In all the experiments, we assumed that the input depth images are semantically segmented and contain non-zero pixels corresponding only to the two objects between which we wish to estimate the articulation model. Given this input, ScrewNet estimates the articulation model parameters for the pair of objects in an object-centric coordinate frame defined at the center of the bounding box of the object. Note while the approach proposed by Abbatematteo et al. [1] can be used to estimate the articulation model parameters directly in the camera frame, for a fair comparison to our approach, we modified the baseline to predict the model parameters in the object-centric reference frame as well.

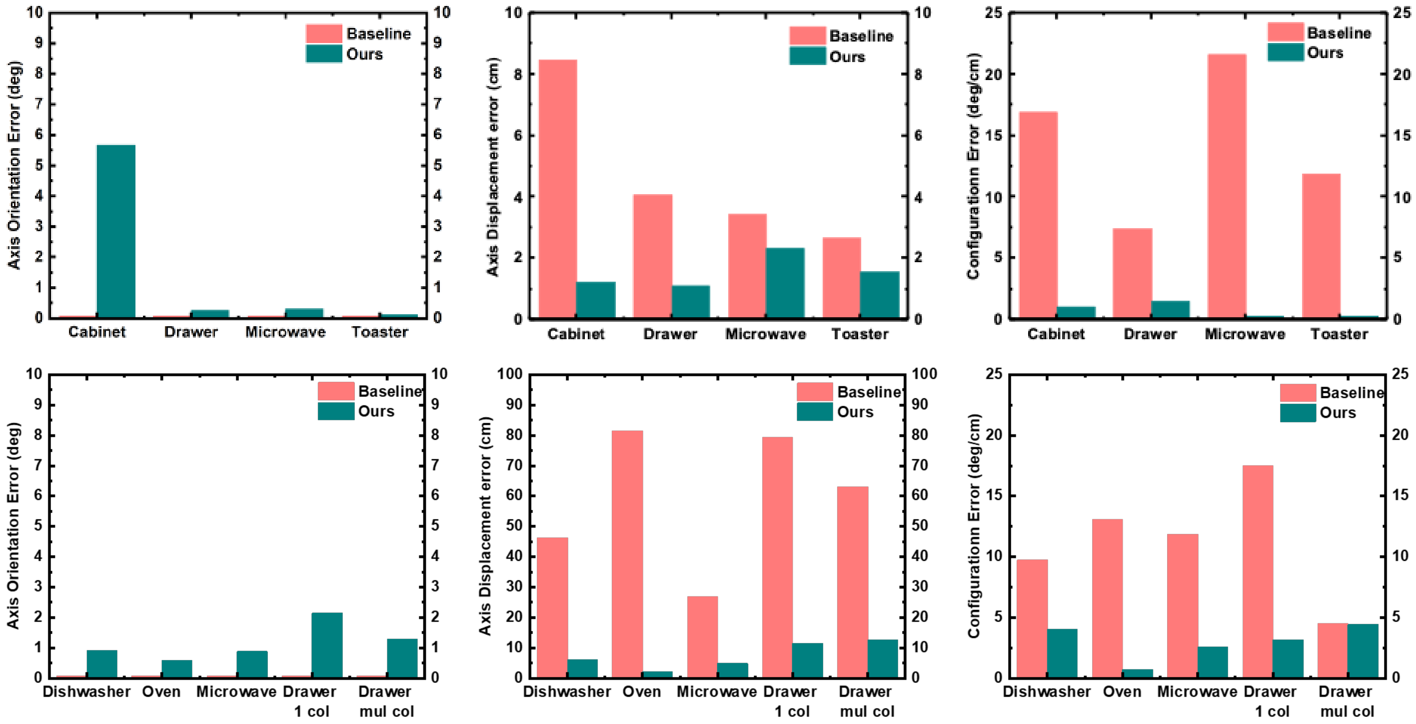


Figure 6.4: [**Same object class**] Mean error values for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from **(top)** the simulated articulated objects dataset [1] and **(bottom)** PartNet-Mobility Dataset [20, 93, 149]. Configuration errors for all drawers are in cm the remaining configuration errors are in degrees.

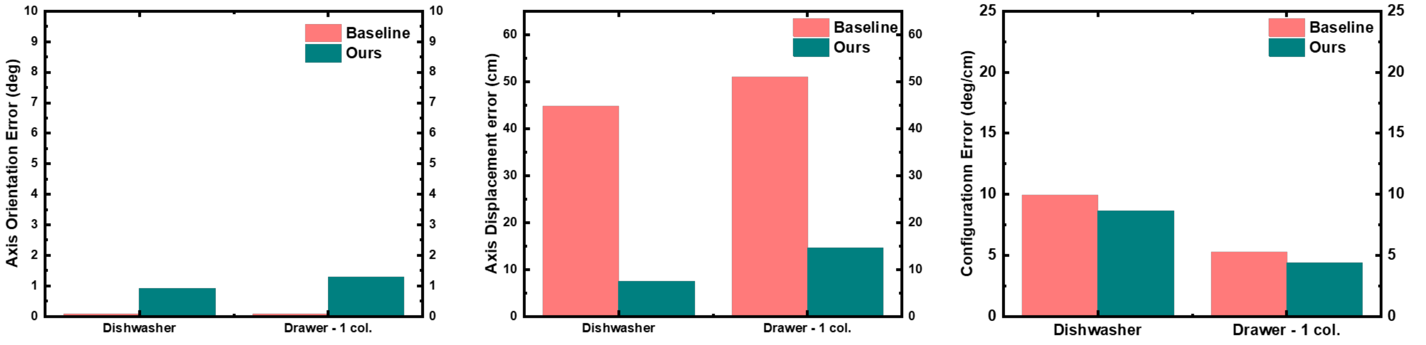


Figure 6.5: [**Same articulation model category**] Mean errors for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from the PartNet-Mobility Dataset.

6.3.1 Same object class

In the first set of experiments, we investigated whether our proposed approach can generalize to unseen object instances belonging to the object classes seen during the training. For this set of experiments, we trained a separate ScrewNet and a baseline network [1] for each of the object classes and tested how ScrewNet fares in comparison to the baseline under similar experimental conditions. We generated 10,000 training examples for each object class in both datasets and performed evaluations on 1,000 withheld object instances. From Fig. 6.4, it is evident that ScrewNet outperformed the baseline in estimating the joint axis position and the observed joint configurations by a significant margin for the first dataset. However, for the joint axis orientation estimation, the baseline method reported lower errors than the ScrewNet. Similar trends in the performance of the two methods were observed on the PartNet-Mobility dataset (see Fig. 6.4). ScrewNet significantly outperformed the baseline method in estimating the joint axis displacement and observed joint configurations, while the baseline reported lower errors than ScrewNet in estimating the joint axis orientations. However, for both the datasets, the errors reported by ScrewNet in screw axis orientation estimation were reasonably low ($< 5°$), and the model parameters predicted by ScrewNet may be used directly for manipulating the object. These experiments demonstrate that under similar experimental conditions, ScrewNet can estimate the joint axis positions and joint configurations for objects better than the baseline method, while reporting reasonably low but higher errors in joint axis orientations.

6.3.2 Same articulation model category

Next, we investigated if our proposed approach can generalize to unseen object classes belonging to the same articulation model category. We conducted this set of experiments only on the PartNet-Mobility dataset as the simulated articulated objects dataset does not contain enough variety of object classes belonging to the same articulation model category (only 3 for revolute and 1 for prismatic). For the revolute category, we trained ScrewNet and the baseline on the object instances generated from the oven and the microwave object classes and tested it on the objects from the dishwasher class. For the prismatic category, we trained them on the objects from the storage furniture class containing multiples columns of drawers and tested it on the storage furniture objects containing a single column of drawers. We trained a single instance of ScrewNet and the baseline for each articulation model category and used them to predict the articulation model parameters for the test object classes. We used the same training datasets as used in the previous set of experiments. Results are reported in Fig. 6.5. It is evident from Fig. 6.6 that ScrewNet was able to generalize to novel object classes belonging to the same articulation model category, while the baseline failed to do so. Both methods reported low errors in the joint axis orientation and the observed configurations. However, for the joint axis position, the baseline method reported mean errors of an order of magnitude higher than the ScrewNet for both the articulation model categories.

6.3.3 Across articulation model category

Next, we studied whether ScrewNet can estimate articulation model parameters for unseen objects across the articulation model category. For these experiments, we trained

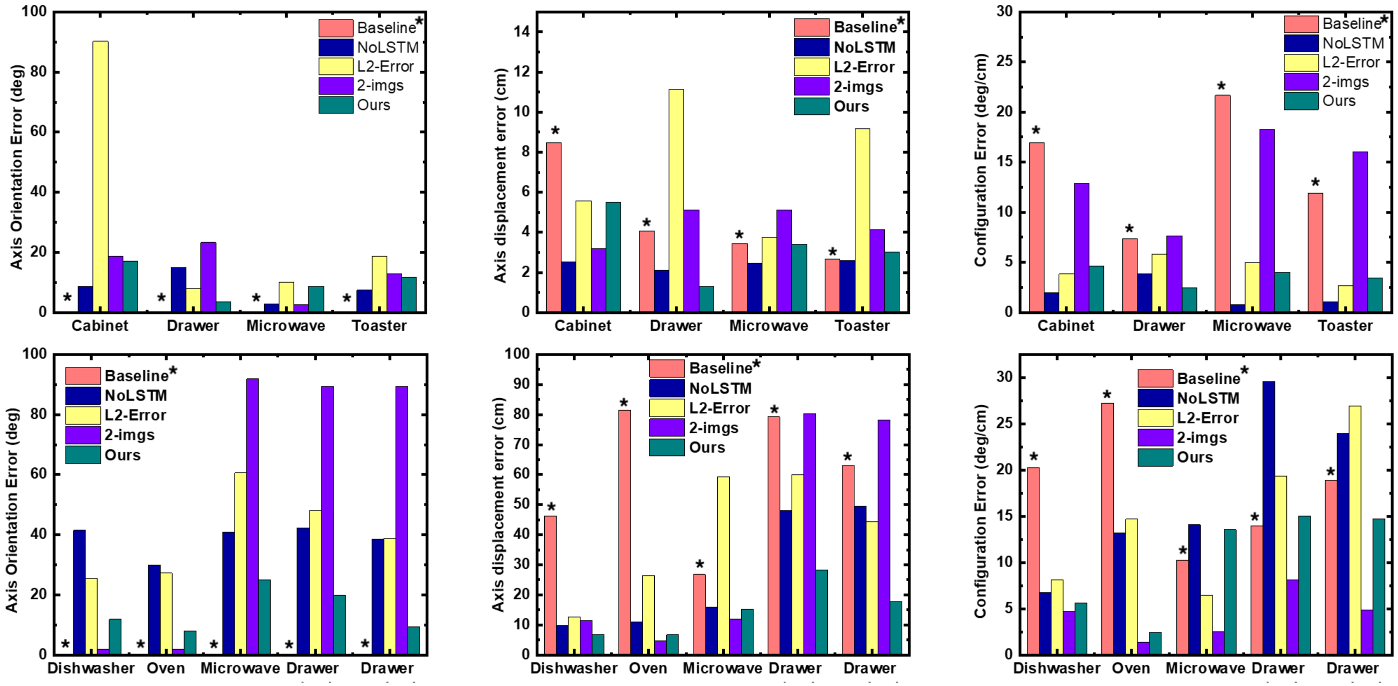


Figure 6.6: [**Across articulation model category**] Mean error values for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from **(top)** the simulated articulated objects dataset [1] and **(bottom)** PartNet-Mobility Dataset [20, 93, 149]. Symbol $\star$ denote that the baseline has a significant advantage over other methods as it uses a separate network for each object class

a single ScrewNet on a mixed dataset consisting of object instances belonging to all object model classes. To test whether sharing training data across articulation categories can help in reducing the number of examples required for training, we used only half of the dataset available for each object class (5000 examples each) while preparing the mixed dataset. We compared its performance with a baseline network that is trained specifically on the particular object class. We also conducted ablation studies to test the effectiveness of the various components of the proposed method.

Fig. 6.6 summarizes the results for the first dataset. Even though we used a single network to estimate the articulation model for objects belonging to different articulation model categories, ScrewNet performed at par or better than the baseline method for all the object model categories. ScrewNet outperformed the baseline while estimating the observed joint configurations for all object classes, even though the baseline was trained separately for each object class. For joint axis position estimation, ScrewNet reported significantly lower errors than the baseline for the cabinet and the drawer classes, and comparable errors for the microwave and the toaster classes. In estimating the joint axis orientations, both methods reported comparable errors for the cabinet, drawer, and the toaster classes. However, for the cabinet object class, ScrewNet reported a higher error than the baseline method, which may stem from the fact that the cabinet object class includes both left-opening and right-opening configurations that have a difference 180° in their axis orientations. On the PartNet-Mobility dataset (see Fig. 6.6), the performances of the methods was similar, with ScrewNet outperforming the baseline method with a significant margin in estimating the joint axis positions and the observed joint configurations while reporting higher errors than the baseline in estimating the joint axis orientations. The results show ScrewNet leverages the unified

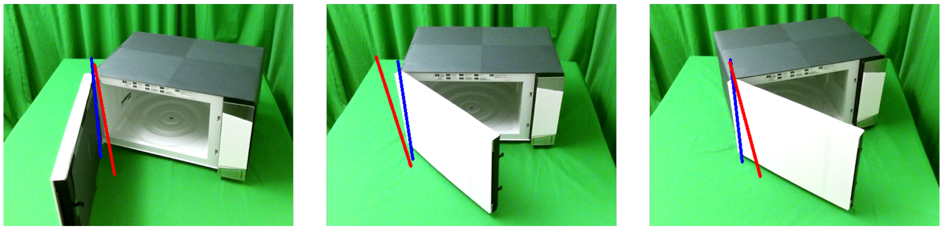


Figure 6.7: [**Real-world images**] Images with overlayed ground-truth joint axis (blue) and predicted axis (red) for different poses of the microwave

representation and performs cross-category articulation model estimation with better on average performance than the current state-of-the-art method while using only half the training examples.

In comparison to its ablated versions, ScrewNet outperformed the L2-error and the two-images versions by a significant margin for both datasets and performed comparably to the NoLSTM version. For the first dataset, the NoLSTM version reported lower errors than ScrewNet in estimating the joint axis orientations, their positions, and the observed joint configurations for the microwave, cabinet, and toaster classes. However, the NoLSTM version failed to generalize across articulation model categories and reported higher errors than the ScrewNet for the drawer class, and sometimes even predicted NaNs. On the second dataset, ScrewNet reported much lower errors than the NoLSTM ablated version for all object model categories. These results demonstrate that for reliably estimating articulation model parameters across categories, both the sequential information available in the input and a loss function that grounds predictions in their physical meaning are crucial.

6.3.4 Real world images

Finally, we evaluated how effectively ScrewNet transfers from simulation to a real-world setting. ScrewNet was trained solely on the combined simulated articulated object dataset. Afterwards, we used the model to infer the joint axis of a microwave, a drawer, and a toaster oven. Figure 6.7 qualitatively demonstrates ScrewNet's performance for three different poses of the microwave. Despite only ever having seen simulated data, ScrewNet achieved a mean error of $\sim 20^{\circ}$ in axis orientation and ~ 1.5cm in axis position on real-world sensor input. These results demonstrate that ScrewNet achieves reasonable estimates of the articulation model parameters for real-world objects when it is trained solely using simulated data. In order to obtain better performances a retraining on real world data would be required.

6.4 Conclusion

Articulated objects are common in human environments and service robots will be interacting with them frequently while assisting humans. For manipulating such objects, a robot will need to learn the articulation properties of such objects through raw sensory data such as RGB-D images. Current methods for estimating the articulation model of objects from visual observations either require textured objects or need to know the articulation model category a priori for estimating the articulation model parameters from the depth images. We present ScrewNet that uses screw theory to unify the representation of different articulation models and performs category-independent articulation model estimation from depth images. We evaluate the performance of ScrewNet on two benchmarking datasets and compare it with a state-of-the-art method. Results show that ScrewNet can estimate articulation models and their parameters for objects across object classes and articulation model categories successfully with better on average performance than the baselines while using half the training data and without requiring to know the model category.

