

© 2023 Yifan Zhu

DATA-EFFICIENT LEARNING FOR MANIPULATION, LOCOMOTION, AND
INFORMATION GATHERING INVOLVING GRANULAR MEDIA AND
DEFORMABLE OBJECTS

BY

YIFAN ZHU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2023

Urbana, Illinois

Doctoral Committee:

Professor Kris Hauser, Chair
Assistant Professor Katie Driggs-Campbell
Professor David Forsyth
Associate Professor Dmitry Berenson, University of Michigan
Professor Ken Goldberg, University of California, Berkeley

ABSTRACT

Models of robots and how they contact the external world is traditionally built based on physics. However, such an approach is limited when the physics of certain phenomena are not well understood, when it is computationally prohibitive to solve for the equations, and when identifying equation parameters and solving conditions is challenging in the real world with partial and noisy observations. Recently, advancements in deep learning have provided a potential way to deal with this challenge, leveraging extremely flexible function approximators such as neural networks. However, the data required for many common robotics tasks could be prohibitive due to the complexity of the physics involved.

This thesis aims to make progress toward addressing the issue of data efficiency for complex physics phenomena such as granular media, heterogeneous deformable objects, and acoustics of human bodies. To this end, this thesis adopts two main methodologies. First, a gray-box learning approach where learning is tightly integrated with physics, is employed to improve data efficiency. The core idea here is to decompose physics into parts that can be described by efficient analytical equations, and parts that are poorly understood or computationally heavy, which are learned from data. In this thesis, I will demonstrate different ways of combining knowledge of physics and learning to achieve data efficiency on multiple challenging problems. The second methodology aims to use meta-learning, or learning to learn, to extract useful prior knowledge from offline data on related tasks to accelerate online learning on novel tasks. I will demonstrate a novel meta-learning technique that enables a robot to use vision and very little online experience to achieve high-quality scooping actions on out-of-distribution granular terrains. We further show that these two methodologies can complement each other by demonstrating that the proposed meta-learning algorithm can improve gray-box learning for deformable objects.

In addition to these two main methodologies, I also discuss my other relevant efforts in solving contact-rich robotics tasks, including automated excavation and manipulation in unstructured environments with an immersive, novice-friendly avatar robot that achieved 4-th place in the ANA XPRIZE Avatar Challenge.

ACKNOWLEDGMENTS

This incredible journey of doing a Ph.D. is finally coming to an end. Looking back at the past 6 years of graduate school, it is clear to me that the amazing people that I have had the fortune to share the journey together are the most important reason for my achievements.

First and foremost, I would like to express gratitude toward my advisor Kris Hauser. Kris is one of the smartest minds that I know, and he offered countless insightful advice from the technical level to the high level of research, and on communication and strategic planning throughout my Ph.D. career. At the same time, his level of high standards has always pushed me to better myself, and will forever be a fortune to me.

I would also like to thank my thesis committee and other faculties that I have collaborated with: thank you to Ken Goldberg, David Forsyth, Dmitry Berenson, Katie Driggs-Campbell, Melkior Ornik, and Daniel Buckland for your guidance and support!

I also owe many thanks to the amazing peers that I have worked or collaborated with. We not only had numerous discussions on research but also had lots of fun together outside of the lab. Thank you to Fan Wang, Gao Tang, Shihao Wang, Hayden Bader, Weidong Sun, Zherong Pan, Kai Lu, Yu Zhou, Tianzhen Li, Yeonju Kim, Patrick Naughton, Joao Marques, Mengchao Zhang, Pranay Thangeda, William Edwards, James Nam, Pusong Li, Al Smith, Shaoxiong Yao, Yixuan Wang, Jing-chen Peng, and Roger Qiu.

I am lucky to have been able to intern at Baidu USA, and I would like to thank the amazing people that I worked with. Thank you to my internship supervisor Liangjun Zhang, and my colleagues Qingkai Lu, Zhixian Ye, Liyang Wang, and Daming Lu.

My other close friends are able to give me some work-life balance during graduate school and I am grateful for that. Thank you to Zhichong Zhu, Wesley Lin, Vikas Shiva, Sanjid Halim, John Tian, Stella Zhang, Jeffrey Yang, Ziwei Zhang, Wenchao Zhu, Haipei Zhu, Xusheng Luo, Haoran Qiu, Jiaqi Guan, Dawei Sun, and Weichao Mao!

I also like to offer many thanks to my girlfriend Siqi Lai for your love and support. Not only did you support me emotionally, but you also offered me valuable help and insights as an academic from a different discipline.

Last but not least I want to thank my parents Chenhai Zhu and Shaoye Song for their unconditional love, who made everything better when times got hard.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION	1
1.1 Gray-Box Learning	3
1.2 Meta-Learning	5
1.3 Dissertation Structure	7
1.4 Summary of Contribution	9
 CHAPTER 2 GRAY-BOX LEARNING FOR FAST SIMULATION OF ROBOT LOCOMOTION ON GRANULAR MEDIA	10
2.1 Introduction	10
2.2 Related Work	11
2.3 Methods	13
2.4 Experiments and Results	18
2.5 Conclusion	22
 CHAPTER 3 CONTACT-IMPLICIT TRAJECTORY OPTIMIZATION WITH LEARNED DEFORMABLE CONTACTS USING BILEVEL OPTIMIZATION	23
3.1 Introduction	23
3.2 Related Work	25
3.3 Method	26
3.4 Results	32
3.5 Conclusion	36
 CHAPTER 4 GRAY-BOX LEARNING FOR SIMULATION OF FORCE RE- SPONSE MODELS FOR HETEROGENEOUS ELASTIC OBJECTS	37
4.1 Introduction	37
4.2 Related Work	39
4.3 Point Deformation Model Learning	40
4.4 Gray-Box Simulator	44
4.5 Conclusion	49
 CHAPTER 5 AUTOMATED HEART AND LUNG AUSCULTATION IN ROBOTIC PHYSICAL EXAMINATIONS	51
5.1 Introduction	51
5.2 Related Work	53
5.3 Method	54
5.4 Results	62
5.5 Conclusion and Future Work	68

CHAPTER 6	FEW-SHOT ADAPTATION FOR MANIPULATING GRANULAR MATERIALS AND LEARNING FORCE RESPONSE MODELS OF SOFT OBJECTS	69
6.1	Introduction	69
6.2	Related Work	70
6.3	Proposed Method	72
6.4	Granular Media Manipulation	75
6.5	Deformable Object Force Response Model Learning	88
6.6	Conclusion and Future Work	93
CHAPTER 7	EXCAVATION OF FRAGMENTED ROCKS WITH MULTI-MODAL MODEL-BASED REINFORCEMENT LEARNING	94
7.1	Introduction	94
7.2	Literature Review	96
7.3	Problem Definition	97
7.4	Dynamics Learning	99
7.5	Planning	101
7.6	Results	102
7.7	Conclusion	108
CHAPTER 8	IMMERSIVE COMMODITY TELEPRESENCE WITH THE TRINA ROBOT AVATAR	109
8.1	Introduction	109
8.2	Related Work	111
8.3	System Overview	112
8.4	Manipulators	115
8.5	Hand Control	124
8.6	End-to-End Evaluation: ANA Avatar XPRIZE	129
8.7	Discussion and Lessons Learned	132
8.8	Conclusion	134
CHAPTER 9	CONCLUSION	135
9.1	Future Directions	135
9.2	Discussion	136
REFERENCES	138

CHAPTER 1: INTRODUCTION

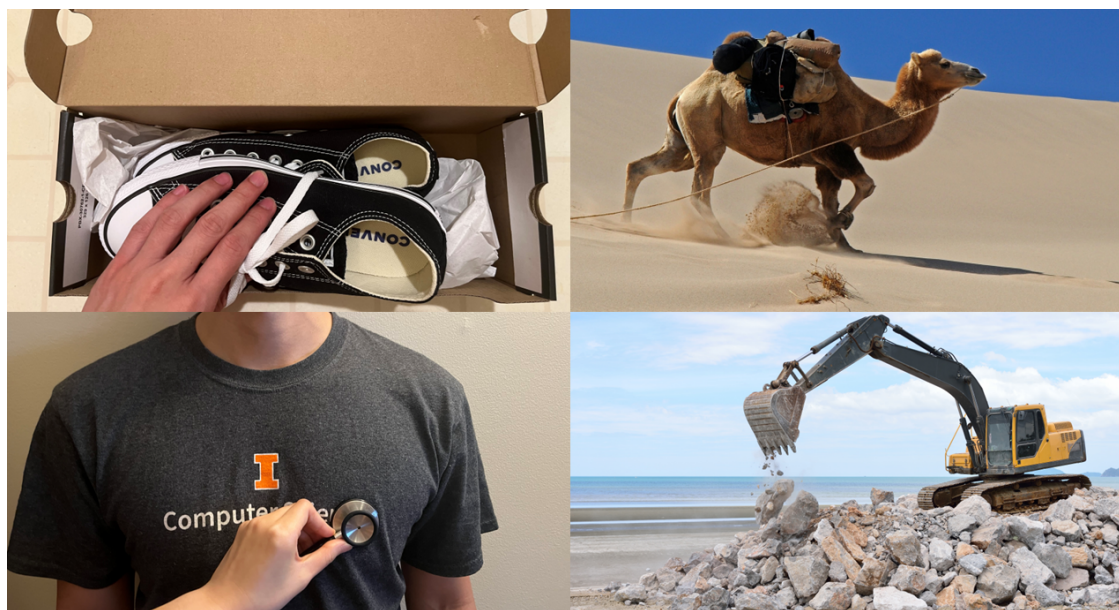


Figure 1.1: The contact-rich tasks tackled in this thesis, which involve various kinds of deformable objects and environments. Top left: manipulation of heterogeneous deformable objects. Top right: locomotion over granular media. Bottom left: auscultation. Bottom right: fragmented rocks manipulation.

One of the key components of a robot system is a predictive model of the world, which is essential for state estimation, planning, and control. The traditional approach builds models of the dynamics of robots and how they would contact with the surrounding environment based on physics [1]. As summarized in Fig. 1.2, while this approach requires little data, it needs an engineer to manually specify analytical models for each different robot task. This approach is also limited when the task at hand involves physics phenomena that are not well understood, with accurate analytical equations not available [2, 3, 4]. Even when analytical equations are available, it could be computationally prohibitive for them to be useful in robotics applications. For instance, Tasora et al. proposed a method to simulate vehicle locomotion on granular media [5], but it costs 3.2 hours of computation on a 20-core processor to perform 1 s of simulation. In addition, solving the inverse problem, i.e. identifying equation parameters and solving conditions, could be very challenging in the real world [6, 7].

Recently, advancements in deep learning have provided a potential way to deal with these challenges, leveraging extremely flexible function approximators such as neural networks [8]. However, due to the complexity of physical phenomena in the real world, large amounts of data are required for many common contact-rich robotics tasks when applying black-

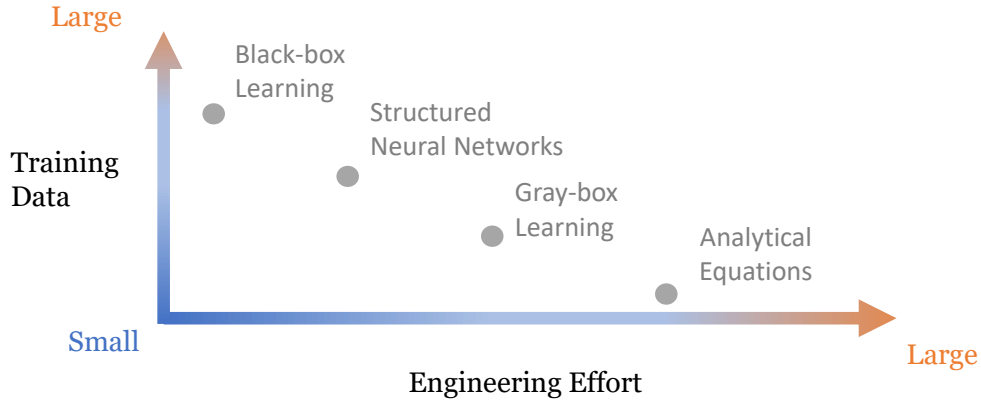


Figure 1.2: An overview of different types of methods and their requirements for training data vs. engineering effort.

box deep learning [9]. This limits the applicability of black-box deep learning as data are expensive and even dangerous to collect in a lot of robotics applications. To mitigate this limitation, structured neural networks aim to improve the data efficiency of neural networks by designing model architecture that enforces physics concepts [10, 11, 12]. For example, Li et al. [11] propose to use graph neural networks (GNNs) to learn the particle dynamics of rigid, granular materials, and fluids, where the GNNs enforce the physics concept that the physical interactions between the particles happen locally. However, such methods still need a large amount of data, making them mostly applicable for simulations and simulated data. To further improve data efficiency, this thesis explores a class of techniques called gray-box learning [13] that instead of embedding physics concepts, tightly couples learning and analytical equations of physics. This class of methods has been widely adopted in science and engineering, and this thesis demonstrates the first application of these techniques to contact-rich tasks with soft objects and environments as those shown in Fig. 1.1 that lead to computationally efficient simulations learned from a small amount of real-world data.