While ScrewNet successfully performs cross-category articulation model estimation, it only predicts point estimates for an object's articulation model parameters. Point estimates alone are insufficient to guarantee safe interactions with an articulated object as the robot lacks enough information to discern the accuracy of the estimated parameters. In the next chapter, we introduce DUST-net, a method that learns distributions over articulation models representing the network's confidence over the estimated model parameters and hence is indicative of the accuracy of the estimated parameters. Analogous to ScrewNet, DUST-net takes raw depth images as input and does not need to know the articulation model categories for objects a priori to learn such distributions.

Chapter 7

Distributional Depth-Based Estimation of Object Articulation Models

Service robots will need to interact with articulated objects such as drawers, refrigerators, and dishwashers, often while assisting humans in performing day-to-day tasks. To interact safely with such objects, a robot must reason about their articulation properties while manipulating them. An ideal method for learning such properties might estimate these parameters directly from raw observations, such as RGB-D images while requiring limited or no a priori information about the task. The ability to additionally provide confidence over the estimated properties would allow such a method to be leveraged for developing safe manipulation policies for interacting with articulated objects [55]. Knowing the estimator's confidence over the estimated properties will also enable the robot to use methods that can increase its chances of task success, such as active learning based methods for information gathering [23] and behavior policy learners that provide safety guarantees [137].

The majority of existing methods to learn articulation models for objects from visual data either need fiducial markers to track motion between object parts [63, 64, 100, 131] or require textured objects [56, 85–87, 111]. Recent deep-learning based methods address this by predicting articulation properties for objects from raw observations, such as depth images [1, 58, 72, 75] or PointCloud data [147, 153]. However, the majority of these methods [1, 72, 147, 153] require knowledge of the articulation model category for the object (e.g., whether it has a revolute or prismatic joint) which may not be available in many realistic settings. Alleviating this requirement, Jain et al. [58] introduced ScrewNet, which uses a unified representation based on screw transformations to represent different articulation types and performs category-independent articulation model estimation directly from raw depth images. However, ScrewNet [58] and related methods [1, 72, 75, 147, 153] only predict point estimates for an object's articulation model parameters. Nonetheless, reasoning about the uncertainty in the estimated parameters can provide significant advantages for ensuring success in robot manipulation tasks, and allows for further advancements such as robust planning [55], active learning using human queries [23], and the learning of behavior policies that provide safety assurances [137]. Motivated by these advantages, we propose a method for learning articulation models, which estimates the uncertainty over model parameters using a novel distribution over the set of screw transformations based on the matrix von Mises-Fisher distribution over Stiefel manifolds [22]. We introduce DUST-net, **D**eep **U**ncertainty estimation on **S**crew **T**ransforms-**net**work, a novel deep learning-based method that, in addition to providing point estimates of the object's articulation model parameters, leverages raw depth images to provide uncertainty estimates that can be used to guide the robot's behavior without requiring to knowledge of the object's articulation model category a priori.

DUST-net garners numerous benefits over existing methods. First, DUST-net estimates articulation properties for objects with uncertainty estimates, unlike most current methods [1, 58, 72, 75, 147, 153]. These uncertainty estimates, apart from helping robots to manipulate objects safely [55], could allow a robot to take information-gathering actions when it is not confident and enhance its chances of success in completing the task. Second, similar to ScrewNet [58], DUST-net can estimate model parameters without the need to to know the articulation model category a priori, by leveraging the unified representation for different articulation model types. Third, this unified representation helps DUST-net to be more computationally and data-efficient than other state-of-the-art methods [1, 72], as it uses a single network to estimate model parameters for all common articulation models, unlike other methods that require a separate network for each articulation model category [1, 72, 147, 153]. Empirically, DUST-net outperforms other methods even when trained using only half the training data in comparison. Fourth, the distributional learning setting yields more robustness to outliers and noise. Fifth, DUST-net is able to reliably estimate distributions over articulation model parameters for objects in the robot's camera frame. By contrast, ScrewNet [58], the most closely related approach to ours, can only predict point estimates for articulation model parameters in the object's local frame.

We evaluate DUST-net through experiments on two benchmarking datasets: a simulated articulated objects dataset [1] and the PartNet-Mobility dataset [20, 93, 149], as well as three real-world objects: a microwave, a drawer, and a toaster oven. We compare DUST-net with two state-of-the-art methods, namely ScrewNet [58] and an MDN-based method proposed by Abbatematteo et al. [1], as well as two baseline methods. The experiments demonstrate that the samples drawn from the distributions learned by DUST-net result in significantly better estimates for articulation model parameters in comparison to the point estimates predicted by other methods. Additionally, the experiments show that DUST-net can successfully and accurately capture the uncertainty over articulation model parameters resulting from noisy inputs.

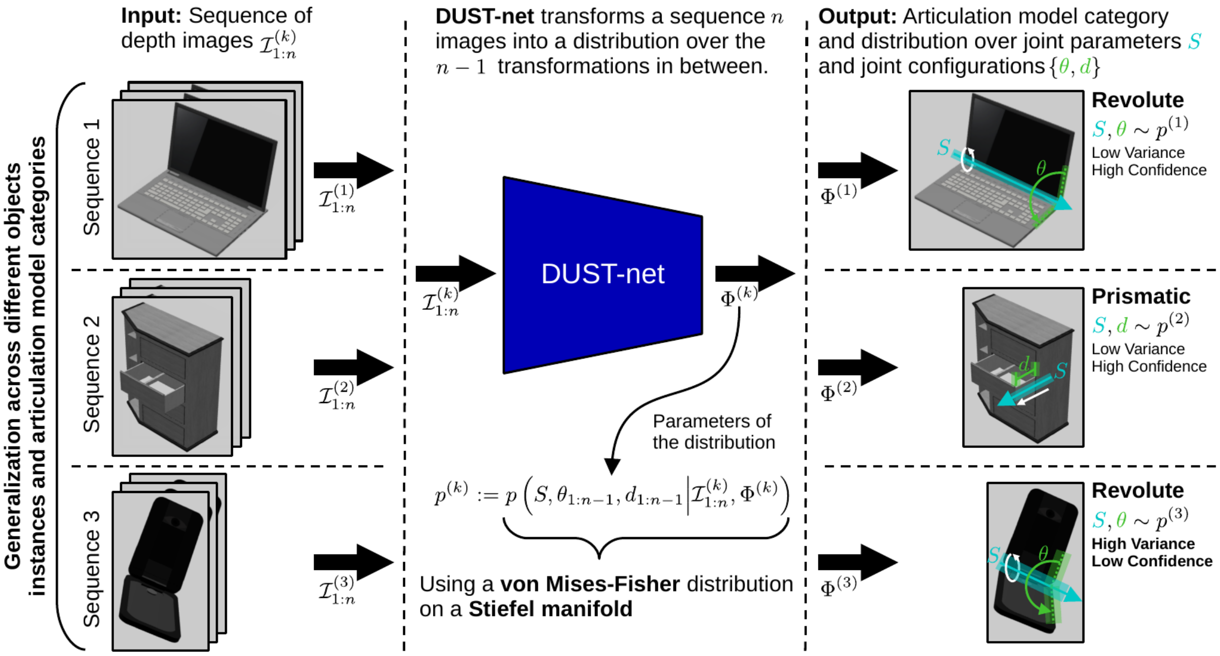


Figure 7.1: DUST-net uses a sequence of images $\mathcal{I}_{1:n}$ to compute the parameters, Φ, of the conditional distribution over the joint parameters S and configurations $\{\theta, d\}_{1:n-1}$. This distribution allows for inference and reasoning, such as uncertainty and confidence, over both the parameters and the configurations. Using a von Mises-Fisher distribution on a Stiefel manifold allows for an efficient reparameterization that inherently obeys multiple constraints that define rigid body transformations.

7.1 Problem Formulation

Given a sequence of n depth images $\mathcal{I}_{1:n}$ of motion between two parts of an articulated object, we estimate the parameters of a probability distribution $p(\phi|\mathcal{I}_{1:n})$ representing uncertainty over the parameters ϕ of the articulation model $\mathcal{M}$ governing the motion between the two parts. Following Jain et al. [58], we define the model parameters ϕ as the parameters of the screw axis of motion, $\mathsf{S} = (\mathbf{l}, \mathbf{m})$, where both $\mathbf{l}$ and $\mathbf{m}$ are elements of $\mathbb{R}^3$. This unified parameterization can be used in articulation models with at most one degree-of-freedom

(DoF), namely rigid, revolute, prismatic, and prismatic [58]. Additionally, we estimate the parameters of a distribution $p(q_{1:n-1}|\mathcal{I}_{1:n})$ representing uncertainty over the configurations $q_{1:n-1}$ identifying the rigid body transformations between the two parts in the given sequence of images $\mathcal{I}_{1:n}$ under model $\mathcal{M}$ with parameters ϕ. Configurations $q_i, i \in \{1...n-1\}$ correspond to a set of tuples, $q_i = (\theta_i, d_i)$, defining a rotation around and a displacement along the screw axis S. We assume that the relative motion between the two object parts is determined by a single articulation model.

7.2 DUST-net

Given a sequence of depth images $\mathcal{I}_{1:n}$ of motion between two parts of an articulated object, DUST-net estimates parameters of the joint probability distribution $p(\phi, q_{1:n-1}|\mathcal{I}_{1:n})$ representing uncertainty over the articulation model parameters ϕ governing the motion between the two parts and the observed configurations $q_{1:n-1}$. When deciding how to learn this distribution, two goals arise. While some parameters, such as the translation of an object part along a screw axis, are defined on Euclidean space, the set of valid screw axes exhibits constraints that prevent standard distributions defined on $\mathbb{R}^6$ from being applied without complicating the learning process. For example, a standard representation for distributions over screw axes is as the product of a Bingham distribution over the line's orientation and a multivariate normal distribution over its position in space [121]. However, this representation produces non-unique estimation targets. A rotation of θ about the screw axis with orientation $\mathbf{l}$ results in the same transformation as a rotation of $-\theta$ about the screw axis with orientation $-\mathbf{l}$. Similarly, a displacement d along $\mathbf{l}$ results in the same transformation as a displacement $-d$ along $-\mathbf{l}$. This leads to ambiguities in the targets in the estimation problem and can hinder the performance of the trained estimator. By selecting a representation that accounts for these symmetries, these non-unique estimation targets are removed. Second, once a suitable parameterization is chosen, we seek a parametric form for the joint distribution which can be learned by a deep network.

First, we consider the problem of parameterizing the set of screw axes. As noted earlier, we define the model parameter ϕ as the parameters of the screw axis of motion $\mathsf{S} = (\mathbf{l}, \mathbf{m})$. However, this parameterization requires that $\mathbf{l}$ has unit norm, and that $\mathbf{l}$ and $\mathbf{m}$ are orthogonal. To eliminate these constraints, we rewrite the moment vector of a screw axis as $\mathbf{m} = \|\mathbf{m}\|\hat{\mathbf{m}}$, where $\|\mathbf{m}\|$ and $\hat{\mathbf{m}}$ represent its magnitude and a unit vector along it respectively, and the Plücker coordinates for the screw axis as $\mathsf{S} = (\mathbf{l}, \hat{\mathbf{m}}, \|\mathbf{m}\|)$. The Plücker coordinates can then be seen as an unconstrained point in the space $\mathbb{S} := \mathrm{V}_{2,3} \times \mathbb{R}^{+}$, where $(\mathbf{l}, \hat{\mathbf{m}}) \in \mathrm{V}_{2,3}$ and $\|\mathbf{m}\| \in \mathbb{R}^{+}$. Consequently, a distribution over screw transformations can be defined as the product of any probability distribution defined on $\mathrm{V}_{2,3}$ and a distribution over $\mathbb{R}^{+}$. Importantly, because of the one-to-one mapping from elements of $\mathrm{V}_{2,3} \times \mathbb{R}^{+}$ to screw axes, the non-unique estimation targets described above are eliminated. Based on this parametrization of screw axes, we define the set of valid configuration parameters as follows. We restrict the range of values for the rotation about the screw axis to be $\theta \in [0, 2\pi)$ and restrict the displacement along the axis to be $d \in \mathbb{R}^{+}$. Note that these constraints do not reduce the representational power of the screw transform $(\mathbf{l}, \mathbf{m}, \theta, d)$ to denote a general rigid body transform, but merely ensure a unique representation.

Having described the parameterization of the set of screw axes and configurations, we now consider the task of defining a joint probability distribution over their values. We propose to represent the distribution over predicted screw axis parameters, $p(\mathsf{S} \mid \mathcal{I}_{1:n})$ whit

$\mathsf{S} \in \mathbb{S}$, as a product of a matrix von Mises-Fisher distribution $\mathcal{F}(\cdot|3, \mathbf{F})$ defined on the Stiefel manifold $\mathrm{V}_{2,3}$ and a truncated normal distribution $\mathcal{N}^+(\cdot|\mu, \sigma)$ with truncation interval $[0, +\infty)$ over $\mathbb{R}^+$. Formally,

$$p(\mathsf{S} \mid \mathcal{I}_{1:n}) = p\left(l, \hat{\mathbf{m}}, \|\mathbf{m}\| \;\middle|\; \mathcal{I}_{1:n}, \mathbf{F}, \mu_{\mathbf{m}}, \sigma_{\mathbf{m}}^2\right) = \mathcal{F}\left(\mathbf{l}, \hat{\mathbf{m}} \mid 3, \mathbf{F}\right)\ \mathcal{N}^+\left(\|\mathbf{m}\| \mid \mu_{\mathbf{m}}, \sigma_{\mathbf{m}}^2\right), \quad (7.1)$$

where $\mathbf{F}$ is a 3×2 matrix representing the parameters of the matrix von Mises-Fisher distribution over $\mathrm{V}_{2,3}$, and $\mu_{\mathbf{m}}$ and $\sigma_{\mathbf{m}}$ denote the mean and standard deviation of the truncated normal distribution.

Given the sequence of n images, we also wish to estimate the posterior over configurations $q_{1:n-1} = \{\theta_{1:n-1}, d_{1:n-1}\}$ corresponding to the rotations about and displacements along the screw axis S. We define the joint posterior representing the uncertainty over the screw axis S and the configurations $\{\theta_{1:n-1}, d_{1:n-1}\}$ about it as a product of the aforementioned distribution and a set of distributions defined over the configuration parameters,

$$p(\mathsf{S}, \theta_{1:n-1}, d_{1:n-1} \mid \mathcal{I}_{1:n}, \Phi) = p(\mathsf{S}; \mathbf{F}, \mu_{\mathbf{m}}, \sigma_{\mathbf{m}}^2)\ \Psi(\theta_{1:n-1}; \psi)\ \Upsilon(d_{1:n-1}; \upsilon) \quad (7.2)$$

where $\Phi = \{\mathbf{F}, \mu_{\mathbf{m}}, \sigma_{\mathbf{m}}^2, \psi, \upsilon\}$ is the set of parameters for the distribution and Ψ and Υ represent the set of distributions having parameters ψ and υ over the configurations $\theta_{1:n-1}$ and $d_{1:n-1}$, respectively. In this work, we consider Ψ and Υ to be products of truncated normal distributions such that $\Psi = \prod_{i=1}^{n-1} \mathcal{N}^+(\theta_i|\mathrm{M}_\theta^i, \sigma_\theta^2)$ and $\Upsilon = \prod_{i=1}^{n-1} \mathcal{N}^+(d_i|\mathrm{M}_d^i, \sigma_d^2)$ with $\mathrm{M}_\theta = \{\mu_\theta^1, ..., \mu_\theta^{n-1}\}$, $\mathrm{M}_d = \{\mu_d^1, ..., \mu_d^{n-1}\}$, σ_θ, and σ_d denoting the set of means and the standard deviations of the set of truncated normal distributions over the configurations $\theta_{1:n-1}$ and $d_{1:n-1}$, respectively.

The proposed distribution enables DUST-net to enjoy several benefits over the current state-of-the-art methods [1, 58, 72, 75, 147, 153] in estimating articulation model parameters for objects, as noted earlier. First, DUST-net predicts articulation model parameters for objects along with uncertainty estimates over them instead of just point estimates, unlike other methods[1, 58, 72, 75, 147, 153]. Second, due to a unified parameterization for different articulation models, DUST-net can estimate model parameters without needing to know articulation model category a priori unlike other state-of-the-art methods [1, 72]. Third, it is more computationally and data-efficient in comparison to existing methods [1, 58, 72, 75, 147, 153]. Fourth, by representing Plücker coordinates of the screw axis as a point in the space $\mathbb{S}$, DUST-net handles the two constraints, $||\mathbf{l}|| = 1$ and $\langle \mathbf{l}, \mathbf{m} \rangle = 0$, imposed on the Plücker coordinates implicitly. This ensures that the predicted model parameters are always valid. Lastly, due to estimating the model parameters in a distributional learning setting, DUST-net is more robust to noise and outliers in comparison to ScrewNet [58], the most closely related approach to ours. Moreover, due to the distributional representation, DUST-net does not require its input to correspond to a time series and does not need an LSTM layer in its network architecture. This helps DUST-net to be more computationally and data-efficient in comparison to ScrewNet. Both these factors help DUST-net to be able to estimate the articulation model parameters for objects both in the camera frame and in the local frame of objects, unlike ScrewNet which can only do so in the object's local frame.

7.2.1 Distribution Parameter Matrix F

The parameter matrix for the matrix von Mises-Fisher distribution over $\mathrm{V}_{3,2}$ is a 3×2 matrix, $\mathbf{F}$. This presents two possible parameterizations of the matrix: first, to estimate each of the 6 elements of the 3×2 matrix $\mathbf{F}$ and second, to estimate the matrices Γ, Λ, and Ω defining the SVD of $\mathbf{F}$, given by $\mathbf{F} = \Gamma \Lambda \Omega^T$. The second parameterization decouples the two objectives of distribution mode alignment with the ground truth labels and uncertainty representation; the mode of the distribution is given by $M = \Gamma\Omega^T$, and the concentration matrix for the distribution is given by $K = \Omega\Lambda\Omega^T$. This decoupling allows the network to independently optimize both objectives, whereas in the first parameterization, changes in the elements of **F** causes changes in both components.

By definition, Λ is a 2×2 diagonal matrix with two independent parameters, and $\Omega \in O(2)$ is a rotation matrix in two dimensions with one independent parameter, the rotation angle ω. The matrix $\Gamma \in \tilde{\mathrm{V}}_{3,2}$ can be constructed from a rotation matrix $\mathrm{R} \in O(3)$ by keeping only the first two columns of R. Hence, the matrix Γ can be defined by three independent Euler angles, (α, β, γ) denoting rotation according to the ZYX convention in the rotating frame. Euler angles can suffer from the problem of gimble lock [121], which we resolve by restricting the Euler angles to be in the ranges $\alpha \in [0, 2\pi), \beta \in [0, \pi)$, and $\gamma \in [0, 2\pi)$.

7.2.2 Normalization Factor of Matrix von Mises-Fisher Distribution

One of the main challenges of using the matrix von Mises-Fisher distribution is the calculation of its normalization factor ${}_0F_1(\frac{m}{2}, \frac{1}{4}\Lambda^2)$, which is a hypergeometric function of matrix argument [22]. In this work, we approximate this hypergeometric function using a truncated series in terms of zonal polynomials, which are multivariate symmetric homogeneous polynomials and form a basis of the space of symmetric polynomials [22]. Through our preliminary experiments, we found that this truncated series is a good approximation of ${}_0F_1$ as it converges to a finite value, if the singular values of the F, i.e. λ_1 and λ_2 are less than $\lambda_{max} = 50$.

7.2.3 Architecture

DUST-net sequentially connects a ResNet-18 CNN [45] and a 2-layer MLP. ResNet-18 extracts task-relevant features from the input images, which are used by the MLP to predict a set of parameters Φ for the distribution $p(\mathsf{S}, \theta_{1:n-1}, d_{1:n-1} \mid \mathcal{I}_{1:n}, \Phi)$. The network is trained end-to-end, with ReLU activations for the hidden fully-connected layers. The first four output (out of 40) of the last linear layer of MLP, corresponding to the parameters (α, β, γ) and ω representing the matrices Γ and Ω respectively, are fed through a ReLU-6 layer to ensure that the predictions map to their respective ranges. Remaining output is fed through a Softplus layer to ensure that the predicted parameters are non-negative.

7.2.4 Training

The training data for the model consists of sequences of depth images of objects parts moving relative to each other and the corresponding screw transforms $\mathbf{y} = (l, \hat{\mathbf{m}}, \|\mathbf{m}\|, \theta_{1:n-1}, d_{1:n-1})$. The objects and depth images are rendered in Mujoco [139]. We train DUST-net by maximizing the log-probability of the labels $\mathbf{y}$ under the distribution $p(\mathbf{y} \mid \mathcal{I}_{1:n}, \Phi)$:

$$\mathcal{L}(\mathbf{y}, \Phi) = -\log p(\mathbf{y} \mid \Phi) \tag{7.3}$$

We assume that the observed configurations in $\mathcal{I}_{1:n}$ share the same variance. We use the precision parameters rather than the standard deviations, $\sigma_{\mathbf{m}}, \sigma_\theta$ and σ_d to represent the distribution during training for better numerical stability. Following the discussion on training MDNs by Makansi et al. [82], we separate the training in three stages. In the first stage, we assume the dispersion of the matrix von Mises-Fisher distribution to be fixed with $\Lambda = \mathrm{diag}(\lambda_0, \lambda_0),\ \lambda_0 = 1$ and learn parameters corresponding to Γ and Ω matrices. In the

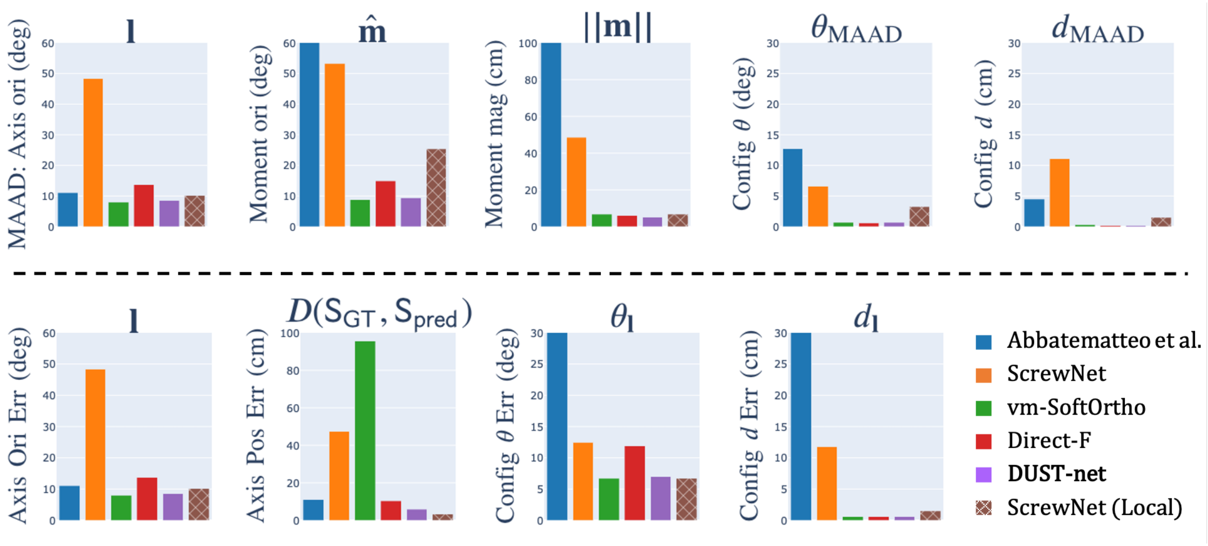


Figure 7.2: Mean error values on the MAAD (top) and Screw Loss (bottom) metrics for the simulated articulated objects dataset [1] (lower values are better). Point estimates for DUST-net (violet) correspond to the modes of the distributions predicted by DUST-net.

second stage, we fix the Λ matrix and learn the rest of the parameters in the set Φ. Finally, we train to predict the complete set Φ.

7.3 Experiments

In this section, we evaluate DUST-net on its ability to learn articulation model parameters and uncertainty estimates. We conducted three sets of experiments evaluating DUST-net's performance under different criteria: (1) how accurate point estimates of the articulation model parameters drawn from DUST-net's estimated distribution are in comparison to the existing methods, (2) how effectively DUST-net captures the uncertainty over parameters arising from noisy input, and (3) how effectively DUST-net transfers from simulation to a real-world setting. We evaluated DUST-net's performance on two simulated benchmarking datasets: the objects dataset provided by Abbatematteo et al. [1], and the PartNet-Mobility dataset [20, 93, 149], as well as a set of three real-world objects. From the simulated articulated object dataset [1], we considered the cabinet, microwave, and toaster oven for revolute articulations and the drawer object class for prismatic articulations. From the PartNet-Mobility dataset [20, 93, 149], we considered five object classes: the dishwasher, oven, and microwave object classes for the revolute articulation model category, and the storage furniture object class consisting of either a single column of drawers or multiple columns of drawers, for the prismatic articulation model category. Among the three sets of experiments, we conducted the first two sets of experiments on the simulated datasets, while the last set of experiments were conducted on the real-world object dataset. In all the experiments, we assumed that the input depth images are semantically segmented and contain non-zero pixels corresponding only to the two objects between which we wish to estimate the articulation model.

We compared DUST-net's performance in estimating point estimates for articulation model parameters with two state-of-the-art methods, ScrewNet [58] and an MDN-based approach proposed by Abbatematteo et al. [1]. ScrewNet estimates the object's articulation model parameters in a local frame located at the center of the object, whereas DUST-net does so directly in the camera frame. We compare our method with ScrewNet predicting parameters both in the object local frame and the camera frame. Additionally, we propose two baseline methods that estimate distributions over articulation model parameters and compare to them DUST-net. The first baseline method (vm-SoftOrtho) can be viewed as an extension of ScrewNet to a distributional setting. It represents the uncertainty over the screw axis orientation vector $\mathbf{l}$ and the direction of moment vector $\hat{\mathbf{m}}$ using two independent

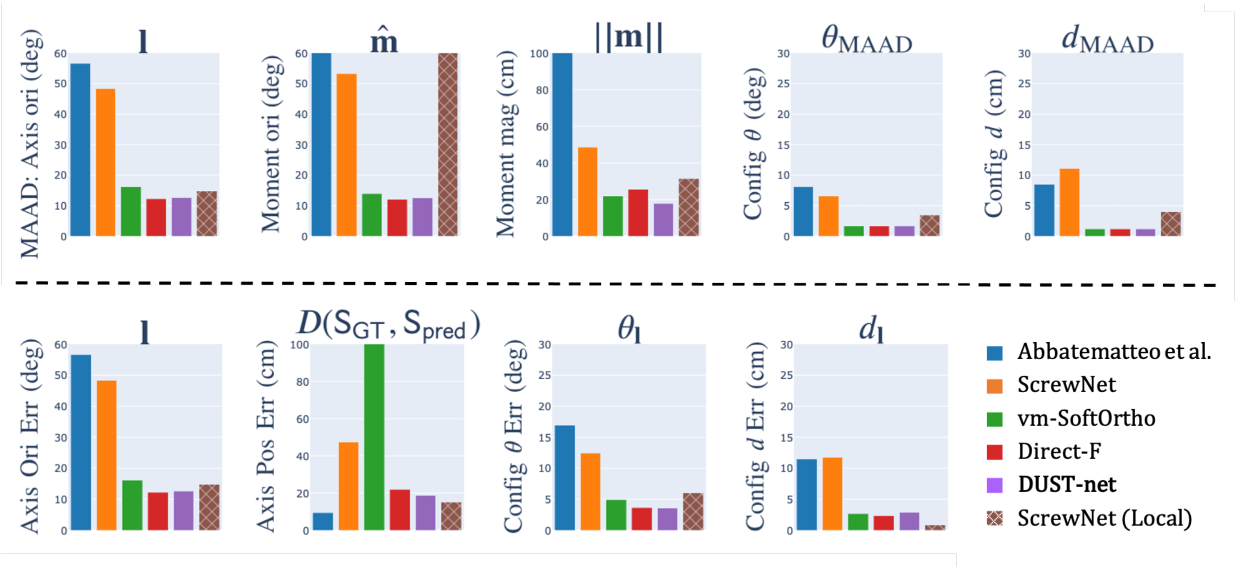


Figure 7.3: Mean error values on the MAAD (top) and Screw Loss (bottom) metrics for the PartNet-Mobility dataset [20, 93, 149] (lower values are better). Point estimates for DUST-net (violet) correspond to the modes of the distributions predicted by DUST-net.

von Mises-Fisher distributions and imposes a soft orthogonality constraint over the modes of the two distributions. The distributions over the moment vector magnitude $\|\mathbf{m}\|$ and configurations $q_{1:n-1}$ are considered to be normal distributions. This method suffers from the same drawback as ScrewNet, i.e., the use of a soft orthogonality constraint during training, and therefore cannot predict a valid set of screw axis parameters directly, unlike DUST-net. The second baseline method (Direct F) uses the same probability distribution as DUST-net to represent the uncertainty over the articulation model parameters, but estimates the individual elements of the $\mathbf{F}$ matrix directly. As a result, it fails to capture the uncertainty over model parameters accurately.

7.3.1 Accuracy of Point Estimates

The first set of experiments evaluated DUST-net's accuracy in predicting point estimates for articulation model parameters. We use the mode of the estimated distribution as the point estimate for model parameters. We used two metrics to evaluate accuracy: Mean Absolute (Angular) Deviation (MAAD) and Screw Loss (Metric proposed in ScrewNet [58]). MAAD metric indicates how close the individual screw parameters are to targets, whereas the Screw Loss indicates how close the complete predicted screw transforms is to the target transforms. The MAAD metric calculates the angular distance between the orientation of the predicted and ground-truth axis orientation vectors $\mathbf{l}$ and the orientation vectors of the screw axis moment vectors $\hat{\mathbf{m}}$. For the remaining parameters ($\|\mathbf{m}\|, \theta_{1:n-1}, d_{1:n-1}$), it calculates the mean absolute deviation between the predicted and ground-truth values. The screw loss reports the angular distance between the predicted and ground-truth screw axis orientation vectors $\mathbf{l}$ as orientation error and the length of the shortest perpendicular between the predicted and ground-truth screw axes as the distance between them. Configuration errors $\theta_{1:n-1}$ are reported as the difference between the predicted rotation about the predicted screw axis and the true rotation, whereas errors over $d_{1:n-1}$ are calculated as the Euclidean distance between the points displaced by the predicted and true displacements along respective axes.