While gray-box learning leverages analytical equations to improve data efficiency, such a method is not always possible. For example, for an excavator manipulating fragmented rocks in Fig. 1.1, the dynamics of the bucket-rock interaction is highly dependent on the latent arrangement, geometry, and physics properties of rock particles, making embedding physics equations extremely challenging. In addition, the large amounts of data required to train black-box deep learning models might not be available for lots of robotics tasks. Therefore, this thesis explores meta-learning, or learning-to-learn [14], that achieves faster learning by extracting useful prior for the task at hand from available data on relevant tasks. In addition, meta-learning can also complement gray-box learning by improving data efficiency given data

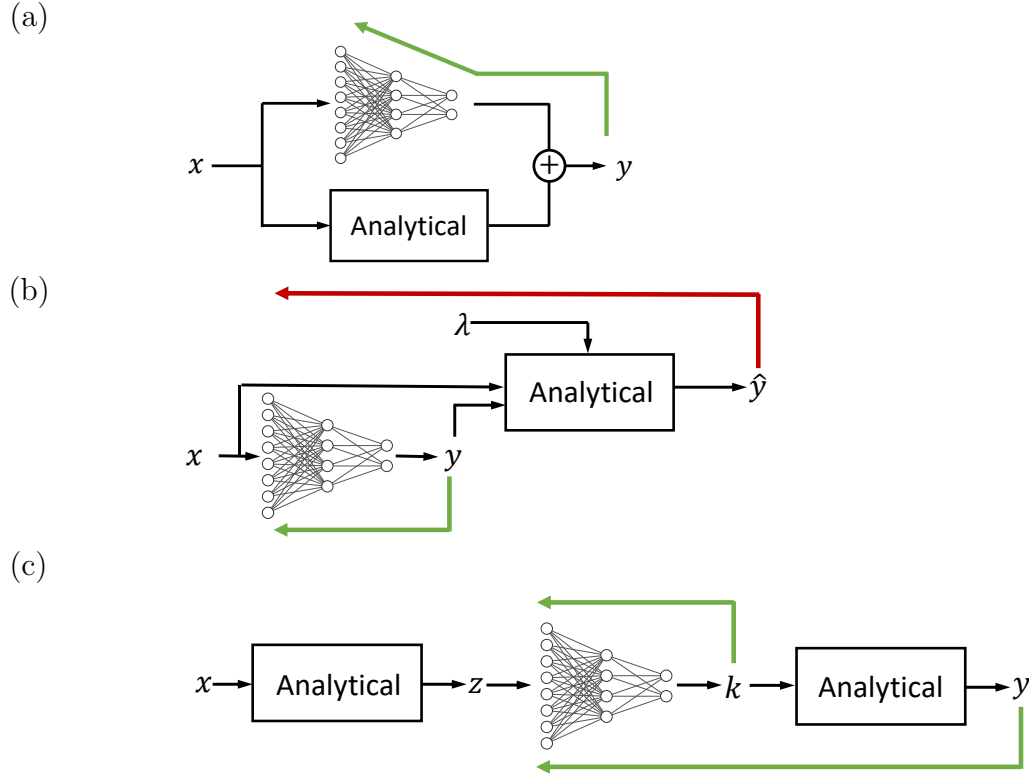


Figure 1.3: Different classes of methods for gray-box learning. The gradient flows during training are marked in green and red. (a) The machine learning model learns the residual of an analytical model. (b) Physics-informed Learning: use analytical equations to guide the training of the machine learning models, and performs joint learning of the machine learning model and the analytical equation coefficients λ . There are two gradient flows during training: the training loss includes data loss for the training data (green), and also the analytical equation violations for arbitrary data points (red). (c) Learning simulator component: a component of the simulator is learned from data. There exist two different types of training for different methods (two different gradient flows), depending on the smoothness of the analytical equations. [Best viewed in color.]

from closely related tasks, which is also demonstrated in this thesis.

Gray-box learning and meta-learning are introduced in more detail in the next two sections.

1.1 GRAY-BOX LEARNING

The idea of gray-box learning is to enhance the data efficiency of black-box learning methods by tightly combining learning and analytical equations based on physics. Here I provide a taxonomy of gray-box learning methods and discuss their applicability. The three major classes of gray-box learning methods are illustrated in Fig. 1.3, and their applicability, pros, and cons are summarized in Tab. 1.1.

Method	Applicability	Pros	Cons
Residual Learning	Efficient and reasonably accurate analytical equations are available, and so are the parameters of these equations	<ul style="list-style-type: none"> • Simple to implement 	<ul style="list-style-type: none"> • Applicable to a smaller range of problems compared to other methods
Physics-informed Learning	The form of governing PDEs of the underlying physics phenomenon is known, and only partial and sparse measurements are available.	<ul style="list-style-type: none"> • Can handle partial and sparse training data • Identifies analytical equation coefficients 	<ul style="list-style-type: none"> • Computationally heavy due to joint training of a neural network and PDE parameters, especially when high resolution is required.
Learn Simu- lator Compon- ent	The underlying physics includes parts that can be described by efficient and accurate analytical equations with known parameters.	<ul style="list-style-type: none"> • Does not require full knowledge of the physics 	<ul style="list-style-type: none"> • Requires more manual design of the model

Table 1.1: Summary of the different classes of gray-box learning methods.

Residual Learning The most straightforward way to leverage analytical equations in learning is to perform residual learning, where a machine learning model learns the residuals of an analytical models [15, 16, 17, 18, 19]. This requires that there already exist efficient and reasonably accurate analytical equations with known parameters and learning is used to correct for small biases and noises from real-world observations. While the implementation of these methods is straightforward, their application is limited. In Chapter 5, I adopt this approach and propose a semi-parametric residual Gaussian process model to model the acoustic properties of human patients leveraging the knowledge of human anatomy, which improves the data efficiency over a pure Gaussian process model [20]. This method alleviates the requirement of the equation parameters by additionally estimating them from online observations.

Physics-informed Learning This class of methods uses analytical equations to guide the training of the machine learning models [21, 22, 23, 24, 25], and is a popular technique in the field of Physics [21] where the analytical equations are in the form of partial differential equations (PDEs). As shown in Fig. 1.3, this class of methods learns the machine learning model and the analytical equation coefficients jointly during training. The training loss includes data loss for the collected training data, and also the analytical equation violations for arbitrary data points. As a result, physics-informed learning (PIL) can handle partial and scattered observations. In addition, PIL only requires that the form of the governing equations is known and that they are differentiable. However, when the equations are com-

putationally heavy to compute or scale poorly with the desired resolution, joint training with a neural network becomes computationally prohibitive.

Learn Simulator Component The last major class of methods learns the component of a simulator. In the context of robotics, this component is usually the contact model [3, 26, 27, 28], which is motivated by the fact that the contact dynamics between rigid objects are not well-understood while the contact-free dynamics is known very well. Therefore, the contact model instead is learned from data, either in an end-to-end manner [3, 27, 28] or just from the input and output data of the contact model [26], depending on whether the rest of the simulator is differentiable. The core idea here is to decompose physics into parts that can be described by efficient analytical equations, and parts that are poorly understood or computationally heavy, which are learned from data. Compared to PIL and residual learning, this class of methods does not require that the underlying governing equations are fully known, and can use learning to replace analytical equations that are computationally heavy. This class of methods is the most flexible but requires an engineer to carefully design the pipeline. While the literature that adopts this idea usually deals with the case of rigid body dynamics with Coulomb friction, in this thesis, I will demonstrate different ways of combining analytical simulators and learning to achieve data efficiency on problems that involve complicated physics phenomena such as granular media, heterogeneous deformable objects, and acoustics of human bodies, shown in Fig. 1.1.

1.2 META-LEARNING

Meta-learning was first proposed in the literature in 1987 [29], and by 1998 a book is written on the topic [14]. In the contemporary era, the development of meta-learning has mainly been associated with end-to-end deep learning, due to its potential to alleviate the drawbacks of deep learning, such as improving data efficiency and transferrability. The concept of meta-learning is particularly attractive in robotics, where an autonomous agent’s ability to master new skills quickly is a hallmark of intelligence. The development of meta-learning has mainly been spurred by computer vision, and in particular image classification [30], but there has been an increasing amount of interest in the robotics community, especially for the sequential decision-making setting.

The goal of meta-learning is to enable fast adaptation of a model during testing on a target task by training the model with offline data on similar tasks first. These are also referred to as the *meta-testing* and *meta-training* phases in the literature. To accomplish this goal, during *meta-training*, the idea of meta-learning is to perform simulated testing using data on

offline tasks by treating each offline task as a task the model is adapting to and find model hyperparameters that maximize the speed of adaptation. Below I first present a taxonomy of meta-learning methods in the literature.

Metric-based methods. The idea of metric learning resembles that of nearest neighbors algorithms. In metric-based meta-learning, the aim is to learn good metric or distance functions so that they will generalize to new tasks [31, 32, 33]. This class of methods is mainly motivated by image classification and is not suitable for problems considered in this thesis, which involves regression.

Optimization-based methods. This class of meta-learning methods aims to learn how to perform optimization during the online testing stage for fast adaptation, including learning the initial model parameters [34, 35], learning the model weights updates in each iteration [36, 37], directly predicting the model weights [38, 39], and learning the optimizer hyperparameters [40]. Perhaps the most famous meta-learning algorithm is the MAML algorithm [34], which optimizes for the initial weights of a neural network such that the neural network, after finetuning on the data in the new task, generalizes well. This class of methods usually applies to neural networks. However, these methods are prone to overfitting in the few-shot regime [41], especially when the robotics tasks considered in this thesis involve very sparse support data points in the spatial dimension and prediction needs to be made on the entire spatial dimension.

Kernel-based methods. Gaussian processes (GPs) is a popular non-parametric method for learning in the small-data regime. However, GPs do not naturally handle high-dimensional inputs and deep kernels [42] have been proposed to instead pass inputs through a neural network first to a low-dimensional feature before inputting into a GP. Deep kernel methods for meta-learning [43, 44, 45] tunes the kernel parameters to work well for each of the training tasks. One drawback of this class of methods is that they do not scale very well to large data, but this is not a concern for the tasks considered in this thesis.

Modular Learning Another class of methods is modular learning [46, 47]. In these methods, modules learned for different tasks are separately learned and stored, and on novel tasks, the learned modules are assembled to maximize performance. However, this class of methods is largely unexplored in the literature and does not show superior performance compared to other methods.

1.3 DISSERTATION STRUCTURE

This thesis aims to make progress toward tackling the challenge of learning for robots that make contact with complicated objects and environments by exploring techniques that improve the *data efficiency* of learning algorithms. Examples of these objects and environments are shown in Fig. 1.1, including granular media, heterogeneous 3D deformable objects, and the acoustics of human bodies.

In Chapter 2 and 4, I adopt gray-box learning and propose simulators with learned contact models for granular media and heterogeneous deformable objects. In Chapter 2, I propose a novel data-driven granular contact model based on the stick-slip behavior between rigid objects and granular media, which can be learned with a few hundred real-world data points. It is then combined with rigid body dynamics in an optimization-based simulator based on the Maximum Dissipation Principle. I evaluate the proposed method on three different types of granular media and show that the simulator simulates plausible interactions between robots and granular media in real time, orders of magnitude faster than state-of-the-art analytical methods. Chapter 3 extends this simulator to be a differentiable one by solving it with a primal interior-point method with a pre-specified duality gap. This chapter then proposes a bilevel, contact-implicit trajectory optimization formulation that searches for long-horizon robot trajectories on granular terrains, where the lower-level solves the contact force with the differentiable simulator to obtain the dynamics constraints, and the upper-level trajectory optimization problem is solved with direct transcription. I evaluate the proposed method by optimizing locomotion trajectories of a quadruped robot on various granular terrains offline and show that it can produce long-horizon walking gaits of high quality, while standard trajectory optimization and reinforcement learning methods can not. These two chapters are originally published as [48, 49].

Chapter 4 proposes a point-based surface representation and a heterogeneous, nonlinear point force response model learned from a robotic arm acquiring force-displacement curves from a few poking interactions. They are then used in a simulator that solves for deformable object surface deformation and contact wrench when contacting arbitrarily-shaped rigid objects based on the Hertzian contact model. I evaluate this method on a variety of challenging soft objects, and show that the proposed approach learns force response with sufficient accuracy to generate plausible contact responses for novel rigid objects. This chapter is originally published as [50].

In Chapters 2 and 4, learning happens in a supervised manner, where a small labeled dataset is provided to a machine learning algorithm. Chapter 5, instead, demonstrates an informative path-planning approach, where the robot plans information-gathering paths to

collect data online. Specifically, this chapter proposes the first automated system for the task of heart and lung auscultation. To select auscultation locations that generate high-quality sounds, a Bayesian Optimization formulation adopts a gray-box semi-parametric Gaussian process model that leverages visual anatomical cues to predict where high-quality sounds might be located, while using auditory feedback to adapt to patient-specific anatomical qualities. I evaluate the system by experimenting with 4 human subjects and show that the system autonomously captures heart and lung sounds of similar quality compared to teleoperation by a human trained in clinical auscultation. This chapter is originally published as [20].

Chapter 6 presents a vision-based few-shot meta-learning method with a deep kernel model trained with meta-learning to learn online from very limited experience on novel tasks. It introduces a novel meta-training approach, Deep Meta-Learning with Controlled Deployment Gaps (CoDeGa), that explicitly trains the deep kernel to make predictions robustly under large domain shifts. This chapter applies this method to applications: few-shot scooping and gray-box learning of deformable object force response models which is first introduced in Chapter 4. The few-shot scooping problem is motivated by autonomous lander missions on extraterrestrial bodies, and I evaluate the method on a variety of challenging novel terrains with large domain shifts and show that employed in a Bayesian Optimization sequential decision-making framework, the method allows the robot to adapt quickly to out-of-distribution terrains, surpassing other state-of-the-art meta-learning methods and non-adaptive scooping strategies in the literature. This application is originally published in [51]. In the second application, I demonstrate that CoDeGa enhances the point force response model learning, improving generalization for online learning on novel objects compared to learning from scratch. When used in the analytical simulator, CoDeGa also improves contact response simulations for novel rigid objects. This shows that CoDeGa is general to other robotics tasks and complements gray-box learning.

In addition to the core theme of improving data efficiency of learning in deformable environments, Chapters 7 and 8 show other relevant topics on contact-rich tasks for robots. Chapter 7 presents a multi-modal model-based reinforcement learning approach to manipulation of fragmented rocks, where the design of the action space leverages the domain knowledge of excavation. I evaluate this method by running extensive physical experiments and show that our method is able to significantly outperform manually designed strategies. This chapter is originally published as [52]. Chapter 8 takes a different angle at solving the challenges of completing contact-rich tasks in the real world. The idea is to leverage human intelligence through teleoperation of a robot avatar. This chapter discusses my effort on designing an avatar system as immersive as possible to use for human teleoperators. The

avatar system achieved 4-th place in the \$10 million ANA XPRIZE Avatar Challenge and completed all tasks in the finals, which evaluated avatar systems on whether they could be controlled by novice operators to complete various manipulation, navigation, and social interaction tasks. The contents of this chapter are currently under review for the International Journal of Social Robots.

1.4 SUMMARY OF CONTRIBUTION

The contributions of this thesis are summarized as follows:

- First simulator that simulates robot locomotion on granular media in real-time while considering realistic granular contacts (Chapter 2), and first trajectory optimization method that simulates long-horizon robot gaits over granular terrains (Chapter 3).
- A data-driven simulator that simulates plausible interaction between a robot and heterogeneous deformable objects using data acquired from only a few touches (Chapter 4).
- First system for automated heart and lung auscultation (Chapter 5).
- A novel meta-training algorithm for training deep kernels that allows a robot to use vision and very little online experience to achieve high-quality scooping actions on out-of-distribution granular terrains (Chapter 6).
- A multi-modal model-based reinforcement learning approach to the excavation of fragmented rocks (Chapter 7).
- An immersive robot avatar with commodity telepresence hardware that achieved 4th place at the recent \$10 million ANA Avatar XPRIZE competition, completing all tasks in the finals (Chapter 8).

CHAPTER 2: GRAY-BOX LEARNING FOR FAST SIMULATION OF ROBOT LOCOMOTION ON GRANULAR MEDIA

In this chapter, we propose a gray-box learning approach for simulating robot locomotion on granular media. We first develop a novel contact model based on the stick-slip behavior between rigid objects and granular grains, which is then learned from a moderate amount of data from experiments. The contact model represents all possible contact wrenches that the granular substrate can provide as a convex volume, called a *feasible wrench space*. The simulation uses an optimization-based contact force solver based on the Maximum Dissipation Principle that combines rigid body dynamics with the *feasible wrench space*, formulated as constraints. We show that our method is able to simulate plausible interaction responses with several granular media at interactive rates, orders of magnitudes faster than state-of-the-art analytical methods. ¹

2.1 INTRODUCTION

Physics simulators are widely used in robotics. They may be used to design and evaluate mechanisms and behaviors, and can be used inside motion planners and grasp planners to predict the outcomes of actions. Rigid body simulators model robots and objects as collections of articulated rigid bodies with either hard or spring-like point contacts with coulomb friction, and are quite commonly used [53]. Finite Element Method (FEM) simulations are used to predict the behavior of deformable materials[54, 55], and Discrete Element Method (DEM) simulations are used to predict the behavior of granular media such as sand and mud. Although FEM and DEM simulators can capture a greater variety of phenomena than rigid body simulations, their computational expense is prohibitive for most uses in robotics. For example, one study reports using 3.2 hours of computation on a 20-core processor to perform 1 s of DEM simulation for a vehicle moving on granular media composed of 150,000 bodies [5]. Hence, we ask the question, can empirical models be incorporated into rigid body simulation to efficiently simulate interaction with more complex deformable materials?

We focus on the problem of simulating robot locomotion on granular media. The interaction between a rigid foot and a granular medium is quite complex and cannot be captured accurately using analytic models. Our novel contribution is a gray-box learning approach, where a substrate contact model is first learned via experiments, and then is used in a contact force solver for rigid-body simulation.

¹This chapter is reproduced from Yifan Zhu, Laith Abdulmajeid, and Kris Hauser, "A Data-driven Approach for Fast Simulation of Robot Locomotion on Granular Media". In International Conference on Robotics and Automation (ICRA) 2019.

We take advantage of the stick-slip behavior of a rigid object, in our case a robot foot, moving on a granular substrate and model this interaction as a single contact resisted by a frictional wrench inside a (non-Coulomb) frictional wrench space. The wrench space varies with the foot’s depth and orientation within the substrate.

During data acquisition, we use a vector of parameters (referred to as the *configuration*) to define the relative pose between the foot and substrate. For a single configuration, we record a set of frictional wrenches exerted on the foot moving at different directions. The database is constructed by repeating this for a set of discrete configurations. Data can be gathered from a DEM simulation or a physical testing apparatus. DEM simulation has the advantage of easy setup and can scale using high performance computing resources, but it can be difficult to tune parameters to fit real world materials. We also constructed a physical testing apparatus in this work.

During the simulation stage, both foot and substrate are modeled as rigid objects and are allowed to interpenetrate. The ground reaction wrench is constrained to lie within the predicted wrench space at the current foot-terrain configuration, and contact forces are calculated using a Maximum Dissipation Principle (MDP) optimization. Due to convexity assumptions, which we find to be satisfied well in practice, the optimization problem can be solved in time polynomial in the number of robot links. As a result, we are able to simulate robot locomotion on granular media in interactive time. Simulation results are demonstrated on both a single foot and a model of the Robosimian quadruped robot traversing flat and sloped terrains composed of granular materials.

2.2 RELATED WORK

Several works have studied legged motion on granular media. Some authors employ simple spring-damper [56, 57] or viscoplastic [58] models for terrain deformations. Despite the convenience of these models, they can deviate far from the empirical behavior of granular media. Xiong et al record the stress that a plate experiences when penetrating a substrate and integrates the stress to determine the stability of a robotic leg with a plate-shaped foot while standing [59]. Using this as a stability region criterion, a controller is developed to produce stable walking gaits for a planar biped traversing flat granular terrain. This model is limited to stability determination and can not predict force response while the object is moving. Li et al develop a method based on resistive force theory to predict the resistance on a thin-plate-shaped leg moving through granular media [60]. It theorizes that the net resistive force acting on an object is the linear superposition of resistive forces on infinitesimal leg elements. They predict net force on a leg by integrating empirically-determined stress

on leg elements and achieve about 30% error. This algorithm treats the penetration as a kinematic event and can not predict static stability. In contrast, our method unifies static and kinematic events and simulates dynamic motions of rigid body interacting with deformable terrains.

A related community studies off-road vehicle locomotion. Various empirical and physics-based models for tire-terrain interaction have been proposed. A comprehensive survey [61] reviews the main efforts in this area. Empirical methods build upon empirical tire-terrain interaction equations. Meanwhile, physics-based solutions rely on DEM and FEM simulations, which are computationally heavy [5]. Because the geometry and elastic properties of tires differ greatly from that of robot feet, the methods developed for off-road vehicles are not suitable for legged robot locomotion.

Data-driven simulation approaches have been adopted in many contexts. Bauza et al use a variation of Gaussian Process (GP) to learn the outcome of planar pushing (treated as a kinematic event) and its variability [62]. The method is able to outperform analytic models after learning only 100 pushing samples. Although it is also possible to use a pure data-driven approach in our case, we employ a gray-box approach to reduce the amount of training data. Another work proposes a data-driven method to predict the post-collision velocity of a planar object [63]. This method learns the optimal parameters of an analytic model given pre-impact states and outperforms pure analytic model. Our work is similar to this method in that we are also learning parameters for a contact model. However, instead of optimizing the parameters by minimizing the discrepancy between simulation and reality, we are extracting parameters directly from experiments. Wang et al use a data-driven piece-wise linear model to simulate cloth, which often has complicated nonlinear, anisotropic elastic behavior due to woven pattern and fiber properties [64]. Local stress-strain parameters are obtained by linearly interpolating the local strain and angle relative to woven pattern from a database obtained through experiments. Bickel et al use a similar approach to learn the local stress-strain relationship through Radial Basis Function (RBF) interpolation on a database of real experiment samples and use these parameters for FEM simulation [65]. Both papers are able to capture the non-linear and heterogeneous deformation behaviors of the materials studied. Like these two works, we are also learning model parameters that capture dominant physical properties and performing dynamics simulations. But in our case, rather than modeling the material stress-strain response, we are concerned with approximating the macroscopic response of granular media on a penetrating rigid object.

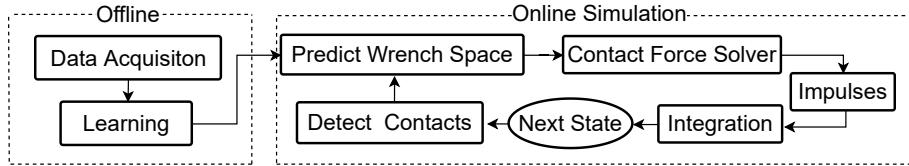


Figure 2.1: The structure of our proposed method for simulating robot locomotion on granular terrain.

2.3 METHODS

The structure of our method is summarized in Fig. 2.1. During offline learning, we run experiments on a granular medium to gather empirical force/torque for our contact model at different configurations. Then we learn a model to predict the wrench space for a new configurations. For each time step during online simulation, we predict the wrench space for each contact between a robot and the granular substrate. Finally, the wrench space constraints are incorporated in an optimization-based contact wrench solver.

The interaction between rigid objects and granular media is complex, and in this work, we only model the most dominant stick-slip behavior observed in low-velocity motion. Other properties like memory effects, high-velocity inertial effects [66], force “overshoot” [59], and force fluctuations [67], are left to future work.

An object moving in granular media causes the grains around it to rearrange and slip against each other. Dry friction between the grains contributes to the majority of the resistance the object experiences. If the external force applied on an object is too weak to cause slippage between the grains, the object sits still. Studies have shown that at low speeds, velocity-independent drag force dominates [68] and this drag increases with larger penetration depth [59]. In addition, this drag is independent of the surface friction between the object and an individual grain [67].

Our proposed contact model limits the possible frictional contact wrenches between a rigid object and a substrate to a given *feasible wrench space*. In the planar case considered here we look at only a 3D wrench space (2D force and 1D torque). The reference frames in this model are shown in Fig. 2.2. We assume a roughly cylindrical shape of the object. Point p_r is a fixed point on the object, and will be the point about which torques are measured. We define it on the central axis of the cylinder and 0.25 m away from the center of the bottom plate p_{center} . Point p_c is the closest point on the surface of the substrate to p_{center} . Frame F_c is centered at p_c and its z -axis points is the outward normal of the surface of the substrate. Frame F_r has the same orientation as F_c but its origin lies on p_r .

The feasible wrench space \mathcal{W} (Fig. 2.3a) is defined as a set of all wrenches $w \in \mathbb{R}^3$ that can

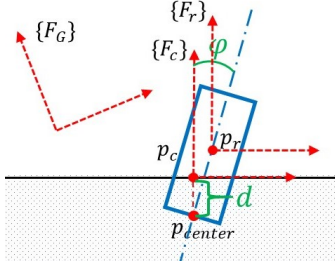


Figure 2.2: Reference frames of a granular contact as described in the text. The configuration of the cylinder relative to the substrate is parametrized by d and ϕ . F_G is the global frame.

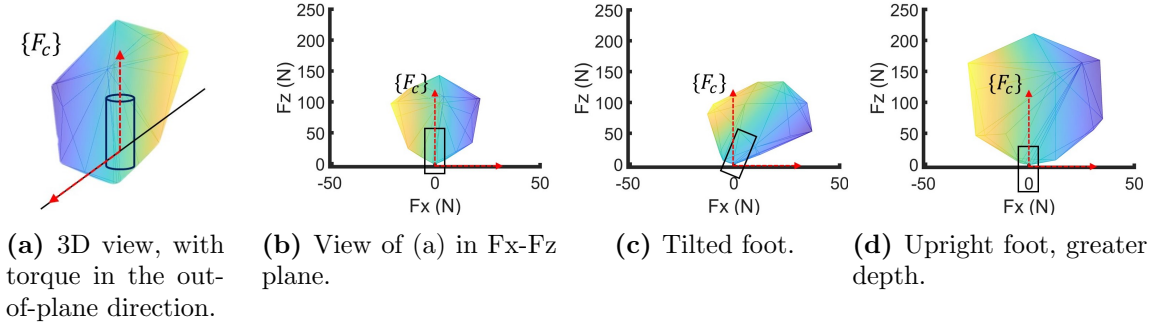


Figure 2.3: Visualizations of the resistance wrenches faced by a rigid cylinder on pebbles, where $\{F_C\}$ is the contact frame.

be exerted by the substrate on the object, with forces and torques measured with respect to F_r . When the force and torque needed to achieve equilibrium is within \mathcal{W} , the object remains still. When the object starts moving/slipping, the friction resisting the object lies on the boundary $\partial\mathcal{W}$. This volume also changes depending on the depth d and orientation ϕ of the object relative to the surface frame F_c . We capture dependent parameters of one object-substrate contact in a vector $\theta = (d, \phi)$. Our method uses machine learning to predict the feasible wrench space $\mathcal{W}(\theta)$, including its depth- and orientation-dependence.