Results for the synthetic articulated objects dataset and the PartNet-Mobility dataset are shown in Figures 7.2 and 7.3, respectively. Results demonstrate that under both metrics, the estimates obtained from DUST-net are typically more accurate than those obtained from the state-of-the-art methods. The first baseline, vm-SoftOrtho, performs comparably with DUST-net on both datasets when only MAAD estimates are considered. However, Figures 7.2 and 7.3 show that it produces a very high distance ($\approx$ 1m) between the predicted

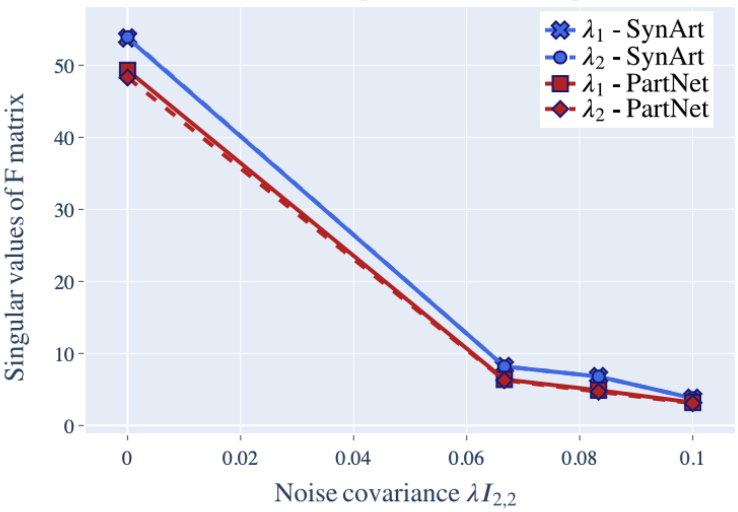


Figure 7.4: Variation of the mean of the singular values of predicted distribution concentration matrices over screw axes by DUST-net with artificially injected noise. Predicted singular values decrease monotonically with input noise, showing that the network's confidence over the predicted parameters decreases with input noise.

and ground-truth screw axes. This error arises due to the soft-orthogonality constraint used by vm-SoftOrtho, as DUST-net and the second baseline method, both of which handle the constraint implicitly, do not report high errors on that metric. Meanwhile, the second baseline, Direct F, performs comparably with DUST-net on both metrics for both datasets, but fails to capture the uncertainty over parameters with the required accuracy.

7.3.2 Uncertainty Estimation

The second set of experiments evaluated how effectively DUST-net's predicted distribution captures the epistemic uncertainty over the predicted articulation parameters. We evaluate this by adding artificial noise to the training labels form the two simulated datasets while training DUST-net. As more noise is added, we expect the confidence estimates pro-

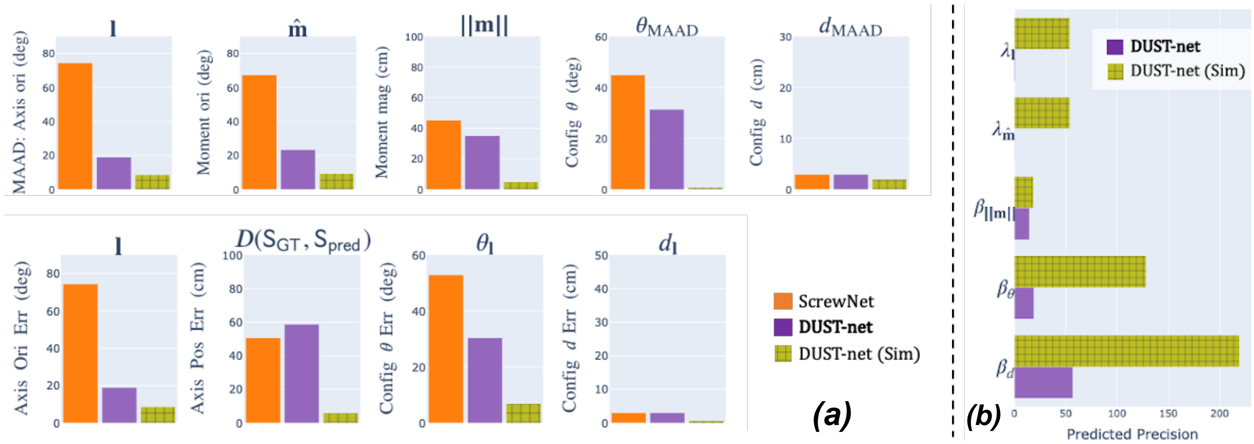


Figure 7.5: **(a)** Mean error values on MAAD (top) and Screw Loss (Bottom) metrics for real-world objects when the network was trained solely using simulated data [1] **(b)** Predicted concentrations over articulation model parameters. DUST-net estimation performance on simulated data [1] (hatched green) included for comparison. DUST-net reported lower confidence in its predictions for real-world objects than simulated data (b), analogous to its degraded estimation accuracy(a).

duces by DUST-net to decrease as well. We add noise to the labels by sampling perturbations from a matrix von Mises-Fisher distribution with varying singular values λ_1 and λ_2 of the distribution parameter matrix $\mathbf{F}$ and the truncated normal distributions with varying precision parameters $\beta_j, j \in \{\|\mathbf{m}\|, \theta, d\}$. Figure 7.4 show the variation of the mean of the singular values of the predicted distribution concentration matrices over screw axes by DUST-net with injected noise. In the noiseless case, the singular values of the matrix von Mises-Fisher distribution increases until they reach their maximum allowed value at $\lambda_{max} = 50$. When label noise is added, our results show that DUST-net's confidence over its predicted parameters decreases monotonically as more noise is added to the labels, supporting our hypothesis.

7.3.3 Sim to Real Transfer

Lastly, we evaluated how effectively DUST-net transfers from simulation to a real-world setting. DUST-net was trained solely on the simulated articulated object dataset [1]. Afterward, we used it to infer the articulation model parameters for three real-world objects. Results (Fig. 7.5(a)) report that DUST-net outperforms the current state-of-the-art method, ScrewNet, in estimating the model parameters for real-world objects. However, the estimated parameters using DUST-net are not yet accurate enough to be used directly for manipulating these objects. This sub-par performance stems from the significant differences between the training (clean and information-rich simulation data) and test datasets, which consists of noisy depth images acquired with a Kinect sensor and contain high salt-and-pepper noise, spurious features, and incomplete objects. A noteworthy insight from the results is that DUST-net also reports very low confidence over the predicted parameters for real-world objects, compared to when it is tested on the simulated data (Fig. 7.5(b)). This clearly delineates why it is beneficial to estimate a distribution over the articulation model parameters instead of point estimates. Given only point estimates of articulation model parameters, a robot has no way to determine if the estimates are reliable for manipulating the object safely or not. In contrast, DUST-net's reported confidence over the predictions could allow the robot to develop safe motion policies for articulated objects [55, 137] or use active learning based methods [23] to reduce uncertainty over the articulation parameters.

7.4 Conclusion

We introduced DUST-net, which utilizes a novel distribution over screw transforms on a Stiefel manifold to perform category-independent articulation model estimation with uncertainty estimates. We evaluated our approach on two benchmarking datasets and three real-world objects and compared its performance with two current state-of-the-art methods [1, 58]. Results show that DUST-net can estimate articulation models, their parameters, and model uncertainty estimates for novel objects across articulation model categories successfully with better accuracy than the state-of-the-art methods.

At present, DUST-net can only predict parameters for 1-DOF articulation models directly. For multi-DoF objects, an additional image segmentation step is required to mask out all non-relevant object parts. This procedure can be repeated iteratively for all object part pairs to estimate relative models between object parts that can be combined later to construct a complete kinematic model for the object [56]. An interesting extension of DUST-net could estimate parameters for multi-DoF objects directly by learning a segmentation network along with it. Another exciting direction of future work is to use DUST-net in an active learning setting where, if the robot is not confident enough about the estimates of the articulation model parameters, it can actively take information-gathering actions to reduce uncertainty.

Chapter 8

Future Work

In this chapter we discuss open questions and areas of future work related to the work presented in this dissertation.

8.1 Robot Motion Planning Under Uncertainty and Hybrid Dynamics

In Chapter 4, we proposed to model robot manipulation tasks involving sudden transitions in dynamics using hybrid dynamics models. We then introduced the hierarchical POMDP-HD planner that leverages the hybrid structure of state transition dynamics models to develop motion plans for such tasks under uncertainty. A limitation of modeling robot manipulation tasks using hybrid models is that the number of local dynamics models in the model, i.e., the discrete states, can increase combinatorially with the number of objects involved in the task and possible contact configurations between them. POMDP-planning with a large number of states is a difficult task as the time complexity for finding a solution for the problem can increase exponentially with the number of states [62]. This problem may further exacerbate for a hierarchical planner such as the POMDP-HD planner as it generates a complete continuous state trajectory at its lower level for each high-level candidate plan. A potential solution for resolving this issue may be to take a greedy approach for generating the candidate high-level plans. Instead of considering candidate plans starting from all possible current discrete states, the high-level planner may develop plans only for those states, in which its belief over being in that state is more than an ϵ threshold. This greedy behavior can significantly reduce the number of states that the planner needs to consider while generating the high-level plans and help it to develop motion plans tractably even for complex tasks.

8.2 Learning Object Kinematics from Observations

In this dissertation, we introduced two methods, ScrewNet (Chapter 6) and DUST-net (Chapter 7), that can learn articulation properties for objects directly from raw observations. However, at present, they can only predict parameters for 1-DOF articulation models directly. For multi-DoF objects, an additional image segmentation step is required to mask out all non-relevant object parts. This procedure can be repeated iteratively for all object part pairs to estimate relative models between object parts that can be combined later to construct a complete kinematic model for the object using MICAH (Section 5.2.2). An interesting extension of these methods would be to estimate parameters for multi-DoF articulated objects directly by learning a segmentation network along with them.

Another limitation of DUST-net is that its predictions for the concentration parameters Λ indicating the singular-values of the parameter matrix $\mathbf{F}$ for the matrix von Mises-Fisher distribution over the Stiefel manifold $\mathcal{V}_{3,2}$ are upper-bounded. This limitation arises due to the numerical approximation for the normalization factor ${}_0F_1$ of the distribution. Pal et al. [102] have recently proposed an approximation for the hypergeometric function ${}_0F_1$ for a given value of the input argument matrix D and an error level ϵ. With this approximation, the upper bounds on the concentration parameters Λ may be removed.

DUST-net's ability to predict distributions over articulation model parameters for objects opens other exciting avenues for future research directions as well. One such direction is to use DUST-net in an active learning setting where, if the robot is not confident enough about the estimates of the articulation model parameters, it can actively take information-gathering actions to reduce uncertainty. Another interesting direction would be to combine DUST-net with approaches to learn behavior policies that provide safety assurances, such as the one suggested by Taylor et al. [137], to develop an end-to-end system for interacting with articulated objects safely.

8.3 Combined Framework

In this dissertation, we introduced methods for learning state transition functions for objects directly from observations. We also introduced the POMDP-HD planner that develops motion plans for long-horizon robot manipulation tasks under uncertainty. A fruitful future research direction would be to combine all these methods in a single framework. Given a sequence of depth images depicting the motion of different object parts, the combined framework will directly generate an uncertainty robust motion plan for novel long-horizon manipulation tasks involving the object. A schematic for the complete framework is shown in Figure 8.1. Given the depth images, the combined framework may first use DUST-net and the hybrid automaton construction algorithm (HAC) to learn a state transition function for the object and later use the learned model with the POMDP-HD planner to develop long-horizon motion plans under uncertainty for novel manipulation tasks involving the object. This framework will enable service robots to perform long-horizon manipulation tasks involving novel objects reliably and safely even while using noisy sensors.

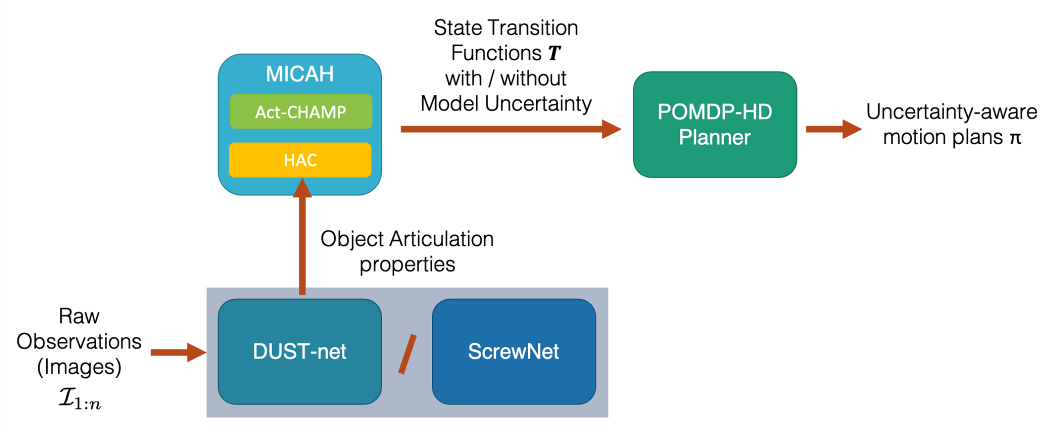


Figure 8.1: A schematic of the combined framework. Given a sequence of depth images depicting the motion of different parts of an articulated object, will directly generate an uncertainty robust motion plan for novel long-horizon manipulation tasks involving the object

8.4 POMDP-based Task and Motion Planning

Toussaint et al. [143] recently proposed a Logic-Geometric Programming (LGP) [141, 143] based framework to jointly perform task and motion planning for robot manipulation tasks involving multiple objects. The framework uses hybrid models to model the dynamics of the objects involved in a task and interactions between them. Ha et al. [42], Migimatsu and Bohg [91] have further extended the framework to robot manipulation tasks with stochastic dynamics. An interesting future research direction would be to extend the framework further to a POMDP task setting, i.e., for robot manipulation tasks with stochastic dynamics and observational uncertainty. As the POMDP-HD planner [55] directly operates over hybrid dynamics models, combining it with the LGP framework may lead to a promising solution for performing POMDP-based task and motion planning.

8.5 Learning Object Dynamics Models from Observations

A basic rigid-body model of an object has four components: link geometry parameters, link inertia parameters, joint models with their parameters, and a link connectivity graph [121]. While various methods have focused on learning these components from data individually (link geometry [19, 92], link inertia parameters [38, 78], joint models [56–58], and link connectivity graph [56, 156]), learning a complete rigid body dynamics model for objects has largely been unexplored. Learning such models for objects will help robots to reason directly about the applied external forces and torques on the objects, enabling them to perform dynamic manipulation tasks involving those objects dexterously. An exciting future research direction would be to explore methods that can enable robots to learn dynamics models for complex objects, such as high-DoF articulated objects and objects exhibiting configuration-dependent dynamics, directly from observations. A promising research direction lies in exploring the conjugation of these methods for learning physically-grounded object dynamics models [38, 78] with the the methods presented in this dissertation for learning kinematics models for objects from observations.

Chapter 9

Conclusion

Service robots that can assist humans in performing day-to-day tasks will need to be general-purpose robots that can perform a wide array of tasks without requiring much supervision. However, current robots are typically highly specialized, can only perform fixed, predefined tasks reliably, need to operate in controlled environments, and require complete and accurate knowledge about their surroundings to function effectively. Through this dissertation, we aim to bridge this gap by introducing methods that enable robots to learn about novel tasks directly through observations and use this knowledge for performing novel manipulation tasks with high reliability even under uncertainty.

Lacking complete information about its tasks and environments, a service robot cannot use pre-programmed motion policies or online motion generation methods that do not reason about the lack of information as they may be unreliable under partial information. Reasoning about the lack of complete information becomes critical for manipulation tasks that a service robot would have to perform. Their manipulation tasks will often require interacting with multiple objects that make or break contacts during the task. Contacts can cause sudden transitions in objects' dynamics, which upon compounding with state estimation errors, can cause the robot to fail in completing its tasks.

Addressing this, in this dissertation, we used the Partially Observable Markov Decision Processes (POMDPs) [62] framework to develop motion plans for robots that reason about the uncertainty due to partial information as well while generating motion plans. However, POMDPs incur exponentially increasing computational costs with planning time horizon [62, 80, 97, 103, 145], which restricts the current POMDP-based planning methods to problems having short time horizons. Another challenge that arises while using planning-based methods is that they require state transition dynamics functions for the world they are operating in to develop plans. As a service robot will need to perform novel tasks while assisting humans, it must be able to learn the state transition functions for its tasks in an online fashion using the information available to it.