It may be possible to generalize θ to include other dependent parameters, such as 3D orientation, the local geometry of the substrate, the geometry of the foot, and granular medium type. However, when more parameters are used, the amount of data needed to model \mathcal{W} accurately grows rapidly. We focused on d and ϕ to strike a balance between data acquisition difficulty and generalization.

2.3.1 Database and Learning

Data Formulation To model $\mathcal{W}(\theta)$ we capture points on its limit surface; this exploits the property that if $w \in \mathcal{W}$, then $cw \in \mathcal{W}$ for all $c \in [0, 1]$ [69]. Our approach moves the foot through a granular substrate in various directions and records the resistance wrenches.

By the stick-slip assumption, this populates a set of points on $\partial\mathcal{W}(\theta)$. $\mathcal{W}(\theta)$ is then approximated as the convex hull of the measured boundary wrenches. Later, we show this assumption to hold well in practice.

Specifically, we sample n parameter values $\theta_1, \dots, \theta_n$, and for each θ_j we obtain m points $w_{1,j}, \dots, w_{m,j}$ on $\partial\mathcal{W}(\theta_j)$. To obtain $w_{i,j}$, we move the foot through the substrate at the depth and orientation specified by θ_j at a given velocity v_j . Then, $w_{i,j}$ is simply the measured resistance wrench.

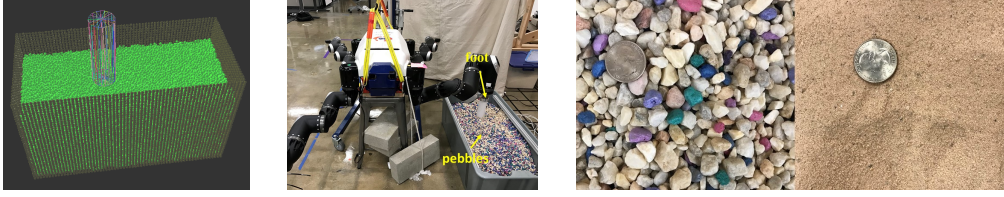
Data Acquisition It may be tempting to gather these measurements simply by moving to a given value of θ , and executing short movements with different velocities. However, this poorly ensures that the medium is in slip phase, due to elastic deformation and force overshoot [59]. Moreover, the medium is disturbed and compacted when the foot moves through it, which runs the risk of inconsistent measurement due to irregular loading. To address these issues, our acquisition method executes longer controlled trajectories and resets the material in-between runs.

For one run with parameter θ_{des} and velocity v_{des} , both defined with respect to F_c , we prepare the substrate in a consistent manner and follow a constant-velocity trajectory that passes through θ_{des} . We record the resistance along the trajectory, filter it, and then obtain a wrench estimate at the instant when the foot is at θ_{des} .

Specifically, we choose the duration of the whole trajectory to be t_{total} and pass θ_{des} at $0.8 \cdot t_{total}$, which is chosen for consistency of repeated measurements. The process follows these steps, as shown in Fig. 2.5:

1. Prepare the surface such that it is flat and not compacted.
2. Move the foot from above the granular surface directly to the starting point of the trajectory. This point is $\theta_{des} - 0.8t_{total}v_{des}$.
3. Start moving at velocity v_{des} for duration t_{total} , while recording force, torque and position.
4. Filter the data with a low-pass filter to remove noise. Calculate exactly when the configuration reaches θ_{des} and record the wrench at this time.
5. Transform the wrench to frame F_r .

The simulation and real robot setup we use for acquiring data is shown in Fig. ???. Simulation is performed using YADE[70], an open-source DEM software, where a floating cylinder is moved through a box of 58,500 spheres. The robot we use is Robosimian, a quadruped with



(a) Simulation setup in the DEM simulator YADE for data acquisition. (b) Data acquisition apparatus with Ro-bosimian. (c) Pebbles and sand used in physical experiments. A 0.25 USD coin is shown for scale.

Figure 2.4: Data acquisition is performed in simulation or a physical robot.

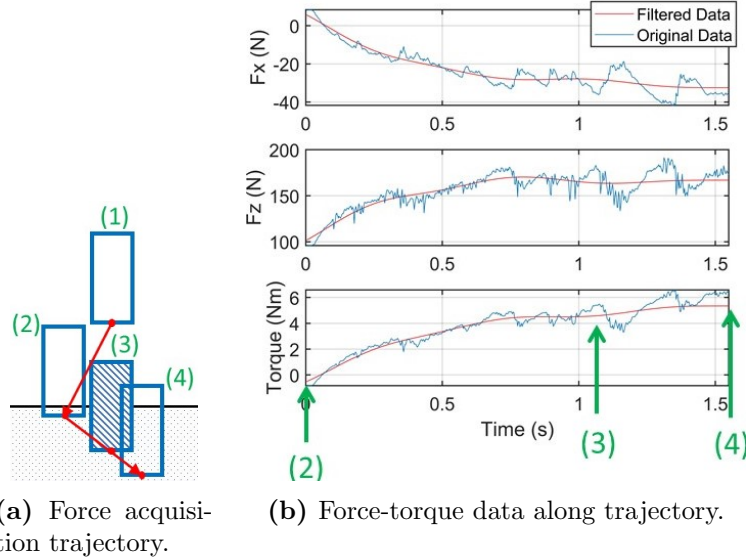


Figure 2.5: The robot foot trajectory for a run with $\theta_{des} = [-0.03, \frac{\pi}{2}]$ and $v_{des} = [0.014, -0.014, 0]$ in sand. The four milestones are: (1) starting pose for the foot, (2) start of the constant velocity trajectory, (3) the robot reaches θ_{des} , (4) end of trajectory.

28 active degrees of freedom and F/T sensors on the ends of its four limbs. One of the feet is driven through a container filled with a granular medium and acquire measurements with the F/T sensor connecting the foot and ankle links.

Learning Our database consists of n parameter values $\theta_1, \dots, \theta_n$, and for each θ_j we have m extreme wrenches $w_{1,j}, \dots, w_{m,j}$ on $\partial\mathcal{W}(\theta_j)$. To model $\partial\mathcal{W}(\theta)$, we learn m regression models that predict the m extreme wrenches for a novel value of θ . Specifically, each wrench regression $w_i(\theta)$ is learned so that $w_i(\theta_j) \approx w_{i,j}$ for $j = 1, \dots, n$.

Experiments find that Radial Basis Function (RBF) interpolation, Gaussian Process (GP), and SVM regression achieve similar performance. Results using 5-fold cross validation show that RBF interpolation outperforms GP and SVM slightly on our dataset, so we use it in our experiments.

In order to approximate $\mathcal{W}(\theta)$, we then take the convex hull of $w_1(\theta), \dots, w_m(\theta)$ and represent the feasible set using the minimal half-space representation:

$$A(\theta)w \leq b(\theta). \quad (2.1)$$

2.3.2 Simulation Framework

Our simulation framework is built upon the work of [71], which solves for contact forces based on the Maximum Dissipation Principle (MDP). Specifically, at each time step the simulation calculates the friction forces maximize the rate of energy dissipation, i.e., minimizing kinetic energy at the next simulation frame. Compared to more traditional linear complementary problem (LCP) methods [72, 73], this method has comparable accuracy and is more convenient for extending simulations beyond Coulomb friction models.

When applying MDP to our case, we note that the movement of particles in the granular substrate also contributes KE to the system. While we cannot know the microscopic movements of all the particles, we assume that the aggregate velocity of the particles is a multiple of the velocity at the bottom of the foot. Our approach then minimizes the sum of the KE of simulated bodies and the approximated KE of granular particles.

We use generalized coordinate representation, and define the following notation:

- h : time step size
- q, v : generalized coordinates and velocities
- u : joint torque control inputs
- k : generalized external forces
- M : generalized inertia matrix
- J_u : joint control torque jacobian
- c_t : stacked vector of contact wrench impulses
- c_j : stacked vector of joint constraint impulses
- J_t : contact wrench Jacobian
- J_c : joint constraint Jacobian
- J_p : p_{center} velocity Jacobian

- M_g : mass matrix of the granular particles

During each simulation step, a contact detector examines whether foot i penetrates the granular medium, and if so, calculates the contact parameters θ_i . We clamp ϕ_i to be between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$ because more extreme angles would be quite far from the sampled data. The bodies are considered to be in contact when d_i is negative. The foot may still touch the substrate if $d_i > 0$ and $\phi_i \neq 0$. However, since the resistance is small in this case, we ignore it for simplicity. For wider feet, one might not want to leave this out. For each contact, we predict its corresponding feasible wrench space $\mathcal{W}(\theta_i)$ using the learned model in the foot’s local reference frame, rotate it to the global frame if it is on a slope and convert it to half-space representation.

Then, we calculate the joint and contact wrench impulses by solving the following optimization problem:

$$\begin{aligned}
\min_{c_t, c_j} \quad & v^{t+1^T} M v^{t+1} + (J_p v^{t+1})^T M_g (J_p v^{t+1}) \text{ s.t.} \\
& J_c v^{t+1} = 0 \\
& h A_1(\theta_1) c_{t_1} \leq b_1(\theta_1) \\
& \vdots \\
& h A_N(\theta_N) c_{t_N} \leq b_N(\theta_N).
\end{aligned} \tag{2.2}$$

In this problem, the velocity on the next time step is $v^{t+1} = M^{-1}(J_t^T c_t + J_c^T c_j + J_u^T u + h k) + v^t$. The second term in the objective represents the kinetic energy of the granular substrate. J_p transforms the velocity at center of mass of each foot to that at p_{center} . M_g is used to approximate the aggregate momentum of change of the grains as a function of the velocity of p_{center} . The inclusion of M_g has a very small effect when simulating the entire robot but improves the accuracy of simulating a single robot ankle. The resulting problem is a QP and can be solved in polynomial time. The position is then updated according to $q^{t+1} = q^t + h v^{t+1}$.

2.4 EXPERIMENTS AND RESULTS

2.4.1 Data Generation

We test on 1 granular medium in DEM simulation and 2 physical media. The DEM simulation medium (Beads) uses particles with the following properties:

- Young’s modulus: $3 * 10^8$

Medium	Depth $d(m)$	Tilt $\phi(rad)$
Beads	[0,-0.06,-0.12,-0.18,-0.24]	[0,0.25,0.5,0.75,1,1.25,1.5]
Pebbles	[0,-0.015,-0.03,-0.045]	[0,0.3,0.6,0.9]
Sand	[0,-0.01,-0.03,-0.05,-0.07]	[0,0.3,0.6,0.9]

Table 2.1: Discretized configurations for three media.

- Poisson’s ratio: 0.3
- Density: 1631 kg/m³
- Coefficient of friction: 0.577
- Shape: spheres with radius of 0.01 m.

The physical granular media are pond pebbles with diameter of about 0.8-1.0 cm (Pebbles) and fine play sand (Sand), shown in Fig. 2.4c.

We sample n configurations on a grid of d and ϕ . The ranges respect the size of the foot and the torque limits of the robot, such that the foot does not get submerged completely into the medium or exceed the stall force of the robot. The resolution of the grid is chosen according to practical time and computational resource limitations. Table I summarizes the the grid parameters. At zero depth, we do not run any experiments and set all wrenches to zero. Also, to achieve better data efficiency, we exploit symmetry in the foot shape, so that the wrench space of (d_i, ϕ_i) has the same shape as that of $(d_i, -\phi_i)$, but with the signs of force in the x direction and torque flipped.

For each configuration, we sample a uniform grid of m velocities on the surface on a 3D unit sphere using spherical coordinates. The unit velocities are then scaled in the x, z, and ϕ axes by (0.2m/s, 0.2m/s 0.6 rad/s) for Beads and (0.02m/s, 0.02m/s 0.3 rad/s) for Pebbles and Sand. The discrepancy comes from the fact that the latter two are much stiffer and the robot foot cannot penetrate into them too much. For DEM simulation, we collect data for $m=26$ and $m=58$. Running our simulators using these two different densities of sampling, we do not observe a significant difference in the object’s post-contact motion. For real world experiments, which are more time consuming, we collect data for $m=26$ only.

2.4.2 Validity of Convexity Assumption

We evaluate the extent to which the wrench space convexity assumption holds. As a concavity metric, we use the relative distance between the convex hull surface and innermost point in the convex hull. Specifically, for a given θ_j this dimensionless measure is $\text{Concavity}_j =$

Medium	Average Concavity	Worst Concavity
Beads	0.0174	0.1207
Pebbles	0.0013	0.0155
Sand	0	0

Table 2.2: Concavity of empirical wrench spaces

$\max_i((b(\theta_j) - A(\theta_j)w_{i,j})/\|w_{i,j}\|)$. To avoid points with small magnitude, we exclude data where depth is zero, or the magnitude of the point is less than 1. If all of the inner points are less than 1, we treat the concavity as zero. Tab. 2.2 lists for each medium the average concavity across configurations as well as the worst case. Most of the wrench spaces are convex. There are only a few wrench spaces with concavity larger than 10%, and these occur at wrenches with small magnitudes.