Addressing these challenges, this dissertation advanced the state-of-the-art of robot motion planning under uncertainty and model learning from observations by answering the question:

How can a robot learn state transition functions for complex manipulation tasks directly from observations and use them to perform long-horizon planning under uncertainty?

In Chapter 4, we focused on developing motion plans for long-horizon robot manipulation tasks that reason about the uncertainty due to partial information as well. We first proposed to model the world state transition functions for robot manipulation tasks involving sudden transitions, such as those due to contacts, using hybrid models. We then introduced a hierarchical POMDP planner that leverages the representational power of hybrid models to develop reliable motion plans for long-horizon tasks under uncertainty.

Next, we addressed the challenge of learning state transition functions for tasks involving manipulating household objects directly from observations. One of the largest sets of household objects that a service robot would need to interact with frequently is household appliances. Household appliances, such as refrigerators, microwaves, and drawers, are predominantly articulated objects and consist of multiple functional parts connected by mechanical joints such as hinges and sliding joints. Motivated by this, in this dissertation, we introduced methods for learning kinematics models for such objects directly from observations and presented an algorithm to construct the state transition functions from the learned kinematics models for manipulating these objects.

In Chapter 5, we presented the first method, MICAH, which consists of two algorithms. First, ACT-CHAMP, an action-conditional model inference algorithm that estimates kinematic models and possible changepoints between them for articulated objects given unsegmented data. It is well-suited for articulated objects that exhibit configuration-dependent articulation properties, such as a stapler that intrinsically changes its articulation state (e.g., rigid vs. rotational) based on the relative angle between its arms. Second, an algorithm for constructing hybrid automata representing state transition functions for articulated objects using the detected changepoints and estimated local models from Act-CHAMP. We demonstrated the accuracy of MICAH in learning models by using it to learn such models for three household objects and using the learned models to perform novel manipulation tasks involving those objects.

We introduced the second method, ScrewNet, in Chapter 6. ScrewNet removed MICAH's requirement of needing series of observations of relative poses between the object parts as input. It presented a way to learn articulation properties of objects directly from raw depth images without requiring to know their articulation model category a priori. We demonstrated ScrewNet's effectiveness by evaluating its performance on two benchmarking simulation datasets and a real-world dataset and showing that it outperforms a state-of-the-art method while requiring only half the training data.

Lastly, in Chapter 7 we presented the third method, DUST-net, that learns distributions over articulation model parameters directly from raw depth images. Results demonstrated that the distributional representation helps DUST-net to achieve better accuracy than other state-of-the-art methods while making it more computationally and data-efficient than other methods. By reasoning about the uncertainty in the estimated parameters, DUST-net opens several new avenues for ensuring success in robot manipulation tasks, such as using uncertainty-robust planning methods and using active learning approaches to reduce uncertainty over the articulation parameters. The learned distributions over the articulation model parameters for an object using DUST-net can later be combined with the hybrid automata construction algorithm of MICAH to construct a state transition function with model uncertainty estimates for manipulating the object.

This dissertation introduced methods for learning state transition functions for robot manipulation tasks directly from observations and using them to perform long-horizon motion planning to complete the tasks. Jointly these methods present a unified framework that can enable robots to adapt to their work environments quickly and learn to perform novel manipulation tasks reliably without much supervision. We believe that this dissertation provides foundational results on robot motion planning uncertainty and online model learning that can bridge the gap between the current robots and the dexterous general-purpose service robots.

Appendices

Appendix A

POMDP-HD Planer

A.1 Preliminaries

A.1.1 Trajectory Optimization using Direct Transcription

Direct Transcription is a trajectory optimization method in which a constrained non-linear optimization problem is set up with the user-defined objective function over a set of *knot-points* $\{\mathbf{x}_i,\ \mathbf{u}_i\}$) chosen to discretize the continuous space trajectory into a set of decision variables. The system dynamics are imposed as the constraints on the optimization problem. For discrete-time systems, these knot-points can be taken as the system state $\mathbf{x}_t$ and the control input $\mathbf{u}_t$ at each time step t. However, planning for longer horizons will then require specifying a high number of *knot-points* $(\mathbf{x}_i,\ \mathbf{u}_i)$ which can result in very high computational costs. This can be resolved by approximately parameterizing the space of possible trajectories by a series of M segments and solving the optimization problem for a knot points only at the start and end points of segments. The intermediate points on the trajectory can be obtained via numerical integration. Let $x'_{1:M}$ and $u'_{1:M-1}$ be sets of state and action variables that parameterize the trajectory in terms of segments. The i^{th} segment can be assumed to start at time $i\delta$ and ends at time $i\delta + \delta - 1$, where $\delta = \dfrac{T}{M}$ for a time horizon T.

A general objective function for trajectory optimization can be given as

$$J(x_{1:T}, u_{1:T}) \approx \hat{J}(x'_{1:M}, u'_{1:M}) = \sum_{j=1}^{M} x_i'^{T} Q x'_i + u_i'^{T} R u'_i \tag{A.1}$$

where Q and R represent the cost matrices associated with the state and the input respectively. The system dynamics incorporated as constraints can be defined as:

$$x'_2 = \phi(x'_1, u'_1), \quad \dots \quad x'_k = \phi(x'_{k-1}, u'_{k-1}) \tag{A.2}$$

where the function $\phi(x'_i, u'_i)$ can be seen as performing numerical integration of the current state variable x'_i till the next state variable x'_{i+1}. The function ϕ is given as

$$x'_{i+1} = \phi(x'_i, u_i) = F(x'_i, u_i) + \sum_{t=i\delta}^{i\delta+\delta-1} [F(x_{t+1}, u_i) - F(x_t, u_i)] \tag{A.3}$$

where $F(x_t, u_t)$ represents the system dynamics.

Trajectory optimization using direct transcription can be extended for belief space planning by assuming Gaussian noise over continuous states [61]. If the belief over continuous states is defined as $b_t = \mathcal{N}(\mu_t, \Sigma_t)$, trajectory optimization can be formulated as an optimization problem over variables μ_t and s_t, where μ_t represents the mean of the belief state and $s_t = \{s_1^T, ..., s_d^T\}^T$ is a vector composed of d columns of $\Sigma_t = [s_1, ..., s_d]$. Analogous to the deterministic case, problem is constrained to follow belief space dynamics. The corresponding objective function can be given as

$$\begin{aligned} J(b_{1:T}, u_{1:T}) &\approx \hat{J}(b'_{1:M}, u'_{1:M}) \\ &= s_M^T \Lambda s_M + \sum_{j=1}^{M} \mu_i'^T Q \mu'_i + u_i'^T R u'_i \end{aligned} \tag{A.4}$$

where Q, R and Λ represent the cost matrices associated with belief mean, control input and the belief covariance at final discrete time step respectively. Belief dynamics can be incorporated in the formulation as the constraints:

$$b'_2 = \Phi(b'_1, u'_1), \quad ... \quad b'_k = \Phi(b'_{k-1}, u'_{k-1}) \tag{A.5}$$

where the function $\Phi(b'_i, u_i)$ is given as

$$b'_{i+1} = \Phi(b'_i, u_i) = \mathcal{F}(b'_i, u_i) + \sum_{t=i\delta}^{i\delta+\delta-1} [\mathcal{F}(b_{t+1}, u_i) - \mathcal{F}(b_t, u_i)] \tag{A.6}$$

where $\mathcal{F}(b'_i, u_i)$ represents extended system dynamics. Propagation of belief b_t through system dynamics $\mathcal{F}(b'_i, u_i)$ has been previously discussed by Platt et al. [61] in further details.

A.2 Further Experimental Details

A.2.1 Domain-I

Matrices defining the cost function over error in states, control input, additional cost for final state error and covariance were taken as $Q = diag(0.5, 0.5)$, $R = diag(10.0, 10.0)$, $Q_T = 1e4$ and $\Lambda = 1e7$ respectively. Number of Gaussians used to model continuous belief $L = 1$.

A.2.2 Domain-II

Feedback was obtained on the location of the airplane in the world frame by doing an online color-based object cluster extraction, using multi-plane segmentation from the Point Cloud Library (PCL) on the point cloud data of a Microsoft Kinect v2 sensor. Matrices defining the cost function over error in states, control input, additional cost for final state error and covariance were taken as $Q = diag(0.5, 0.5)$, $R = diag(0.1, 0.1)$, $Q_T = 5000$ and $\Lambda = 10^7$ respectively. Number of Gaussians used to model continuous belief $L = 1$.

Appendix B

ScrewNet

B.1 Experimental details

B.1.1 Dataset

Objects used in the experiments from each of the dataset are shown in the Figures B.1 and B.2. We sampled a new object geometry and a joint location for each training example in the simulated articulated object dataset, as proposed by [1]. For the PartNet-Mobility dataset, we considered 11 microwave (8 train, 3 test), 36 dishwasher (27 train, 9 test), 9 oven (6 train, 3 test), 26 single column drawer (20 train, 6 test), and 14 multi-column drawer (10 train, 4 test) object models. For both datasets, we sampled object positions and orientations uniformly in the view frustum of the camera up to a maximum depth dependent upon the object size.

B.1.2 Experiment 1: Same object class

Numerical error values for the first set of experiments for the simulated articulated objects dataset are presented in the Table B.1. It is evident from the Table B.1 that the baseline succeeded in achieving nearly zero prediction error (0.08°) in joint axis orientation estimation for all object classes. ScrewNet also performed well and reported low prediction

Figure B.1: Object classes used from the simulated articulated object dataset [1]. Object classes: cabinet, drawer, microwave, and toaster (left to right)

Figure B.2: Object classes used from the PartNet-Mobility dataset [20, 93, 149]. Object classes: dishwasher, oven, microwave, drawer- 1 column, and drawer- multiple columns (left to right)

errors ($< 0.5°$) for the drawer, microwave, and toaster object classes. For the cabinet object class, while ScrewNet reported a higher mean error ($\sim 5°$), it is relatively small compared to the difference in axis orientations, 180°, between the two possible configurations of the cabinet (left-opening or right-opening). For the other two model parameters, namely the joint axis position and the observed configurations, ScrewNet significantly outperformed the baseline method.

	Axis Orientation (deg)	**Axis displacement (cm)**	**Configuration**
Cabinet - Baseline	$\mathbf{0.082 \pm 0.000}$	8.472 ± 6.277	16.921 ± 8.212 deg
Cabinet - Ours	5.667 ± 13.888	$\mathbf{1.236 \pm 0.66}$	$\mathbf{1.083 \pm 0.707}$ deg
Drawer - Baseline	$\mathbf{0.082 \pm 0.000}$	4.067 ± 1.483	7.389 ± 2.439 cm
Drawer - Ours	0.252 ± 0.000	$\mathbf{1.114 \pm 0.035}$	$\mathbf{1.517 \pm 0.016}$ cm
Microwave - Baseline	$\mathbf{0.083 \pm 0.000}$	3.441 ± 1.218	21.670 ± 7.097 deg
Microwave - Ours	0.304 ± 0.000	$\mathbf{2.322 \pm 1.323}$	$\mathbf{0.329 \pm 0.005}$ deg
Toaster - Baseline	$\mathbf{0.082 \pm 0.000}$	2.669 ± 1.338	11.902 ± 4.242 deg
Toaster - Ours	0.114 ± 0.000	$\mathbf{1.566 \pm 0.018}$	$\mathbf{0.314 \pm 0.018}$ deg

Table B.1: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test object instances for each object class from the simulated articulated objects dataset [1]. Lowest error values for a particular test object set are reported in bold.

Numerical error values for the first set of experiments for the PartNet-Mobility dataset are reported in the Table B.2. Similar trends followed in the performance of the two approaches. The baseline achieved very high accuracy in predicting the joint axis orientation, whereas ScrewNet reported reasonably low but slightly higher errors ($< 2.5°$). For the joint

	Axis Orientation (deg)	Axis displacement (cm)	Configuration
Dishwasher - Baseline	**0.082 ± 0.000**	46.267 ± 20.247	9.735 ± 4.6 deg
Dishwasher - Ours	0.918 ± 0.000	**6.136 ± 5.455**	**4.037 ± 1.613** deg
Oven - Baseline	**0.082 ± 0.000**	81.444 ± 27.083	13.087 ± 4.649 deg
Oven - Ours	0.583 ± 0.223	**2.111 ± 1.910**	**0.720 ± 0.140** deg
Microwave - Baseline	**0.082 ± 0.000**	26.781 ± 10.273	11.856 ± 2.456
Microwave - Ours	0.879 ± 0.063	**4.893 ± 4.252**	**2.549 ± 0.939** deg
Drawer- 1 column - Baseline	**0.082 ± 0.000**	79.228 ± 13.944	17.524 ± 3.700 cm
Drawer- 1 column - Ours	2.140 ± 0.000	**11.567 ± 9.748**	**3.181 ± 0.793** cm
Drawer- Multi. cols. - Baseline	**0.082 ± 0.000**	63.064 ± 18.913	4.483 ± 6.403 cm
Drawer- Multi. cols. - Ours	1.287 ± 0.000	**12.557 ± 8.317**	**4.419 ± 2.891** cm

Table B.2: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test cases for each object class from the PartNet-Mobility Dataset

axis position and the observed configurations, ScrewNet outperformed the baseline method on this dataset as well.

B.1.3 Experiment 2: Same articulation model category

Numerical results for the second set of experiments are reported in the Table B.3. It is evident from the Table B.3 that the ScrewNet was able to generalize to novel object classes belonging to the same articulation model category, while the baseline method failed to do so. While both approaches reported comparable errors in estimating the joint axis orientations and the observed configurations, the baseline reported errors of an order of magnitude higher than ScrewNet in the joint axis position estimation.

	Axis Orientation (deg)	Axis displacement (cm)	Configuration
Oven - Baseline	**0.082 ± 0.000**	44.699 ± 12.259	9.915 ± 3.934 deg
Oven - Ours	0.918 ± 0.000	**7.486 ± 1.273**	**8.650 ± 0.207** deg
Drawer- 1 column - Baseline	**0.082 ± 0.000**	50.990 ± 25.984	5.283 ± 8.862 cm
Drawer- 1 column - Ours	1.287 ± 0.000	**14.548 ± 5.823**	**4.399 ± 0.654** cm

Table B.3: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test objects belonging to each object classes from the PartNet-Mobility Dataset

B.1.4 Ablation studies

We consider three ablated versions of ScrewNet. First, to test the effectiveness of the proposed loss function, we consider an ablated version of ScrewNet which is trained using a raw L2-loss between the labels and the network predictions (named as L2-Error version while reporting results). As the second ablation study, we test whether using an LSTM layer in the network helps with the performance or not (named as NoLSTM version while reporting results). We replace the LSTM layer of the ScrewNet with a fully connected layer such that the two networks, ScrewNet and its ablated version, have a comparable number of parameters. Lastly, to check if a sequence of images is helpful in the model estimation or not, we consider an ablated version of ScrewNet that estimates the articulation model using just a pair of images (named as 2_imgs version while reporting results). Note that ScrewNet and all its ablated versions use a single network each. Numerical results for the simulated articulated objects dataset are presented in the Table B.4, and for the PartNet-Mobility dataset are shown in the Table B.5.