2.4.3 Simulation Experiments

Accuracy Test To evaluate the accuracy of our method, we command a foot to follow a trajectory and compare the resistance on the robot acquired from experiments and our simulator. Our simulator uses a simple PID controller to follow the reference trajectory. The mass and the moment of inertia of the simulated foot are 7.357 kg and 0.061 kg·m². The PID gains being used are $K_p = [8000, 20000, 300]$, $K_i = [4000, 6000, 1000]$ and $K_d = [200, 400, 10]$. M_g is tuned empirically by matching simulated and measured trajectories and wrenches, and we use $M_g = \text{diag}(14.714, 14.714, 0.122)$, which is twice the inertia of the foot.

Fig. 2.6b shows a good match between the simulated resistance and measured resistance on a straight trajectory in Sand, showing resistance force and torque increasing with depth. This is similar for Pebbles medium and is not shown here. Results for a curved path in Pebbles are shown in Fig. 2.6d. The simulator is able to predict the general trend and approximate magnitudes of the resistance. However, the simulated resistance stays high as the downward movement slows, while the actual resistance drops. We believe that the discrepancy is caused by un-modelled memory effects, such as compaction, and some integral wind-up from the simulated PID controller. Modeling memory effects would be an interesting problem for future work.

Drop Test with a Single Foot Next, we perform a test of dropping behavior with one single object. We drop the object from a small height at two different surface orientations, displayed in Fig. 2.7. When the object is dropped in vertical orientation of flat terrain, it remains erect. In slightly sloped terrain, the object begins to tip, but ultimately is able to

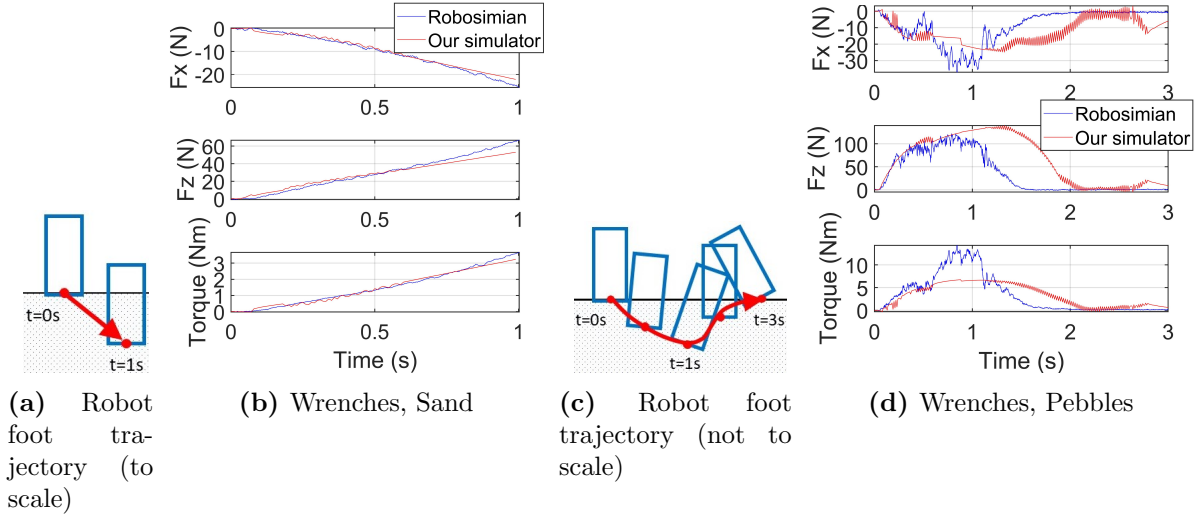


Figure 2.6: Comparing measured wrenches between physical Robosimian and our simulator using a straight path on Sand and a curved path on Pebbles.

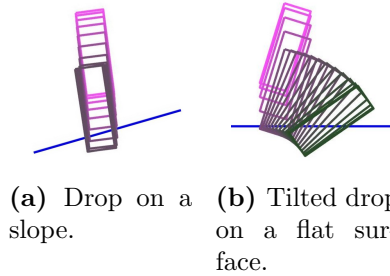


Figure 2.7: Drop tests.

balance because the medium is able to provide a resistance torque. When slightly tilted at 0.2 radian, the object tips over and eventually stops on a flat surface.

2D Quadruped Locomotion Test Our next tests perform a highly simplified model of the sagittal movement of a quadruped walking on soft terrain. Fig. 2.8 shows images from the resulting animations. Note that the robot faces to the right of the page and the simulated terrain is sand, if not specified. In the first simulation, the robot has its weight shifted to the rear legs and pushes the front feet forward. Because the rear feet are able to resist a larger force, only the front feet slip. The second simulation is one trotting gait, where the diagonal two limbs move simultaneously. The feet sink and lift realistically as the weight of the body shifts. In the third simulation, the robot is dropped onto a sloped terrain, with its joints locked. The robot initially slides and accelerates, but eventually stops when the terrain flattens. Pebbles allow shallower penetration and stops the robot at a higher potential energy, compared to sand. This is expected because pebbles are able to provide greater resistance force at the same penetration depth.

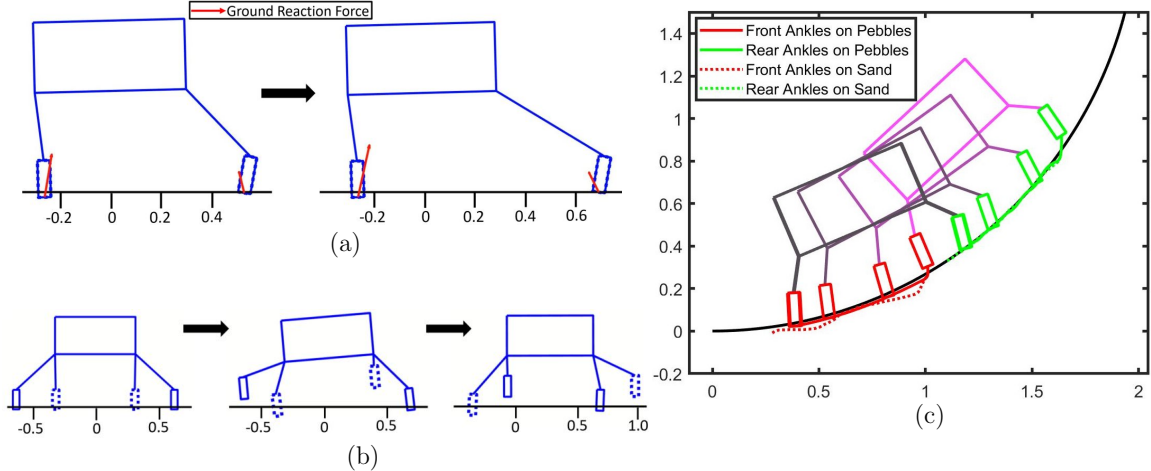


Figure 2.8: Locomotion tests with different terrains on a simplified 2D Robosimian model facing right. (a) Pushing the front feet forward. Because the rear legs are sunk deeper in the sand, they resist motion more, so the front feet slip. (b) A trot gait showing deeper penetration during the two-foot phase. (c) Sliding on a slope and eventually stopping, with joints fixed. The red and green lines show the trace of the front and rear feet, respectively. The snapshots of the robot are from the simulation on pebbles.

Computation Speed In this test, we use a more complicated quadruped model that includes 1 body and 4 limbs, with each limb consisting of 2 links and 1 foot. In total, a system of 13 rigid bodies and 12 revolute joints are simulated in Matlab on a standard PC with an Intel i7 2.4GHz processor. The resultant optimization problem has 4 contacts, 36 variables, 24 constraints associated with joints, and 4 sets of constraints from granular contacts. One frame on average takes about 0.343s to compute, with about 90% of the time spent on solving the optimization, and about 5% on interpolating the wrench spaces. By utilizing a compiled language and a more task-specific convex optimization package, we expect to speed up computation significantly.

2.5 CONCLUSION

This chapter proposed a contact model for rigid objects and granular media to capture depth-dependent stick-slip phenomena. We present a procedure to acquire data-driven feasible wrench spaces and learn the parameters for this contact model. An optimization-based contact force solver is developed to use this data to simulate continuous granular contact with multiple contacts at interactive rates. Future work should consider improving the prediction accuracy by incorporating memory effects of granular material like compaction and pile formation. It is also important to examine improving the data efficiency as more parameters are included in the learning model, which makes generalization more challenging.

CHAPTER 3: CONTACT-IMPLICIT TRAJECTORY OPTIMIZATION WITH LEARNED DEFORMABLE CONTACTS USING BILEVEL OPTIMIZATION

Using the simulator based on gray-box learning presented in Chapter 2, this chapter shows a bilevel, contact-implicit trajectory optimization (TO) formulation that searches for robot trajectories on granular terrains. On the lower-level, contact forces are solved via a quadratic program (QP) with the maximum dissipation principle (MDP), based on which the dynamics constraints are formulated in the upper-level TO problem that uses direct transcription. Our method uses a contact model for granular media that is learned from physical experiments, but is general to any contact model that is stick-slip, convex, and smooth. We employ a primal interior-point method with a pre-specified duality gap to solve the lower-level problem, which provides robust gradient information to the upper-level problem. We evaluate our method by optimizing locomotion trajectories of a quadruped robot on various granular terrains offline, and show that we can obtain long-horizon walking gaits of high qualities.²

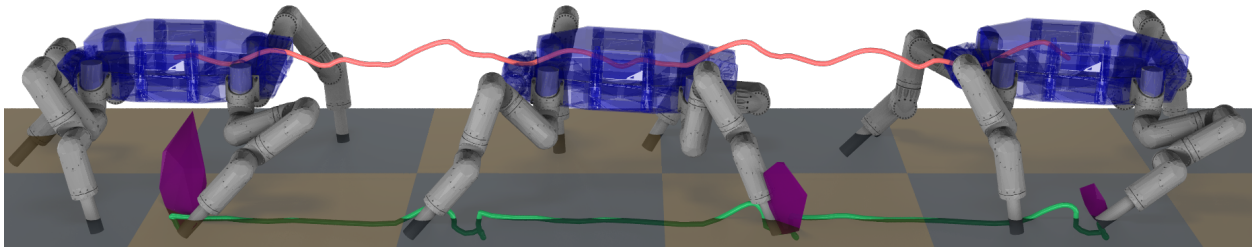


Figure 3.1: An optimized trajectory on a flat terrain with sand (tiles are 1m in length). The centroid of the robot torso over the 9s trajectory is traced in red, and the front right foot position in green. The contact wrench space of the front right foot is in purple. [Best viewed in color.]

3.1 INTRODUCTION

Legged robot locomotion planners typically assume rigid contacts with Coulomb friction, but non-rigid terrains with different frictional characteristics are pervasive in the world, e.g. sand, mud, carpet, etc. To tackle these challenging soft terrains, existing techniques rely on feedback control while assuming rigid contacts [74], or modeling the terrain as a mass-spring damper [56, 75] or as viscoplastic [58] in the controller. While inaccurate contact models

²This chapter is reproduced from Yifan Zhu, Zherong Pan, and Kris Hauser, "Contact-Implicit Trajectory Optimization With Learned Deformable Contacts Using Bilevel Optimization". In International Conference on Robotics and Automation (ICRA) 2021.

result in sub-optimal motions, our proposed trajectory optimization (TO) framework adopts realistic soft contact models learned from physical experiments on granular terrains.

This chapter is motivated by our prior work [48] that built a semi-empirical simulator for robots traversing granular terrains. Based on the stick-slip behavior of a rigid body contacting granular media, a granular contact is first modeled as all possible contact wrenches between a robot ankle and the granular terrain, which is a convex volume learned from data. Then during online simulation, contact forces are solved by a quadratic program (QP) with linear constraints during each simulation frame, based on the maximum dissipation principle (MDP) [71].

In this work, we incorporate this simulator into motion planning with granular contacts, where we take advantage of the accurate learned contact models to obtain high-quality trajectories. In particular, we employ TO, which is a powerful framework for generating locally optimal trajectories for robotic systems (for details on TO, refer to the survey by Betts [76]). Compared to sampling-based motion planning methods [77], which can generate globally optimal trajectories, TO methods scale better to high-dimensional robotic systems. Our contribution in this work is two-fold. First, we propose a bilevel TO framework that is capable of automatically searching for locomotion trajectories of legged robots traversing granular terrains. Second, we build a differentiable rigid body simulator with learned granular contact models. Each simulation frame consists of solving a QP, whose gradients are robustly obtained by employing a primal interior-point method with a pre-specified duality gap.

On the upper-level of the TO framework, we adopt a direct transcription formulation that is also contact-implicit. The dynamics constraints of the TO are obtained through the lower-level optimization, which is our differentiable physics simulator. We are then able to supply analytic gradients of the dynamics constraints to the optimization solver for efficiency and accuracy.

In this work, we focus on locomotion planning on granular terrains for a Robosimian quadruped developed by JPL, although it could be applied to general contact-rich motion planning, with a contact model that is convex and twice-differentiable. This turns out to be a very challenging problem, ill-posed and with potentially many local minima. However, we show that our method can automatically generate locomotion trajectories of long horizons offline on a variety of terrain shapes, where competitive methods such as reinforcement learning and standard trajectory optimization fail to generate trajectories on a simple flat granular terrain.