	Axis Orientation (deg)	**Axis displacement (cm)**	**Configuration**
Cabinet - Baseline*	$\mathbf{0.082 \pm 0.000}$	8.472 ± 6.277	16.921 ± 8.212 deg
Cabinet - NoLSTM	8.688 ± 20.504	$\mathbf{2.521 \pm 4.341}$	$\mathbf{1.984 \pm 5.172}$ deg
Cabinet - L2-Error	90.186 ± 12.244	5.580 ± 5.138	3.847 ± 5.377 deg
Cabinet - 2_imgs	18.716 ± 40.197	3.188 ± 5.795	12.898 ± 8.846 deg
Cabinet - Ours	16.988 ± 14.971	5.479 ± 4.363	4.65 ± 5.904 deg
Drawer - Baseline*	$\mathbf{0.082 \pm 0.000}$	4.067 ± 1.483	7.389 ± 2.439 cm
Drawer - NoLSTM	14.957 ± 25.526	2.116 ± 2.287	3.878 ± 2.883 cm
Drawer - L2-Error	7.931 ± 13.745	11.141 ± 3.159	5.847 ± 1.468 cm
Drawer - 2_imgs	23.310 ± 27.888	5.118 ± 2.829	7.664 ± 4.883 cm
Drawer - Ours	3.473 ± 8.839	$\mathbf{1.302 \pm 0.999}$	$\mathbf{2.448 \pm 1.092}$ cm
Microwave - Baseline*	$\mathbf{0.082 \pm 0.000}$	3.441 ± 1.218	21.67 ± 7.097 deg
Microwave - NoLSTM	2.725 ± 8.813	$\mathbf{2.439 \pm 1.708}$	$\mathbf{0.803 \pm 2.519}$ deg
Microwave - L2-Error	10.125 ± 10.953	3.76 ± 3.021	4.957 ± 4.489 deg
Microwave - 2_imgs	2.547 ± 3.480	5.115 ± 5.076	18.269 ± 12.658 deg
Microwave - Ours	8.770 ± 13.363	3.398 ± 2.675	4.033 ± 5.998 deg
Toaster - Baseline*	$\mathbf{0.082 \pm 0.000}$	2.669 ± 1.338	11.902 ± 4.242 deg
Toaster - NoLSTM	7.410 ± 17.645	$\mathbf{2.597 \pm 1.86}$	$\mathbf{1.030 \pm 2.230}$ deg
Toaster - L2-Error	18.750 ± 17.243	9.173 ± 4.229	2.661 ± 2.823 deg
Toaster - 2_imgs	12.833 ± 22.596	4.123 ± 3.196	16.016 ± 10.703 deg
Toaster - Ours	11.583 ± 14.798	3.003 ± 1.75	3.471 ± 2.876 deg

Table B.4: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test objects belonging to each object classes from the simulated articulated objects dataset. Symbol * denote that the baseline has a significant advantage over other methods as it uses a separate network for each object class, while all ScrewNet and its ablations use a single network

	Axis Orientation (deg)	**Axis displacement (cm)**	**Configuration**
Dishwasher - Baseline*	**0.082 ± 0.000**	46.267 ± 20.247	9.735 ± 4.600 deg
Dishwasher - NoLSTM	41.485 ± 41.184	9.815 ± 6.782	**5.415 ± 4.097** deg
Dishwasher - L2 Error	25.405 ± 15.119	12.653 ± 8.119	7.828 ± 1.913 deg
Dishwasher - 2_imgs	1.935 ± 0.021	11.544 ± 4.729	5.706 ± 4.152 deg
Dishwasher - Ours	11.850 ± 15.267	**6.789 ± 5.630**	6.081 ± 3.043 deg
Oven - Baseline*	**0.082 ± 0.000**	81.429 ± 27.244	13.026 ± 4.670 deg
Oven - NoLSTM	29.968 ± 39.034	11.014 ± 13.235	10.574 ± 6.332 deg
Oven - L2 Error	27.197 ± 13.103	26.452 ± 14.704	11.823 ± 1.067 deg
Oven - 2_imgs	1.939 ± 0.018	**4.791 ± 1.370**	10.498 ± 7.481 deg
Oven - Ours	7.881 ± 7.763	6.786 ± 2.443	**5.010 ± 1.233** deg
Microwave - Baseline*	**0.082 ± 0.000**	26.781 ± 10.273	11.856 ± 2.456 deg
Microwave - NoLSTM	40.911 ± 32.830	15.993 ± 14.080	3.865 ± 2.350 deg
Microwave - L2 Error	60.566 ± 7.705	59.286 ± 6.485	7.463 ± 1.612 deg
Microwave - 2_imgs	91.826 ± 0.012	**11.994 ± 2.549**	5.212 ± 3.606 deg
Microwave - Ours	24.959 ± 24.847	15.271 ± 13.561	**3.507 ± 1.987** deg
Drawer- 1 col. - Baseline*	**0.082 ± 0.000**	79.228 ± 13.944	17.524 ± 3.700 cm
Drawer- 1 col. - NoLSTM	42.318 ± 35.604	47.991 ± 29.586	10.923 ± 6.449 cm
Drawer- 1 col. - L2 Error	48.136 ± 9.533	60.046 ± 19.375	14.202 ± 2.153 cm
Drawer - 1 col. - 2_imgs	89.372 ± 0.047	80.356 ± 8.087	25.753 ± 18.374 cm
Drawer- 1 col. - Ours	19.876 ± 21.684	**28.329 ± 15.005**	**5.729 ± 4.259** cm
Drawer- Multi. cols. - Baseline*	**0.082 ± 0.000**	63.064 ± 18.913	4.483 ± 6.403 cm
Drawer- Multi. cols.- NoLSTM	38.393 ± 33.113	49.419 ± 23.998	6.181 ± 5.228 cm
Drawer- Multi. cols.- L2 Error	38.866 ± 5.243	44.422 ± 26.927	6.422 ± 0.766 cm
Drawer- Multi. cols. - 2_imgs	89.361 ± 0.053	78.131 ± 4.888	12.229 ± 3.961 cm
Drawer- Multi. cols. - Ours	9.292 ± 15.295	**17.813 ± 14.719**	**0.915 ± 1.772** cm

Table B.5: [Experiment: Across articulation model category] Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test objects belonging to each object classes from the PartNet-Mobility Dataset. Symbol $\star$ denote that the baseline has a significant advantage over other methods as it uses a separate network for each object class, while all ScrewNet and its ablations use a single network

Appendix C

DUST-net

C.1 Hypergeometric function ${}_pF_q$

A general hypergeometric function ${}_pF_q$ in the matrix argument can be written as an infinite series in terms of zonal polynomials, which are multivariate symmetric homogeneous polynomials and form a basis of the space of symmetric polynomials [22]. Given an $m \times m$ symmetric, positive-definite matrix Y, the hypergeometric function ${}_pF_q$ of matrix argument Y is defined as

$${}_pF_q\left(\begin{matrix} a_1, \ldots, a_p \\ b_1, \ldots, b_q \end{matrix} \middle| Y\right) := \sum_{n=0}^{\infty} \sum_{\nu \in \mathcal{P}_n} \frac{(a_1)_\nu \cdots (a_p)_\nu}{(b_1)_\nu \cdots (b_q)_\nu} \cdot \frac{\mathcal{C}_\nu(Y)}{n!}, \tag{C.1}$$

where

- $\mathcal{P}_n$ is the set of all ordered integer partitions of n
- $(a)_\nu$ is the generalized Pochhammer symbol, defined as

  $$(a)_\nu = (a)_{(\nu_1,\ldots,\nu_k)} := \prod_{i=1}^{k} \left(a - \frac{i-1}{2}\right)_{\nu_i};$$

  , where, $(a)_{\nu_i} = a(a+1)...(a+\nu_i-1), (a)_0 = 1$,
- and $\mathcal{C}_\nu(Y)$ denotes the zonal polynomial of Y, indexed by a partition ν, which is a symmetric homogeneous polynomial of degree n in the eigenvalues $y_1, \ldots, y_m$ of Y, satisfying

  $$\sum_{\nu \in \mathcal{P}_n} \mathcal{C}_\nu(Y) = (\operatorname{tr} Y)^n = (y_1 + \cdots + y_m)^n. \tag{C.2}$$

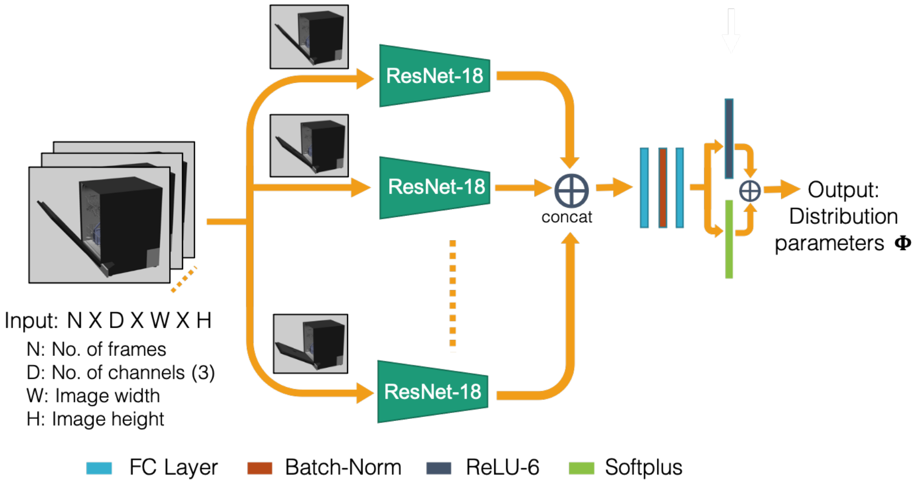


Figure C.1: DUST-net architecture

Using zonal polynomials, we can define the hypergeometric function ${}_0F_1(\frac{3}{2}, \frac{1}{4}\Lambda^2)$ defining the normalization factor of the matrix von Mises-Fisher distribution over Stiefel manifold $\mathcal{V}_{3,2}$ as

$$ {}_0F_1(\frac{3}{2}, \frac{1}{4}\Lambda^2) := \sum_{n=0}^{\infty} \sum_{\nu \in \mathcal{P}_n} \frac{1}{(\frac{3}{2})_\nu} \frac{C_\nu(\Lambda)}{n!}, \tag{C.3} $$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2)$, $\mathcal{P}_n$ is the set of all ordered integer partitions of n, $(a)_\nu$ is the generalized Pochhammer symbol, and $C_\nu(\Lambda)$ denotes the zonal polynomial of Λ indexed by a partition ν. This series converges for all input matrices for a general hypergeometric function ${}_pF_q$ if $p \leq q$, which holds in our case [22]. Recently, Jiu and Koutschan [60] investigated the zonal polynomials in detail and developed a computer algebra package to calculate these polynomials in SageMath. We use this package to calculate the the hypergeometric function ${}_0F_1(\frac{3}{2}, \frac{1}{4}\Lambda^2)$. However, as the number of terms in the series grows combinatorially with n, we truncate the series at $n = 25$ for computational reasons. Through our experimental analysis, we found that this truncated series is a good approximation of ${}_0F_1$ as the series converges to a finite value, if the singular values of the F, i.e. λ_1 and λ_2 remain below a maximum value $\lambda_{max} = 50$.

C.2 Network Architecture

Figure C.1 shows the detailed network architecture for DUST-net. DUST-net uses an off-the-shelf convolutional network, ResNet-18, to extract task-relevant visual features from the input images, which are later passed through a two-layer MLP to predict a set of parameters Φ for the distribution $p(\mathsf{S}, \theta_{1:n-1}, d_{1:n-1} \mid \mathcal{I}_{1:n}, \Phi)$. We use ReLU activations for the hidden fully-connected layers. The first four output parameters (out of 40) of the last linear layer of MLP correspond to the parameters (α, β, γ) and ω, representing the matrices Γ and Ω respectively, which lie in ranges $[0, 2\pi), [0, \pi), [0, 2\pi)$, and $[0, 2\pi)$ respectively. We pass the first four values of the output of the last linear layer through a ReLU-6 layer [49] to correctly map the predicted values with their respective ranges. The rest of the parameters are required to be non-negative. We pass the remaining output values of the last linear layer through a Softplus layer for non-negative output.

C.3 Experimental details

C.3.1 Datasets

Objects used in the experiments from each of the dataset are shown in the Figures C.2 and C.3. We sampled a new object geometry and a joint location for each training example in the simulated articulated object dataset, as proposed by [1]. For the PartNet-Mobility dataset, we considered 11 microwave (8 train, 3 test), 36 dishwasher (27 train, 9 test), 9 oven (6 train, 3 test), 26 single column drawer (20 train, 6 test), and 14 multi-column drawer (10 train, 4 test) object models. For both datasets, we sampled object positions and orientations uniformly in the view frustum of the camera up to a maximum depth dependent upon the object size. The objects and depth images are rendered in Mujoco [139]. We apply random frame skipping and pixel dropping to simulate noise encountered in real world sensor data.

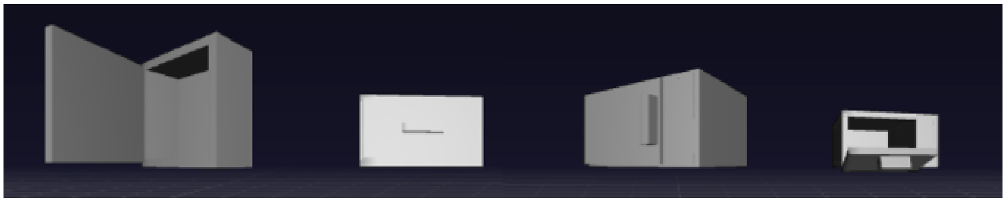


Figure C.2: Object classes used from the simulated articulated object dataset [1]. Object classes: cabinet, drawer, microwave, and toaster (left to right)

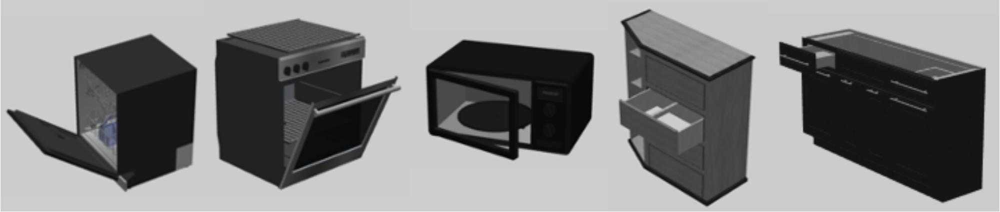


Figure C.3: Object classes used from the PartNet-Mobility dataset [20, 93, 149]. Object classes: dishwasher, oven, microwave, drawer- 1 column, and drawer- multiple columns (left to right)

We consider three household objects — a microwave, a drawer, and a toaster oven, in the real world objects dataset for evaluating DUST-net's performance. The objects are shown in Figure C.4.

To generate the labels for screw displacements, we follow the same procedure as used by Jain et al. [58]. Considering one of the objects, o_i, as the base object, we calculate the screw displacements between temporally displaced poses of the second object o_j with respect to it. Given a sequence of n images $\mathcal{I}_{1:n}$, we calculate a sequence of $n-1$ screw displacements ${}^1\boldsymbol{\sigma}_{o_j} = \{{}^1\sigma_2, ...{}^1\sigma_n\}$, where each ${}^1\sigma_k$ corresponds to the relative spatial displacement between the pose of the object o_j in the first image $\mathcal{I}_1$ and the images $\mathcal{I}_{k,\ k\in\{2...n\}}$. Note ${}^1\boldsymbol{\sigma}_{o_j}$ is defined in the frame $\mathcal{F}_{o_j^1}$ attached to the pose of the object o_j in the first image $\mathcal{I}_1$. We then transform ${}^1\boldsymbol{\sigma}_{o_j}$ to the camera frame by defining the 3D line motion matrix $\tilde{D}$ between the frames $\mathcal{F}_{o_j^1}$ and $\mathcal{F}_{o_i}$ [8], and transforming the common screw axis ${}^1\mathsf{S}$ to the target frame $\mathcal{F}_{o_i}$. The configurations ${}^1q_k$ remain the same during frame transformations. The 3D line motion matrix $\tilde{D}$ between two frames can be constructed using the rotation matrix R and a translation vector $\mathbf{t}$ between two frames $\mathcal{F}_A$ and $\mathcal{F}_B$, as:

$$\begin{bmatrix} {}^B\mathbf{l} \\ {}^B\mathbf{m} \end{bmatrix} = {}^B\tilde{D}_A \begin{bmatrix} {}^A\mathbf{l} \\ {}^A\mathbf{m} \end{bmatrix}, \qquad \text{where,} {}^B\tilde{D}_A = \begin{bmatrix} R & \mathbf{0} \\ [\mathbf{t}]_\times R & R \end{bmatrix}, [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \tag{C.4}$$

where $[\mathbf{t}]_\times$ denotes the skew-symmetric matrix corresponding to the translation vector $\mathbf{t}$, and $({}^A\mathbf{l}, {}^A\mathbf{m})$ and $({}^B\mathbf{l}, {}^B\mathbf{m})$ represents the line l in frames $\mathcal{F}_A$ and $\mathcal{F}_B$, respectively [8].

Figure C.4: Real world objects used to evaluate DUST-net's performance. Object classes: microwave, drawer, and toaster (left to right)

C.3.2 Baseline: vm-SoftOrtho

von Mises-Fisher distribution: The von Mises-Fisher distribution (or Langevin distribution) is a unimodal probability distribution on the $(m-1)$ sphere in $\mathbb{R}^m$. A random m-dimensional unit vector $\mathbf{x}$ is said to have the von Mises–Fisher distribution, if its probability distribution function is given by: $f_m(\mathbf{x}|\boldsymbol{\mu}, \kappa) = C_m(\kappa)\exp(\kappa\boldsymbol{\mu}^T\mathbf{x})$, where the concentration parameter $\kappa \geq 0$, the mean direction $\|\boldsymbol{\mu}\| = 1$ and the normalization constant $C_m(\kappa) = \dfrac{\kappa^{\frac{m}{2}-1}}{(2\pi)^{\frac{m}{2}} I_{\frac{m}{2}-1}(\kappa)}$ where I_ν denotes the modified Bessel function of the first kind at order ν [84]. For $m = 3$, the normalization constant reduces to $C_3(\kappa) = \dfrac{\kappa}{4\pi\sinh\kappa} = \dfrac{\kappa\, e^{-\kappa}}{2\pi(1-e^{-2\kappa})}$.

As noted in the main text 7.3, the first baseline method (vm-SoftOrtho) represents the uncertainty over the screw axis orientation vector $\mathbf{l}$ and the direction of moment vector $\hat{\mathbf{m}}$ using two independent von Mises-Fisher distributions. To ensure that the learned distributions respect the Plücker constraint, the method imposes a soft orthogonality constraint over the modes of the two distributions. The distributions over the moment vector magnitude $\|\mathbf{m}\|$ and configurations $q_{1:n-1}$ are considered to be truncated normal distributions.

C.4 Further Results

C.4.1 Accuracy of Point Estimates

Detailed numerical results for the synthetic articulated objects dataset and the PartNet-Mobility dataset are shown in Tables C.1 and C.2, respectively. Results demonstrate that under both metrics, the estimates obtained from DUST-net are considerably more accurate than those obtained from the state-of-the-art methods. DUST-net also correctly estimates very high distribution concentration parameters for the true, noise-free labels. The first

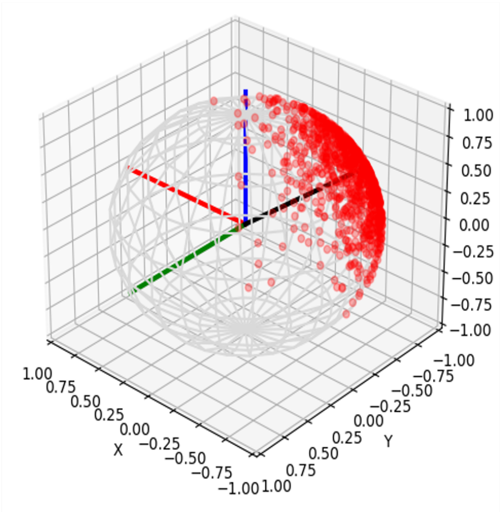


Figure C.5: von Mises-Fisher distribution in $\mathbb{R}^3$. X, Y, Z axes are shown in red, blue and green colors, respectively. Black color represents the mean direction of distribution

baseline, vm-SoftOrtho, performs comparably with DUST-net on both datasets when only MAAD estimates are considered. However, Tables C.1 and C.2 show that it produces a very high distance ($\approx$ 1m) between the predicted and ground-truth screw axes. This error arises due to the soft-orthogonality constraint used by vm-SoftOrtho, as DUST-net and the second baseline method, both of which handle the constraint implicitly, do not report high errors on that metric. Meanwhile, the second baseline, Direct F, performs comparably with DUST-net on both metrics for both datasets, but fails to capture the uncertainty over parameters with the required accuracy.

C.4.2 Uncertainty Estimation

The detailed numerical results from the second set of experiments are shown in Table C.3. In the noiseless case, the singular values of the matrix von Mises-Fisher distribution increases until they reach their maximum allowed value at $\lambda_{max} = 50$, while the precision parameters $\beta_j, j \in \{\|\mathbf{m}\|, \theta, d\}$ for truncated normal distributions over remaining parameters become arbitrarily large.

	MAAD / SL	MAAD		Screw Loss	MAAD	SL	MAAD	SL	Uncertainty				
	$\mathbf{l}$	$\hat{\mathbf{m}}$	$\lVert\mathbf{m}\rVert$	$D(\mathsf{S}_{GT}, \mathsf{S}_{pred})$	θ_i	$\theta_\mathbf{l}$	d_i	$d_\mathbf{l}$	$\lambda_\mathbf{l}$	$\lambda_{\hat{\mathbf{m}}}$	$\beta_{\lVert\mathbf{m}\rVert}$	β_θ	β_d
vm-SoftOrtho	**0.139**	**0.154**	0.068	0.956	**0.012**	**0.117**	0.003	0.006	**56.2**	**55.8**	9.8	47.9	89.5
Direct **F**	0.240	0.261	0.062	0.104	0.010	0.208	0.002	0.006	8.4	7.9	9.8	48.5	75.3
ScrewNet	0.846	0.929	0.486	0.475	0.115	0.217	0.111	0.118	-	-	-	-	-
Abbatematteo et al. [1]	0.194	-	-	0.111	0.223	-	0.045	-	-	-	-	-	-
DUST-net	0.151	0.163	**0.052**	**0.059**	**0.012**	0.122	**0.002**	**0.006**	53.8	54.0	**18.3**	**128.1**	**219.1**
ScrewNet (Local)	0.178	0.443	0.068	0.033	0.057	0.118	0.015	0.015	-	-	-	-	-

Table C.1: Mean error values on the MAAD and Screw Loss(SL) metrics for the simulated articulated objects dataset [1]. Point estimates for DUST-net correspond to the modes of the distributions predicted by DUST-net. Angular values $\{\mathbf{l}, \hat{\mathbf{m}}, \theta_i, \theta_\mathbf{l}\}$ and distances $\{\lVert\mathbf{m}\rVert, D, d_i, d_\mathbf{l}\}$ are reported in radian and meter, respectively. Numerical values are reported for the uncertainty parameters $\{\lambda_i, \beta_j\}$. Symbol $-$ represents value not reported.

	MAAD / SL	MAAD		Screw Loss	MAAD	SL	MAAD	SL	Uncertainty				
	$\mathbf{l}$	$\hat{\mathbf{m}}$	$\lVert\mathbf{m}\rVert$	$D(\mathsf{S}_{GT}, \mathsf{S}_{pred})$	θ_i	$\theta_{\mathbf{l}}$	d_i	$d_{\mathbf{l}}$	$\lambda_{\mathbf{l}}$	$\lambda_{\hat{\mathbf{m}}}$	$\beta_{\lVert\mathbf{m}\rVert}$	β_θ	β_d
vm-SoftOrtho	0.284	0.243	0.221	1.137	0.030	0.086	0.012	0.027	26.9	31.1	5.7	54.5	60.9
Direct **F**	**0.214**	**0.212**	0.257	0.219	0.030	0.064	0.012	**0.024**	8.1	7.3	4.9	59.5	70.9
ScrewNet	0.846	0.929	0.486	0.475	0.115	0.217	0.111	0.118	-	-	-	-	-
Abbatematteo et al. [1]	0.989	-	-	**0.095**	0.141	-	0.085	-	-	-	-	-	-
DUST-net	0.220	0.219	**0.178**	0.189	**0.029**	**0.063**	**0.012**	0.029	**49.3**	**48.3**	**7.7**	**72.0**	**131.9**
ScrewNet (Local)	0.260	1.23	0.314	0.151	0.060	0.106	0.040	0.009	-	-	-	-	-

Table C.2: Mean error values on the MAAD and Screw Loss(SL) metrics for the PartNet-Mobility dataset [20, 93, 149]. Point estimates for DUST-net correspond to the modes of the distributions predicted by DUST-net. Angular values $\{\mathbf{l}, \hat{\mathbf{m}}, \theta_i, \theta_{\mathbf{l}}\}$ and distances $\{\lVert\mathbf{m}\rVert, D, d_i, d_{\mathbf{l}}\}$ are reported in radian and meter, respectively. Numerical values are reported for the uncertainty parameters $\{\lambda_i, \beta_j\}$. Symbol $-$ represents value not reported.

	λ_1	λ_2	$\beta_{\|\mathbf{m}\|}$	β_θ	β_d	λ_1	λ_2	$\beta_{\|\mathbf{m}\|}$	β_θ	β_d	λ_1	λ_2	$\beta_{\|\mathbf{m}\|}$	β_θ	β_d	λ_1	λ_2	$\beta_{\|\mathbf{m}\|}$	β_θ	β_d
Label Noise	No noise					15	15	50	50	50	12	12	50	50	50	10	10	50	50	50
SynArt	53.8	53.9	18.3	128.0	219.0	8.2	8.2	14.6	53.7	51.9	6.8	6.8	10.5	41.6	49.6	3.8	3.8	10.3	41.9	47.4
PartNet	49.3	48.3	7.7	72.0	132.0	6.4	6.3	9.4	29.5	29.2	4.9	4.7	8.9	34.0	37.9	3.2	3.1	9.4	31.2	32.1

Table C.3: Testing variation of DUST-net's confidence over predicted articulation model parameters with input noise. DUST-net's confidence over its predicted parameters decreases monotonically as input noise is increased showing that DUST-net's predicted distribution captures the network's confidence over the predicted articulation parameters effectively.

C.4.3 Real objects

The numerical results from the sim-to-real transfer experiments are shown in Table C.4. Results report that while DUST-net outperforms ScrewNet in estimating the model parameters for real-world objects, the estimated parameters are not yet accurate enough to be used directly for manipulating these objects. However, a noteworthy insight from the results is that DUST-net also reports very low confidence over the predicted parameters. This clearly delineates why it is beneficial to estimate a distribution over the articulation model parameters instead of plain point estimates, as discussed earlier in the section 7.3.3.

		MAAD / SL	MAAD		Screw Loss	MAAD	SL	MAAD	SL	Uncertainty				
		$\mathbf{l}$	$\hat{\mathbf{m}}$	$\lVert\mathbf{m}\rVert$	$D(\mathsf{S}_{GT}, \mathsf{S}_{pred})$	θ_i	$\theta_{\mathbf{l}}$	d_i	$d_{\mathbf{l}}$	$\lambda_{\mathbf{l}}$	$\lambda_{\hat{\mathbf{m}}}$	$\beta_{\lVert\mathbf{m}\rVert}$	β_θ	β_d
Toaster	ScrewNet	2.42	2.48	0.74	0.76	0.45	1.26	0.01	0.00	-	-	-	-	-
Oven	DUST-net	**0.17**	**0.31**	0.52	**0.59**	**0.44**	**0.64**	**0.01**	**0.01**	2.5	0.1	11.6	10.8	75.5
Microwave	ScrewNet	0.79	0.81	**0.13**	0.52	1.19	0.54	0.01	0.01	-	-	-	-	-
	DUST-net	**0.41**	**0.42**	0.22	**0.43**	**0.46**	**0.40**	**0.00**	**0.00**	0.7	0.6	19.7	14.3	39.9
Drawer	ScrewNet	0.69	**0.24**	0.49	**0.24**	**0.72**	0.97	0.08	**0.08**	-	-	-	-	-
	DUST-net	**0.42**	0.50	**0.32**	0.74	0.75	**0.56**	**0.07**	**0.08**	0.2	0.1	12.3	31.6	55.2

Table C.4: Mean error values on the MAAD and Screw Loss metric for estimation of articulation model parameters for real-world objects when network was trained solely using simulated data. Angular values $\{\mathbf{l}, \hat{\mathbf{m}}, \theta_i, \theta_{\mathbf{l}}\}$ and distances $\{\lVert\mathbf{m}\rVert, D, d_i, d_{\mathbf{l}}\}$ are reported in radian and meter, respectively. Numerical values are reported for the uncertainty parameters $\{\lambda_i, \beta_j\}$. Symbol $-$ represents value not reported.

Index

Bibliography

[1] Ben Abbatematteo, Stefanie Tellex, and George Konidaris. Learning to generalize kinematic models to novel objects. In *Proceedings of the Third Conference on Robot Learning*, 2019.

[2] Ali-Akbar Agha-Mohammadi, Suman Chakravorty, and Nancy M Amato. Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research*, 33(2):268–304, 2014.

[3] Ali-akbar Agha-mohammadi, N Kemal Ure, Jonathan P How, and John Vian. Health aware stochastic planning for persistent package delivery missions using quadrotors. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3389–3396. IEEE, 2014.

[4] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[5] Rangaprasad Arun Srivatsan, Mengyun Xu, Nicolas Zevallos, and Howie Choset. Probabilistic pose estimation using a bingham distribution-based linear filter. *The International Journal of Robotics Research*, 37(13-14):1610–1631, 2018.

[6] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[7] Patrick R Barragän, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Interactive bayesian identification of kinematic mechanisms. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2013–2020. IEEE, 2014.

[8] Adrien Bartoli and Peter Sturm. The 3d line motion matrix and alignment of line reconstructions. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[9] Manuel Baum, Matthew Bernstein, Roberto MartÍn-MartÍn, Sebastian Höfer, Johannes Kulick, Marc Toussaint, Alex Kacelnik, and Oliver Brock. Opening a lockbox through physical exploration. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 461–467. IEEE, 2017.

[10] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[11] Miroslav Bogdanovic, Majid Khadiv, and Ludovic Righetti. Learning variable impedance control for contact sensitive tasks. *IEEE Robotics and Automation Letters*, 5(4):6129–6136, 2020.

[12] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[13] Emma Brunskill, Leslie Kaelbling, Tomas Lozano-Perez, and Nicholas Roy. Continuous-State POMDPs with Hybrid Dynamics. *Symposium on Artificial Intelligence and Mathematics*, pages 13–18, 2008.

[14] Adrian Bulat, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. Toward fast and accurate human pose estimation via soft-gated skip connections. In *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*, pages 8–15. IEEE, 2020.

[15] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 173–180. IEEE, 2017.

[16] Arunkumar Byravan, Felix Lceb, Franziska Meier, and Dieter Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor control. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[17] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.

[18] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[19] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision*, pages 608–625. Springer, 2020.

[20] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[21] Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-end learnable geometric vision by backpropagating pnp optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8100–8109, 2020.

[22] Yasuko Chikuse. *Statistics on special manifolds*, volume 174. Springer Science & Business Media, 2003. doi: https://doi.org/10.1007/978-0-387-21540-2.

[23] Yuchen Cui and Scott Niekum. Active reward learning from critiques. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6907–6914. IEEE, 2018.

[24] Andrea F Daniele, Thomas M Howard, and Matthew R Walter. A multiview approach to learning articulated motion models. In *Robotics Research*, pages 371–386. Springer, 2020.

[25] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins. Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7221–7227. IEEE, 2019.

[26] Mehmet R Dogar and Siddhartha S Srinivasa. A planning framework for non-prehensile manipulation under clutter and uncertainty. *Autonomous Robots*, 33(3):217–236, 2012.

[27] Mohamed Elbanhawi and Milan Simic. Sampling-based robot motion planning: A review. *IEEE Access*, 2:56–77, 2014.

[28] Tom Erez and Emanuel Todorov. Trajectory optimization for domains with contacts using inverse dynamics. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4914–4919, 2012. ISBN 9781467317375. doi: 10.1109/IROS.2012.6386181.