3.2 RELATED WORK

While TO has been applied successfully to smooth dynamical systems, planning contact-rich trajectories, e.g. locomotion planning for legged robots, is still a challenging problem due to the underlying non-smooth, hybrid dynamical system. Successes have been achieved by adopting a pre-specified contact sequence [78, 79], however, manually designing contact sequences requires significant engineering and could be daunting for complex systems.

Contact-implicit TO: Recently, contact-implicit TO methods [80, 81, 82, 83, 84] are proposed to solve this problem by folding the contact dynamics into the formulation of the mathematical program, which avoids the need of a contact sequence input. Our method falls into this category as well, and we review the closely related works here. Posa et al. [80] propose a direct transcription method that searches locally optimal trajectories for robots with inelastic impacts and rigid contacts with Coulomb friction. Based on the formulation of multi-contact dynamics as a Linear Complementarity Problem (LCP) [85] for dynamics simulation, the proposed algorithm introduces contact forces as additional optimization variables and incorporates complementarity constraints (CC) in the TO formulation. The resulting algorithm solves a mathematical program with CC and is demonstrated to generate locally optimized trajectories for high-dimensional systems. One drawback of this method is the high non-linearity introduced by the CC, which leads to poor convergence performance. To avoid having CC, Carius et al. [83] propose a bilevel TO framework, where the lower-level solves for the robot dynamics with rigid contacts using a time-stepping scheme of rigid body dynamics forward simulation, whose gradients can be obtained analytically. The upper-level TO uses iLQR [86], a shooting type method, and is shown to be able to compute a trajectory with 300 time steps for a 3 degree-of-freedom (DoF) robot in 2.18 ms. The idea of our method is closely related to this work. Instead of folding forward simulation with rigid contacts into system dynamics constraints, we use granular contacts. In addition, we use a direct TO method which is numerically advantageous for long trajectories. Another method that aims to remove CC is proposed by Landry et al. [82]. They also put forward a bilevel TO framework, where the upper-level is a direct transcription method and the lower-level solves a QP to obtain the tangential contact forces at the rigid contacts. Our method is most closely related to this work. While Landry et al. [82] only include part of the system dynamics in the lower-level optimization (only tangential contact forces), our method solves all the system dynamics implicitly in the lower-level. Our method considers a general contact model and we also adopt a different strategy to choose a gradient for the lower-level problem when it is not uniquely defined. Landry et al. [82] use a first-order method to solve the QP where the gradient of the solution could be non-unique and they resort to choosing a subgradient

with least-squares, which does not have a strong theoretical foundation. However, we use an interior point method with a pre-specified duality gap where the gradient is always uniquely defined.

Bilevel Optimization: Our method falls into the general category of bilevel optimization [87], which is an optimization problem (upper-level) whose constraints involve other optimization problems (lower-level). In addition to the two works that are mentioned above, bilevel optimization has been applied to TO in a variety of works [88, 89]. Farshidian et al. [88] propose a bilevel optimization framework to plan motions for legged robots, where the upper-level optimizes time allocation for a fixed contact sequence, and the lower-level computes optimal continuous control inputs. Tang et al. [89] propose a method to generate time-optimal trajectories for smooth dynamical systems, which optimizes the trajectory shape on the upper-level and the time allocation on the lower-level. While these two methods decompose a challenging optimization via bilevel optimization, our method encodes system dynamics naturally into lower-level optimization.

Locomotion on Soft Terrain: Instead of planning motions on soft terrains, previous methods design controllers compensating for soft contacts. Kang et al. [56] model a soft terrain with compressive springs and stabilize the CoM motion of a humanoid by tilting its ankle according to the attitude angle of the robot. Vasilopoulos et al. [58] calculate controls of a 1-DoF leg to maintain a desired height while jumping on a soft terrain based on a viscoplastic model. Fahmi et al. [75] model a soft terrain as a compressive spring, run an online controller to estimate the compliance, and incorporate the compliance explicitly into a whole-body-control framework for a quadruped. Compared to building controllers, our method plans trajectories with long horizons, allowing for more diverse and optimal motion plans.

3.3 METHOD

3.3.1 Problem statement

We search for a discretized trajectory for a legged robot, represented by the states $\mathbf{x}[\cdot] \in \mathbb{R}^{n_x}$ and controls $\mathbf{u}[\cdot] \in \mathbb{R}^{n_u}$, where n_x and n_u denote the dimensions of state and control. Let a robot state be the concatenation of robot configuration and velocity, i.e. $x = [q, \dot{q}]$ and $\mathbf{x}[k] = [q_k, \dot{q}_k]$, the dynamics of the robot is represented by the Newton-Euler equation:

$$M(q)\ddot{q} + H(q, \dot{q}) = u + J_c^T f, \quad (3.1)$$

where u is the joint torque, M is the generalized mass matrix, H is the gravitational, centrifugal and Coriolis forces, J_c is the contact jacobian, and f is contact wrench. Using an Euler integration scheme, the robot dynamics along the discretized trajectory $\mathbf{x}[\cdot]$ and $\mathbf{u}[\cdot]$ is:

$$\begin{aligned}\dot{q}_{k+1} &= M(q_k)^{-1}(u_k + J_c^T f - H(q_k, \dot{q}_k)) \cdot dt + \dot{q}_k, \forall k \\ q_{k+1} &= \dot{q}_k \cdot dt + q_k, \forall k,\end{aligned}\tag{3.2}$$

where dt is the time step. Formulating this equation requires solving the contact wrench f , which is computed by a differentiable semi-empirical simulator for robots traversing granular terrains. Details of the simulator are described in the next section.

3.3.2 Robot dynamics with granular contacts

Contact model This contact model is first discussed in Chapter 2. Inspired by the stick-slip behavior of granular media, we consider contact models that limit the possible frictional contact wrenches between a rigid robot foot and a granular substrate with a convex wrench space, which can be learned from data obtained in physical experiments [48]. If the external wrench can be resisted by a wrench in the wrench space, the contact sticks, and otherwise it slips. The shape of the wrench space changes as the penetration depth and orientation of the rigid body change, shown in Fig. 3.1, and here we have a learned twice-differentiable contact model w.r.t. the robot configuration:

$$f \in \mathcal{W} \iff f \in \left\{ \sum_{i=1}^V w_i v_i(q) \mid w_i \geq 0 \wedge \sum_{i=1}^V w_i \leq 1 \right\}.\tag{3.3}$$

Instead of the half-space representation presented in Chapter 2, this is a vertex representation of the convex wrench space, where V is the total number of vertices and $v_i(q)$ is a learned vertex, which is dependent on the robot’s configuration and the terrain shape.

Contact detection We only consider contacts between the robot feet and the granular substrate. At a single contact, each learned contact wrench vertex $v_i(q)$ is a smooth function of penetration depth and orientation, i.e.:

$$v_i(q) = v_i(-z^T p(q), R(q)),\tag{3.4}$$

where $p(q)$ is the bottom center point of the robot foot that is having contact, z is the outward normal at the surface point that is closest to p , and $R(q)$ is the 3×3 local-to-global rotation matrix. In our prior work, we take the assumption that the environment is a flat

terrain paved using granular materials. Here, we propose a heuristic method to generalize this model to an arbitrary terrain shape represented using a watertight mesh. Since the terrain is a watertight mesh, we can compute a singled distance value from $p(q)$ to the surface of the mesh, which is denoted as $d(p(q))$. We can also estimate the generalized outward normal direction as:

$$z(q) = \nabla d(p(q)). \quad (3.5)$$

We further denote $\bar{R}(\bar{z})$ as the minimal rotation matrix that transforms $[0, 0, 1]^T$ to \bar{z} . The generalized vertex of the contact wrench space is then defined as:

$$\begin{aligned} \bar{v}_i(q) &= \bar{R}v_i(d(p(q)), \bar{R}^T R(q)), \\ f \in \mathcal{W} &\iff f \in \left\{ \sum_{i=1}^V w_i \bar{v}_i(q) \mid w_i \geq 0 \wedge \sum_{i=1}^V w_i \leq 1 \right\}. \end{aligned} \quad (3.6)$$

Intuitively, this is a generalized pullback operator from the arbitrary shape terrain to the flat ground. Note that to form a well-defined wrench space that can be used in a bilevel optimization, \bar{v}_i must be a twice-differentiable function of q . However, the generalized penetration depth computed using an exact triangular mesh is generally non-differentiable. To remedy this case, we first compute a signed distance field for the mesh on a uniform grid and then use cubic interpolation to approximate d .

Contact wrench solving The robot dynamics is solved based on the MDP [71], which states that the frictional contact wrenches would maximize the energy dissipation of the system. Therefore, a lower-level optimization problem is given by:

$$\begin{aligned} &\underset{f}{\operatorname{argmin}} \\ &\frac{1}{2} \|M(q_k)^{-1}(u_k + J_c^T f - H(q_k, \dot{q}_k))\|^2_{M(q_k)} \\ &\text{s.t. } f \in \mathcal{W}, \end{aligned} \quad (3.7)$$

where $\|\cdot\|_M^2$ is the quadratic form. Here we slightly abuse the notation and let f and \mathcal{W} denote a concatenation of all the contacts instead of a single contact. The program is a QP which can be solved efficiently in polynomial time, and the gradients of this program w.r.t. all problem parameters can be obtained by sensitivity analysis, whose details we refer readers to [90]. One concern in sensitivity analysis is the loss of linear independence constraint qualification (LICQ), which gives non-unique gradients. The constraints in Eqn. 3.7 are learned from data and we cannot guarantee LICQ, and in practice we do observe the loss

of LICQ in some cases. We handle this by using a primal, interior point method with a pre-specified duality gap to solve the QP, where the solution contact wrenches as a function of problem parameters, i.e. $f(q_k, \dot{q}_k, u_k)$, is always single-valued and twice-differentiable.

If we assume that the QP in Eqn. 3.7 is strictly convex, then function f is single-valued. However, using this single-valued function f in a bilevel optimization is inappropriate because it is known that f is only C^0 -continuous [91], while large-scale constraint optimizers usually require twice-differentiable constraints [92]. To ensure both strict convexity and twice-differentiability, we propose to use a primal, interior-point solver with a finite duality gap. In this way, the QP is transformed into the following unconstrained optimization:

$$\begin{aligned} & \underset{f}{\operatorname{argmin}} \\ & \frac{1}{2} \|M(q_k)^{-1}(u_k + J_c^T f - H(q_k, \dot{q}_k))\|^2 + \dot{q}_k^2_{M(q_k)} \\ & - \mu \cdot dt \sum_{i=1}^V \log(w_i) - \mu \cdot \log(1 - \sum_{i=1}^V w_i), \end{aligned} \tag{3.8}$$

where μ is the finite log-barrier coefficient. Since the log-barrier function always contributes a full-rank Hessian matrix, the above problem is strictly convex and the function f is single-valued. Further, by the implicit function theorem, the function f is twice-differentiable as long as the problem data, i.e. M, H, J, v_i , are sufficiently smooth. In practice, we solve the unconstrained optimization using Newton’s method with the initial guess from last iteration of upper-level optimization.

As a result, for a twice-differentiable learned contact model, the contact wrenches are a smooth function of problem parameters, whose gradient information can be robustly obtained via sensitivity analysis, which we denote as $\nabla f(q_k, \dot{q}_k, u_k)$. In addition, since we already calculate the gradients of all problem data during sensitivity analysis. Specifically, denoting Eqn. 3.2 as $g(q_k, \dot{q}_k, u_k)$, its gradient $\nabla g(q_k, \dot{q}_k, u_k)$ is obtained via the chain-rule.

3.3.3 Bilevel trajectory optimization

Our method uses a direct transcription TO method, where the TO is formulated as a non-linear program (NLP) that locally optimizes a physically correct discrete trajectory. Overall,

our method solves the following NLP:

$$\begin{aligned}
& \underset{\mathbf{x}[\cdot], \mathbf{u}[\cdot], \mathbf{c}[\cdot]}{\operatorname{argmin}} && J(\mathbf{x}[\cdot], \mathbf{u}[\cdot], \mathbf{c}[\cdot]) \\
& \text{s.t.} && \mathbf{x}[0] \in \mathcal{X}_{init}, \mathbf{x}[N] \in \mathcal{X}_{goal}, \\
& && \mathbf{x}[\cdot] \in \mathcal{X}, \mathbf{u}[\cdot] \in \mathcal{U}, \\
& && \text{Eqn. 3.2 holds,} \\
& && h(\mathbf{x}[\cdot], \mathbf{u}[\cdot], \mathbf{c}[\cdot]) = 0,
\end{aligned} \tag{3.9}$$

where $\mathbf{c}[\cdot] \in \mathbb{R}^{n_c}$ represents additional problem-specific variables of dimension n_c . The feasible regions for the initial and final states are specified in \mathcal{X}_{init} and \mathcal{X}_{goal} . The feasible states and controls along the trajectory are defined in \mathcal{X} and \mathcal{U} . N is the total number of grids in the discrete trajectory. J is some performance measure we aim to optimize. Other additional equality constraints are encoded in $h(\cdot)$.

This problem is bilevel because the dynamics constraints Eqn. 3.2 entail solving a set of lower-level optimization problems, i.e. Eqn. 3.7, which were introduced in the previous section. In practice, solving such a challenging TO problem using a NLP solver requires analytical gradients of the objective function and the constraints. In particular, the analytic gradients of the dynamics constraints $\nabla g(q_k, \dot{q}_k, u_k)$, given by our differentiable simulator, are supplied to the NLP solver, along with the gradients of other constraints and the objective function in Eqn. 3.9.

3.3.4 Spline constraints

To improve the smoothness of the optimized locomotion gaits, we present a scheme to use splines to regularize the end effector movement and avoid ‘‘stutter step’’. Intuitively, we expect an optimized robot gait where each robot foot traces a smooth path. In particular, for each robot foot, we represent its position and orientation with a cubic spline whose control points are $\mathbf{c}[\cdot] \in \mathbb{R}^n$, where n is 3 in 2D and 6 in 3D. We use $T_j(k) : [0, N-1] \rightarrow \mathbb{R}^n$, to denote the interpolated position and angle of robot foot j along the cubic spline specified by the control points, where N is the total number of grid points in the discrete robot trajectory. In addition, let $FK_j(x) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^n$ denote the forward kinematics of the robot that gives the position and angle of the j -th robot foot, the spline constraints are formulated as:

$$FK_j(x[k]) = T_j(k), \forall j, k. \tag{3.10}$$

In practice, we might only use partial dimensions of a spline, i.e. we may choose to use only the position for the spline control points, without the angle.

3.3.5 Cost function

We propose a cost function that includes 3 different components, where the third component is optional: $J_{total} = J_{path} + J_{periodic} + J_{spline}$. J_{path} is the quadratic path cost defined as $J_{path} = \sum_k \|\mathbf{x}[k] - x_{ref}\|_Q^2 + \|\mathbf{u}[k]\|_R^2$, where x_{ref} is the reference robot state that is manually selected, and Q, R are diagonal weight matrices. We further regularize the trajectory by introducing a periodic cost that encourages the robot limbs to move periodically: $J_{periodic} = \sum_{k=0}^{N-P} c_p (F \cdot q_k - F \cdot q_{k+P})^2$, where c_p is a weight constant, F is a selection matrix that selects only the limbs of the robot, and P is the desired period.

Instead of having the splines be hard constraints, we find in practice that soft constraints might work better for certain terrains. Therefore, we propose an alternative formulation where we use a spline cost to replace the spline constraints. Letting $h(\cdot)$ denote the constraints in Eqn. 3.10, the spline cost is $J_{spline} = c_s \|h(\cdot)\|^2$, where c_s is a weight constant.

3.3.6 Full formulation

A final consideration is that for contact models learned from data, it is prudent to limit the input, i.e. depth and orientation, to stay close to the range of collected data, because learned models have questionable extrapolation capability. We encode this constraint, along with the robot joint position and velocity limits, in \mathcal{X} . In addition, we have the joint torque limits $u[\cdot] \in \mathcal{U}$.

The complete formulation is the following:

$$\begin{aligned}
 & \underset{\mathbf{x}[\cdot], \mathbf{u}[\cdot], \mathbf{c}[\cdot]}{\operatorname{argmin}} \quad J_{total} \\
 & \text{s.t. } \mathbf{x}[0] \in \mathcal{X}_{init}, \mathbf{x}[N] \in \mathcal{X}_{goal}, \\
 & \quad x[\cdot] \in \mathcal{X}, \mathbf{u}[\cdot] \in \mathcal{U}, \\
 & \quad \text{Eqn. 3.2 holds,} \\
 & \quad \text{Eqn. 3.10 holds. (if not using spline cost)}
 \end{aligned} \tag{3.11}$$

Note that $\mathbf{c}[\cdot]$ represents the cubic spline control points for all limbs.

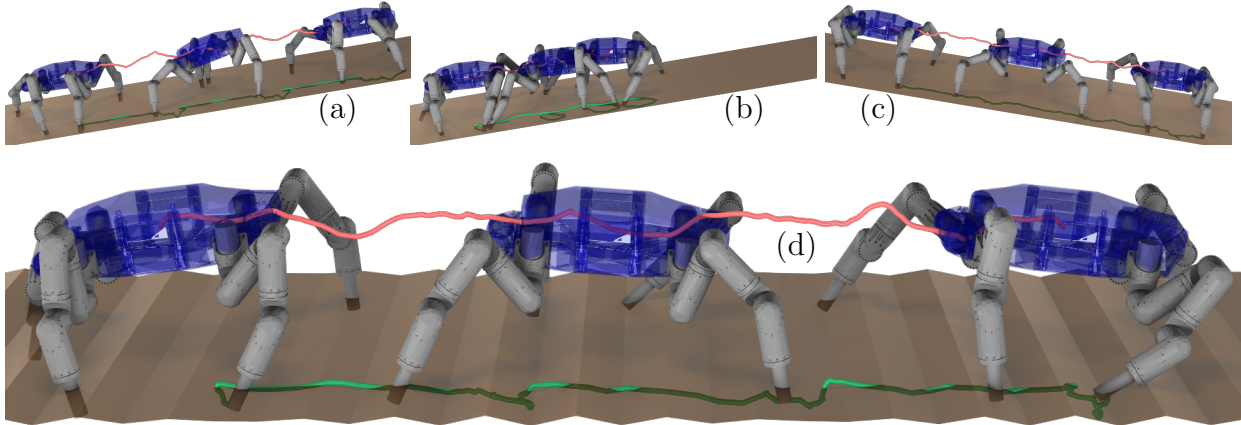


Figure 3.2: (a): optimized gait on the 10° slope. (b): non-optimized gait on the 10° slope. (c): optimized gait on the -10° slope. (d): optimized gait on the wavy terrain. The torso centroid is traced in red, while the front right foot position in green. [Best viewed in color.]

3.4 RESULTS

3.4.1 Robot and contact model

We use a Robosimian quadruped robot model for all the experiments, shown in Fig. 3.1. We consider a 2D model where only sagittal movement of Robosimian traversing soft terrain presents. The robot has a total of 4 limbs and 12 revolute joints, making $n_x = 30$ (position and velocity of the joints and torso) and $n_u = 12$.

The contact model we consider here is the learned granular contact model for sand from our previous work [48]. We use radial basis function (RBF) interpolation as the learning model for data gathered from physical experiments.

3.4.2 Desired gait and performance metric

Our experiment attempts to generate a walking gait that follows a desired torso forward velocity on different terrains. Since the cost of an optimized solution does not directly reflect the quality of a gait, we define 2 metrics to quantitatively evaluate it. The first metric measures how well the desired torso velocity is tracked: $M_v = \sum_{i=0}^N |x_{v,i} - \hat{x}_v| / N$, where $x_{v,i}$ is the torso forward velocity at grid point i , and \hat{x}_v is the desired torso forward velocity. The second metric M_c measures the cost of transport, which is $\sum_{k=0}^N \sum_{i=0}^{n_u} |(F \cdot \dot{q}_k)[i] \cdot \mathbf{u}[k][i]| / (smg)$, where $(F \cdot \dot{q}_k)[i]$ and $\mathbf{u}[k][i]$ are the i -th joint velocity and joint control at grid k , s is the total translation of the torso, and mg is the weight of the robot. Note that for humans, the cost of transport of walking on sand has been reported to be 1.6–2.5 times that of walking on hard ground [93].

3.4.3 Parameter tuning

The optimization problem tackled here is very challenging, and we found in practice that the convergence behavior and trajectory quality are sensitive to parameters. Here is a summary of all the parameters in our formulation:

- Total number of grid points N , time step dt .
- The quadratic state cost matrix Q is a diagonal matrix, whose only nonzero diagonal entries are those for the torso forward velocity, torso orientation, and torso height, denoted by Q_v, Q_o, Q_h .
- The nonzero entries of x_{ref} include the desired forward velocity, torso height, and torso orientation, denoted by $\hat{x}_v, \hat{x}_o, \hat{x}_h$.
- The quadratic control cost matrix R , which is diagonal.
- The period P and the periodic cost weight c_p .
- Spline dimensions, and the number of control points N_c for each limb. Whether to use spline constraints or cost. If using constraint, constraint scaling for spline constraints. If using spline cost, weight c_s .
- μ , the log-barrier coefficient in lower-level optimization.

In our experiments, we manually pick the parameters $N, dt, \bar{x}_v, \bar{x}_o, \bar{x}_h, P, N_c, \mu$, which can be chosen based on the desired properties of the robot gait. For the rest of the parameters, we run a grid search, and pick the set of parameters that generate the gait of highest quality.

3.4.4 Results on a horizontal terrain

In this experiment, the goal is to generate gaits for the Robosimian on a flat horizontal sandy terrain while maintaining a desired forward torso velocity. The parameters (some of which are from grid search) are: $N = 181, dt = 0.05, Q_v = 100.0, Q_o = 0.01, Q_h = 0.01, x_v = 0.4, x_o = 0.0, x_h = 0.75, R = 10^{-5}I, P = 60, c_p = 0.1, \mu = 10e^{-3}$, all in standard units. Instead of using both the positions and orientations for the splines, we use only the orientations as constraints without constraint scaling, and $N_c = 30$.

We use the Knitro NLP solver [92] for solving the TO. For all the experiments, we set the feasibility and optimality tolerance to 10^{-3} . All experiments are conducted on a standard PC with a Xeon 20-core CPU. During experiments, we found that the optimized results are sensitive to the initial guess. We first experimented with a trivial initialization, where the

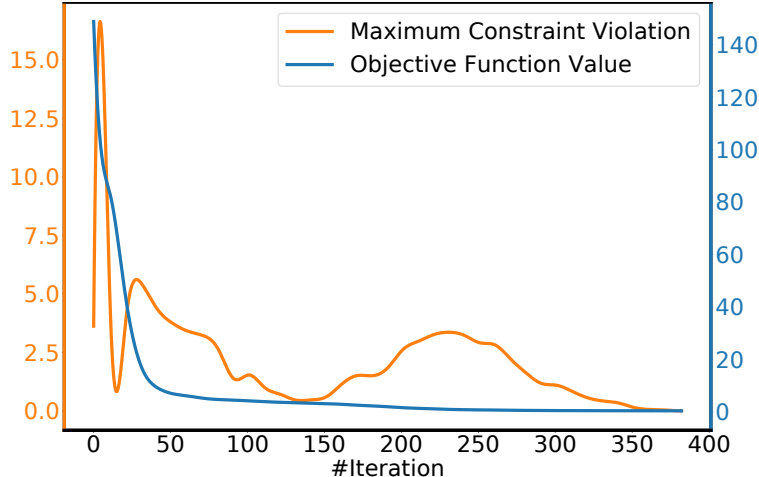


Figure 3.3: The cost and maximum constraint violation over the iterations for optimizing a trajectory on the flat horizon terrain.

Terrain	0°	2.5°	5°	7.5°	10°	12.5°	15°	17.5°	20°	-2.5°	-5°	-7.5°	-10°	-12.5°	-15°	-17.5°	-20°	wavy	10° NO
M_v	0.0010	0.0023	0.013	0.0012	0.013	0.0031	0.036	0.0054	0.29	0.0056	0.0013	0.014	0.0044	0.046	0.040			0.010	0.22
M_c	14.6	12.5	14.8	14.0	17.6	24.5	25.4	27.6	97.3	7.58	12.6	21.5	29.2	48.1	73.2	fail	fail	26.73	35.2
Time (hr)	0.50	2.52	0.25	0.48	0.57	1.03	0.55	1.17	1.1	0.90	0.58	0.36	1.1	3.9	8.2			0.51	NA

Table 3.1: Optimized trajectory performance on different terrains (M_v : velocity error, M_c : cost of transport, NO: Non-optimized)

robot stands still. The optimized trajectory shows the robot sliding and dragging its feet to move forward, without ever lifting them off the terrain. To minimize manual tuning, we end up using a simple trotting in place initialization, where the robot walks in place for 3 steps. In particular, we manually design the trajectory, track the trajectory with a PID controller at 500 Hz, down-sample the states and average the controls to obtain the trajectory with $dt = 0.05$ s.

The cost and constraint violation over the iterations are shown in Fig. 3.3, while the optimized trajectory is shown in Fig. 3.1. The performance metrics are $M_v = 0.0023$ and $M_c = 14.6$, showing that the desired velocity is tracked very well. Visually, the robot has a smooth trajectory with little jerkiness. This particular gait took 0.499 hr to compute.

Baseline comparison Next, we compare with 2 baselines: standard trajectory optimization (iLQR) and reinforcement learning (RL). We use the open-source implementation of the iLQR algorithm [86] at <https://github.com/anassinator/ilqr>. For RL, we tested open source implementation of the PPO [94] and TRPO [95] algorithms at <https://github.com/openai/baselines>. Default algorithm settings were used.

In iLQR, we use J_{path} directly as the cost function, where the weights and parameters are the same as our method. Tests indicated that iLQR needs a feasible initial guess and is

numerically sensitive to the time step and planning horizon. Any combination that would give a 9s horizon would cause the algorithm to fail completely, and could only achieve feasibility with a short horizon. Our tuning showed the best results at $dt = 0.005$ s and 0.25s horizon. We then apply iLQR in an online fashion, where the trajectory is re-planned after executing each optimized control. Because of the short horizon, iLQR is only able to obtain trajectories that shift the body slightly forward before tipping over.