[29] Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

[30] Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele. Learning to refine human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 205–214, 2018.

[31] Justin Fu, Sergey Levine, and Pieter Abbeel. One-shot learning of manipulation skills with online dynamics adaptation and neural network priors. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4019–4026. IEEE, 2016.

[32] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the Part: Learning Interaction Strategies for Articulated Object Part Discovery. 2021. URL `http://arxiv.org/abs/2105.01047`.

[33] Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

[34] Igor Gilitschenski, Gerhard Kurz, Simon J Julier, and Uwe D Hanebeck. Unscented orientation estimation based on the bingham distribution. *IEEE Transactions on Automatic Control*, 61(1):172–177, 2015.

[35] Igor Gilitschenski, Roshni Sahoo, Wilko Schwarting, Alexander Amini, Sertac Karaman, and Daniela Rus. Deep orientation uncertainty learning based on a bingham loss. In *International Conference on Learning Representations*, 2019.

[36] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005.

[37] Philip E. Gill, Walter Murray, Michael A. Saunders, and Elizabeth Wong. User's guide for SNOPT 7.6: Software for large-scale nonlinear programming. Center for Computational Mathematics Report CCoM 17-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2017.

[38] Sam Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[39] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306, 2018.

[40] László Gulyás, Gábor Horváth, Tamás Cséri, and George Kampis. An estimation of the shortest and largest average path length in graphs of given density. *arXiv preprint arXiv:1101.2549*, 2011.

[41] Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1025–1037. PMLR, 2020.

[42] J. S. Ha, D. Driess, and M. Toussaint. A probabilistic framework for constrained manipulations and task and motion planning under uncertainty. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6745–6751, 2020. doi: 10.1109/ICRA40945.2020.9196840.

[43] Tsutomu Hasegawa, Takashi Suehiro, and Kunikatsu Takase. A model-based manipulation system with skill-based execution. *IEEE Transactions on Robotics and Automation*, 8(5):535–544, 1992.

[44] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S Sukhatme. Active artic lation model estimation through interactive perception. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3305–3312. IEEE, 2015.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[46] Wei He, Zhijun Li, and C. L. Philip Chen. A survey of human-centered intelligent robots: issues and challenges. *IEEE/CAA Journal of Automatica Sinica*, 4(4):602–609, 2017. doi: 10.1109/JAS.2017.7510604.

[47] Neville Hogan. Impedance control: An approach to manipulation: Part i—theory. 1985.

[48] Geoffrey A Hollinger and Gaurav S Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.

[49] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[50] David Hsu, Wee Sun Lee, and Nan Rong. What makes some pomdp problems easy to approximate? In *NIPS*, pages 689–696, 2007.

[51] Vadim Indelman, Luca Carlone, and Frank Dellaert. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7):849–882, 2015.

[52] Yuki Inoue, Shinzo Kitamura, and Yutaka Kidawara. Force feedback control and collision avoidance of redundant manipulator. In *Proceedings IROS'91: IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91*, pages 149–152. IEEE, 1991.

[53] Advait Jain and Charles C Kemp. Pulling open novel doors and drawers with equilibrium point control. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 498–505. IEEE, 2009.

[54] Ajinkya Jain and Scott Niekum. Belief space planning under approximate hybrid dynamics. In *Proceedings of the Robotics: Science and Systems (RSS) Workshop on POMDPs in Robotics*, 2017.

[55] Ajinkya Jain and Scott Niekum. Efficient hierarchical robot motion planning under uncertainty and hybrid dynamics. In *Conference on Robot Learning*, pages 757–766, 2018.

[56] Ajinkya Jain and Scott Niekum. Learning hybrid object kinematics for efficient hierarchical planning under uncertainty. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[57] Ajinkya Jain, Stephen Giguere, Rudolf Lioutikov, and Scott Niekum. Distributional depth-based estimation of object articulation models. *arXiv preprint arXiv:2108.05875*, 2021.

[58] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[59] Yan-bin Jia. Plücker Coordinates for Lines in the Space [Lecture Notes]. pages 1–11, August 2019.

[60] Lin Jiu and Christoph Koutschan. Calculation and properties of zonal polynomials. *Mathematics in Computer Science*, pages 1–18, 2020.

[61] R Platt Jr, Russ Tedrake, Leslie Kaelbling, and Tomas Lozano-Perez. Belief space planning assuming maximum likelihood observations. *Robotics: Science and Systems*, 2010. ISSN 2330765X.

[62] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[63] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277. IEEE, 2008.

[64] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 5003–5010. IEEE, 2013.

[65] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer, 2006.

[66] Oliver Kroemer, Christian Daniel, Gerhard Neumann, Herke Van Hoof, and Jan Peters. Towards learning hierarchical skills for multi-phase manipulation tasks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1503–1510. IEEE, 2015.

[67] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *arXiv preprint arXiv:1907.03146*, 2019.

[68] Hanna Kurniawati and Vinay Yadav. An online pomdp solver for uncertainty planning in dynamic environment. In *Robotics Research*, pages 611–629. Springer, 2016.

[69] Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, volume 2008. Zurich, Switzerland, 2008.

[70] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[71] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 156–163. IEEE, 2015.

[72] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020.

[73] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[74] Zhan Wei Lim, David Hsu, and Wee Sun Lee. Monte carlo value iteration with macro-actions. In *NIPS*, pages 1287–1295, 2011.

[75] Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088*, 2020.

[76] Yizhou Liu, Fusheng Zha, Lining Sun, Jingxuan Li, Mantian Li, and Xin Wang. Learning articulated constraints from a one-shot demonstration for robot manipulation planning. *IEEE Access*, 7:172584–172596, 2019.

[77] Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. Lstm pose machines. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5207–5215, 2018.

[78] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.

[79] John Lygeros, Shankar Sastry, and Claire Tomlin. Hybrid systems: Foundations, advanced topics and applications. *under copyright to be published by Springer Verlag*, 2012.

[80] Omid Madani, Steve Hanks, and Anne Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable markov decision problems. In *AAAI/IAAI*, pages 541–548, 1999.

[81] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982, 2017.

[82] Osama Makansi, Eddy Ilg, Ozgun Cicek, and Thomas Brox. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7144–7153, 2019.

[83] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019.

[84] Kanti V Mardia and Peter E Jupp. *Directional statistics*, volume 494. John Wiley & Sons, 1999. doi: 10.1002/9780470316979.

[85] Roberto Martin Martin and Oliver Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2494–2501. IEEE, 2014.

[86] Roberto Martín-Martín and Oliver Brock. Coupled recursive estimation for online interactive perception of articulated objects. *The International Journal of Robotics Research*, page 0278364919848850, 2019.

[87] Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5091–5097. IEEE, 2016.

[88] Roberto Martín-Martín, Michelle A Lee, Rachel Gardner, Silvio Savarese, Jeannette Bohg, and Animesh Garg. Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1010–1017. IEEE, 2019.

[89] Matthew T Mason. *Mechanics of robotic manipulation*. MIT press, 2001.

[90] Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *BMVC*, pages 181–1, 2015.

[91] Toki Migimatsu and Jeannette Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5(2):844–851, 2020.

[92] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, pages 405–421. Springer, 2020.

[93] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[94] George E Monahan. State of the art—a survey of partially observable markov decision processes: theory, models, and algorithms. *Management science*, 28(1):1–16, 1982.

[95] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[96] Igor Mordatch, Emanuel Todorov, and Zoran Popović. Discovery of complex behaviors through contact-invariant optimization. *ACM Transactions on Graphics*, 31(4), 2012. ISSN 07300301. doi: 10.1145/2185520.2185539.

[97] Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon markov decision process problems. *Journal of the ACM (JACM)*, 47(4):681–720, 2000.

[98] Kieran Murphy, Carlos Esteves, Varun Jampani, Srikumar Ramalingam, and Ameesh Makadia. Implicit representation of probability distributions on the rotation manifold. In *International Conference on Machine Learning*, 2021.

[99] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[100] Scott Niekum, Sarah Osentoski, Christopher G Atkeson, and Andrew G Barto. Online bayesian changepoint detection for articulated motion models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1468–1475. IEEE, 2015.

[101] Joni Pajarinen and Ville Kyrki. Robotic manipulation of multiple objects as a pomdp. *Artificial Intelligence*, 247:213–228, 2017.

[102] Subhadip Pal, Subhajit Sengupta, Riten Mitra, and Arunava Banerjee. Conjugate priors and posterior inference for the matrix langevin distribution on the stiefel manifold. *Bayesian Analysis*, 15:871–908, 2020. ISSN 19316690. doi: 10.1214/19-BA1176.

[103] Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[104] Keunhong Park, Arsalan Mousavian, Yu Xiang, and Dieter Fox. Latentfusion: End-to-end differentiable reconstruction and rendering for unseen object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10710–10719, 2020.

[105] Karl Pauwels and Danica Kragic. Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 2015.

[106] Jana Pavlasek, Stanley Lewis, Karthik Desingh, and Odest Chadwicke Jenkins. Parts-based articulated object localization in clutter using belief propagation. *arXiv preprint arXiv:2008.02881*, 2020.

[107] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

[108] Valentin Peretroukhin, Matthew Giamou, David M Rosen, W Nicholas Greene, Nicholas Roy, and Jonathan Kelly. A smooth representation of belief over so (3) for deep rotation learning with uncertainty. *Robotics: Science and Systems*, 2020.

[109] Claudia Pérez-D'Arpino and Julie A Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4058–4065. IEEE, 2017.

[110] Martin Pfanne, Maxime Chalon, Freek Stulp, Helge Ritter, and Alin Albu-Schäffer. Object-level impedance control for dexterous in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(2):2987–2994, 2020.

[111] Sudeep Pillai, Matthew R Walter, and Seth Teller. Learning articulated motions from visual demonstration. *arXiv preprint arXiv:1502.01659*, 2015.

[112] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Policy-contingent abstraction for robust robot control. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 477–484. Morgan Kaufmann Publishers Inc., 2002.

[113] Michael Posa and Russ Tedrake. Direct trajectory optimization of rigid body dynamical systems through contact. In *Algorithmic foundations of robotics X*, pages 527–542. Springer, 2013.

[114] Michael Posa, Scott Kuindersma, and Russ Tedrake. Optimization and stabilization of trajectories for constrained dynamical systems. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1366–1373. IEEE, 2016.

[115] Sergey Prokudin, Peter Gehler, and Sebastian Nowozin. Deep directional statistics: Pose estimation with uncertainty quantification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.

[116] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. doi: 10.15607/RSS.2018.XIV.049.

[117] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020.

[118] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.

[119] Guy Shani, Joelle Pineau, and Robert Kaplow. A survey of point-based pomdp solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

[120] Guanya Shi, Yifeng Zhu, Jonathan Tremblay, Stan Birchfield, Fabio Ramos, Animashree Anandkumar, and Yuke Zhu. Fast uncertainty quantification for deep object pose estimation. *arXiv preprint arXiv:2011.07748*, 2020.

[121] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.

[122] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Advances in neural information processing systems*, pages 2164–2172, 2010.

[123] Trey Smith and Reid Simmons. Heuristic search value iteration for pomdps. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 520–527, 2004.

[124] Trey Smith and Reid Simmons. Point-based pomdp algorithms: improved analysis and implementation. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 542–549, 2005.

[125] Adhiraj Somani, Nan Ye, David Hsu, and Wee Sun Lee. Despot: Online pomdp planning with regularization. *Advances in neural information processing systems*, 26: 1772–1780, 2013.

[126] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020.

[127] Koushil Sreenath, Connie R Hill Jr, and Vijay Kumar. A partially observable hybrid system model for bipedal locomotion for adapting to terrain variations. In *Proceedings of the 16th international conference on Hybrid systems: computation and control*, pages 137–142. ACM, 2013.

[128] Rangaprasad Arun Srivatsan, Gillian T Rosen, D Feroze Naina Mohamed, and Howie Choset. Estimating se (3) elements using a dual quaternion based linear kalman filter. In *Robotics: Science and systems*, 2016.

[129] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4): 341–359, 1997.

[130] Jürgen Sturm, Vijay Pradeep, Cyrill Stachniss, Christian Plagemann, Kurt Konolige, and Wolfram Burgard. Learning kinematic models for articulated objects. In *IJCAI*, pages 1851–1856, 2009.

[131] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011.

[132] Guru Subramani, Michael Gleicher, and Michael Zinn. Recognizing geometric constraints in human demonstrations using force and position signals. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[133] Guru Subramani, Michael Zinn, and Michael Gleicher. Inferring geometric constraints in human demonstrations. *arXiv preprint arXiv:1810.00140*, 2018.

[134] Yu Sun, Shaogang Ren, and Yun Lin. Object–object interaction affordance learning. *Robotics and Autonomous Systems*, 62(4):487–496, 2014.

[135] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, and Rudolph Triebel. Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. *International Journal of Computer Vision*, 128(3):714–729, 2020.

[136] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *arXiv preprint arXiv:1807.03146*, 2018.

[137] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In Alexandre M. Bayen, Ali Jadbabaie, George Pappas, Pablo A. Parrilo, Benjamin Recht, Claire Tomlin, and Melanie Zeilinger, editors, *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 708–717, The Cloud, 10–11 Jun 2020. PMLR. URL `http://proceedings.mlr.press/v120/taylor20a.html`.

[138] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[139] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[140] Philip HS Torr and Andrew Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Computer vision and image understanding*, 78(1): 138–156, 2000.

[141] Marc Toussaint and Manuel Lopes. Multi-bound tree search for logic-geometric programming in cooperative manipulation domains. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4044–4051. IEEE, 2017.

[142] Marc Toussaint, Laurent Charlin, and Pascal Poupart. Hierarchical pomdp controller optimization by likelihood maximization. In *UAI*, volume 24, pages 562–570, 2008.

[143] Marc Toussaint, Kelsey R Allen, Kevin A Smith, and Joshua B Tenenbaum. Differentiable Physics and Stable Modes for Tool-Use and Manipulation Planning. *Robotics: Science and Systems*, 2018. URL `http://www.roboticsproceedings.org/rss14/p44.pdf`.

[144] Jur Van Den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012.

[145] Nikos Vlassis, Michael L Littman, and David Barber. On the computational complexity of stochastic controller optimization in pomdps. *ACM Transactions on Computation Theory (TOCT)*, 4(4):12, 2012.

[146] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.

[147] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019.

[148] Bowen Wen, Chaitanya Mitash, Baozhang Ren, and Kostas E Bekris. se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains. *arXiv preprint arXiv:2007.13866*, 2020.

[149] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[150] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *2018 Robotics: Science and Systems*, 2018.

[151] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 793–802, 2019.

[152] Duo Xu, Mohit Agarwal, Faramarz Fekri, and Raghupathy Sivakumar. Accelerating reinforcement learning agent with eeg-based implicit human feedback. *arXiv preprint arXiv:2006.16498*, 2020.

[153] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. Rpm-net: Recurrent prediction of motion and parts from point cloud. *ACM Trans. Graph.*, 38(6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356573. URL `https://doi.org/10.1145/3355089.3356573`.

[154] Lu Yang, Qing Song, Zhihui Wang, and Ming Jiang. Parsing r-cnn for instance-level human analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 364–373, 2019.

[155] Li Yi, Haibin Huang, Difan Liu, Evangelos Kalogerakis, Hao Su, and Leonidas Guibas. Deep part induction from articulated object pairs. In *SIGGRAPH Asia 2018 Tech. Pap. SIGGRAPH Asia 2018*, volume 37, 2018. doi: 10.1145/3272127.3275027. URL `https://dl.acm.org/doi/10.1145/3272127.3275027`.

[156] Vicky Zeng, Timothy E Lee, Jacky Liang, and Oliver Kroemer. Visual identification of articulated object parts. *arXiv preprint arXiv:2012.00284*, 2020.

[157] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020.

[158] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *arXiv preprint arXiv:2012.13392*, 2020.