For fairness of comparison, we implement a tuned PD joint controller and have joint position command be the action space, instead of torque for RL. The observation space includes the robot state, ankle poses, and the PD controller torques. The reward function is: $-10\|x_v - 0.4\|^2 - 0.2\|x_o\|^2 - 0.001\|a\|^2 - 10^{-5}\|u\|^2 + 1.0$, where a is a vector of the 4 feet angles and u is the control. In addition, we terminate an episode when $x_h < 0.25 \vee x_h > 1.5 \vee |x_o| > 1.1 \vee \|a\|_1 > 1.2$, and give a termination reward of -10 . The time step is 0.005 s and we run the training for 1 M time steps. Overall, results did not demonstrate stable gaits. The roll-out trajectories from the trained controllers caused the robot to fall quickly, and terminated in less than 50 time steps.

These comparisons demonstrate that locomotion on granular terrain is a challenging problem for standard trajectory optimization and RL due to its long time horizons and sensitivity to initial guesses and action choices. In comparison, our method is numerically favorable for long horizons and is fairly tolerant to infeasible initial guess for the NLP solver. However, due to its long computation time it is only suitable for offline gait generation.

3.4.5 Results on different terrain shapes

Next, we consider different terrain shapes. We test planar terrains of different slopes, ranging from -20° (downward slope) to 20° (upward slope), with 2.5° apart, and a wavy terrain. Our experiments find that steeper terrains are more challenging for the NLP solver. However, since we already have an optimized trajectory on a 0° terrain, we can obtain a high-quality initialization for similar terrains with minimal effort. Using this idea, starting from a 0° terrain, we track its optimized trajectory on the next slightly steeper terrain and use the tracked trajectory as initialization for this terrain. For the wavy terrain, we use the gait generated on 0° terrain as the reference trajectory during tracking. The parameters used here are the same as the 0° terrain, except for the optimizer’s parameters that are tuned using grid search. The performance metrics of the optimized trajectories are summarized in Table I. We also include the performance of a trajectory on the 10° terrain where we simply track the gait optimized on the 0° terrain (denoted as 10° NO). Selected optimized and non-optimized gaits are also displayed in Fig. 3.2. Animated results on all terrains are

available in the supplement video. Our method is able to find high-quality trajectories with small velocity error, except for the 20° , -17.5° and -20° terrains (for the -20° terrain, we use the optimized gait on the -15° terrain as reference during tracking), where the NLP solver fail to converge on the later two terrains. This is because these 3 steep terrains are the most challenging among all. We also observe that the steeper the terrain is, the more energy it consumes to cover the same distance, which matches our intuition. It is interesting that going downhill is actually more energy-consuming for the robot. In addition, looking at the non-optimized trajectory in Fig. 3.2.b, the robot ankles would slip, as shown by the foot position trace, which causes the robot to have much larger M_v and M_c . Meanwhile, the optimized gait in Fig. 3.2.a avoids slippage by optimally orienting the feet when touching the ground.

3.5 CONCLUSION

In conclusion, we propose a bilevel TO framework for generating locally optimized trajectories with learned contact models. We demonstrate through our experiments that we can generate high-quality trajectories offline with a small amount of parameter tuning effort. In addition, by leveraging optimized results on easier tasks, we are able to find trajectories on challenging terrains. One drawback is that we generate the trajectories offline for a specific contact model, while there exist a wide variety of contacts in the real world, and any variation in the composition of a granular terrain would lead to a different contact model. In future work, we could first generate a database of trajectories offline with different wrench spaces and learn a model that generates trajectories given a contact model. During online execution, we could first identify the contact model from gathering a small amount of data, query a trajectory from the learned model, and execute it. This follows the general idea of trajectory learning and meta-learning. In addition, we would like to test the trajectories on a physical robot.

CHAPTER 4: GRAY-BOX LEARNING FOR SIMULATION OF FORCE RESPONSE MODELS FOR HETEROGENEOUS ELASTIC OBJECTS

This chapter presents a gray-box learning method for simulating contact with elastically deformable objects. A point-based surface representation and a heterogeneous, nonlinear force response model are learned from a robotic arm acquiring force-displacement curves from a small number of poking interactions. A simulator then estimates displacement and force response when the deformable object is in contact with an arbitrary rigid object. It does so by estimating displacements by solving a Hertzian contact model, and sums the expected forces at individual surface points through querying the learned point stiffness models as a function of their expected displacements. Experiments on a variety of challenging objects show that our approach learns force response with sufficient accuracy to generate plausible contact responses for novel rigid objects.³

4.1 INTRODUCTION

Robotic manipulation of deformable objects has a number of applications, such as in-home environments, cable routing, textile handling, handling packaging in automated warehouses, and soft tissue modeling in biomedical settings, but it remains a topic of active research. Common deformation simulation methods, such as finite element method (FEM) or mass-spring modeling (MSM), require an object’s geometry and its material properties to be represented by a volumetric mesh model. Mesh models are challenging to build from noisy sensor data, and calibrating appropriate material constitutive parameters is a tedious, time-consuming, and error-prone process that depends on an appropriate mesh topology and constitutive equations. Data-driven approaches have been studied in robotic palpation of biological tissue, where a robotic probe gathers a dense field of data to estimate tissue stiffness and identify anomalies.

We envision that robots could use *naturalistic* data acquisition to model deformable objects, in which behavior is observed from a small number of interactions and the model is improved continuously during interaction. Humans can very quickly generalize from the data gathered during manipulation to predict plausible outcomes in vastly different scenarios. This chapter takes preliminary steps toward providing robots with similar capabilities. We take a gray-box approach that integrates machine learning with physics. This general idea is appealing because learning from data can enhance physics modeling when the physics

³This chapter is reproduced from Yifan Zhu, Kai Lu, and Kris Hauser, "Semi-Empirical Simulation of Learned Force Response Models for Heterogeneous Elastic Objects". In International Conference on Robotics and Automation (ICRA) 2020.

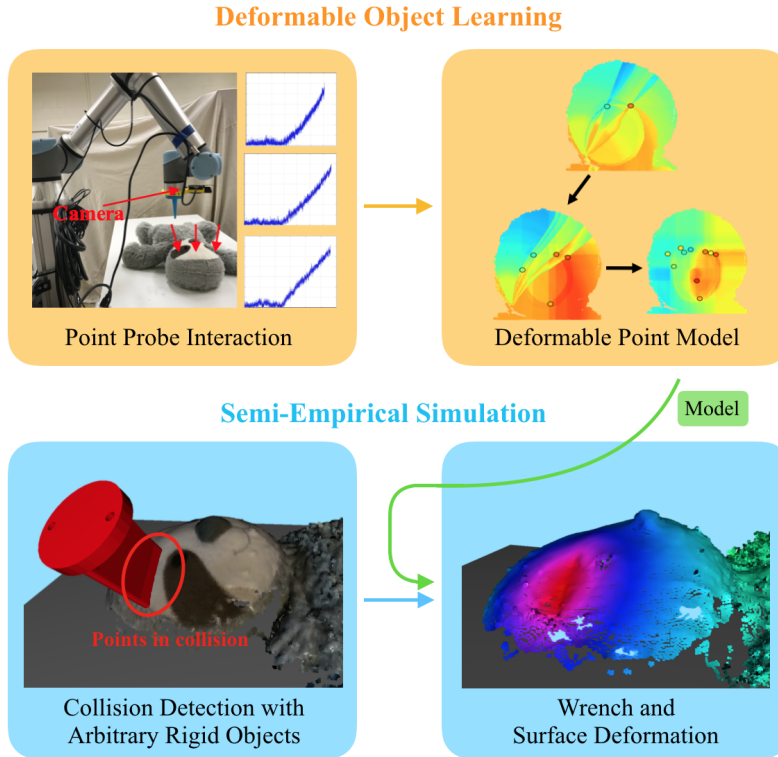


Figure 4.1: Flow of our method. A point interaction model of a deformable object is first learned through data acquired with the system shown here after a small number of pokes. The model is then used in a semi-empirical simulator that solves for the deformation and contact wrench with an arbitrary rigid object. The color of the deformed object represents the amount of surface deformation.

of certain phenomena are not well understood [2, 3, 4], when the parameters for the governing physics equations are hard to identify [6, 7], or when applying the physics equation is computationally prohibitive [48, 96]. In our approach, the simulator 1) accepts a point-based model of the object surface’s force response under displacements, which are learned from data, and then 2) calculates a resultant contact wrench for a novel rigid object using analytical calculations. Our model is quite general, as its deformation response can be heterogeneous and nonlinear. Moreover, the visuo-tactile model constructed by our method can overcome camera calibration errors, sensor noise, and “fuzzy” objects with ill-defined boundaries. It is important to note that *we do not expect a robot to need to touch an object hundreds of times during naturalistic interaction*. Our experiments examine performance under a tiny fraction of pokes as training data. Even with only a few touches, reasonably predictive models can be learned, which suggests that our pipeline is promising for on-line use in naturalistic interaction.

We evaluate this approach using an experimental testbed (Fig. 4.1) consisting of a robot

arm equipped with a RGB-D camera, a tapered probe attached at the end-effector, and a force-torque (F/T) sensor. Experiments show that our approach can learn a point deformation model that predicts force response 0.49 N of root mean squared error (RMSE) over 5 test objects with 10 interactions each. When making contact with novel rigid probes, which are a line-shaped paddle and a cylinder, our simulator’s force prediction has 1.5 N RMSE. This accuracy is comparable to learned models obtained with the same number of interactions. We also show an application of our method to planning for packing a rigid object on top of a deformable object into a box.

4.2 RELATED WORK

Deformable object modeling has long been studied in various fields including computer graphics, continuum mechanics, and biomechanics. An overview of the existing methods on deformable object modeling can be found in Ref. [97]. It still remains a challenge to build models that match real-world objects accurately both in deformation and force response.

Several researchers have studied the problem of building elastic models of real-world object using experimental data. Frank et al. attempt to build homogeneous FEM models by first collecting force probing data using a mobile robot arm, a thin probe, an F/T sensor and a RGBD camera [98]. An error minimization approach is used to estimate the FEM model parameters such that the simulations match recorded data. Boonvisut and Cavusoglu follow a similar scheme to build FEM models for elastic objects under stretching [99]. Both methods require tens of minutes to perform the error minimization, making it less likely to be used online. Bickle et al apply error minimization to inhomogeneous, nonlinear FEM models [65]. Radial basis functions (RBF) are used to interpolate the strain-stress relationship in each finite element, and error minimization optimizes the weights used for RBF interpolation to match observations. Although being able to handle more complex materials, the error minimization step is prone to being stuck in local minima and the data acquisition process requires complex apparatus setup, including multiple cameras, carefully positioned and tuned light sources, and drawing markers on the testing objects.

Empirical data have also been used to tune MSM and meshless methods. Deussen et al use simulated annealing to estimate the elastic properties of an MSM object model, given data from simulation experiments [100]. Burion et al solve a similar problem using a particle filter [101]. Pauwels et al. build meshless models of homogeneous 2D foams whose shape upon contact with a rigid object depends on a deformability constant. This constant is estimated by error minimization of real-world data. Because the method only simulates deformable shape, it is unable to predict forces at contacts.

A highly related topic is robotic palpation. Yamamoto et al use a robot equipped with an F/T sensor to poke at discrete locations on flat phantom tissue to estimate the stiffness across the object surface and find anomalies [102]. Salman et al. adopt a similar approach but select the next poking location using Bayesian optimization [103]. Goldman et al. first design a hybrid motion-force controller to follow cycloidal paths on a tissue surface to estimate the shape, then palpate at discretized surface locations with multiple grid resolutions to measure the impedance matrices across the surface [104]. Liu et al. map the stiffness by moving a mechanical roller across the tissue with a robot arm [105]. While these works are able to map the stiffness of heterogeneous tissues, force response is assumed to be linear.

4.3 POINT DEFORMATION MODEL LEARNING

Fig. 4.1 shows the flow of our overall system. The first stage of our system learns a visuo-tactile model of the deformable object using data. We shall discuss the learning stage in the current section, leaving the discussion of simulation for Section IV. We assume that the object is elastic, quasistatic, and that interaction forces are dominated by normal deformation rather than friction and shear deformation. We hope to relax these assumptions in future work.

The point deformation model consists of an *equilibrium surface model*, and a *force response model*. The equilibrium surface model is a set S of points $x \in \mathbb{R}^9$, with each point $x = [p_x, p_y, p_z, r, g, b, n_x, n_y, n_z]$, representing position, color, and (outward) normal, respectively. The position and normal are represented in the world frame. The force response model is denoted as $f = y(x, d)$, where $f \in \mathbb{R}$ is force magnitude in the normal direction, x is a point in S , and $d \in \mathbb{R}$ is displacement. Letting $\mathbf{n} = [n_x, n_y, n_z] \in \mathbb{R}^3$ denote the normal in x and $\mathbf{p} = [p_x, p_y, p_z] \in \mathbb{R}^3$ the position, this map gives the force a point probe would feel in the direction of \mathbf{n} when the point is moved to position $\mathbf{p} - d \cdot \mathbf{n}$.

4.3.1 Data Collection

Although our methods are applicable to naturalistic interactions, for the comparison purposes in this chapter we employ a separate data acquisition stage similar to Frank et al [98] to build a ground truth dataset. The deformable object is laid on a flat surface, and it is assumed to return to its original position when pressed in the normal direction. If it is found to move while pushed, then a central point on the bottom of the object is affixed to the table using Velcro tape. In future work it may be possible to simply track the movement of the object during acquisition. We include 5 different testing objects in our experiments, Sloth,

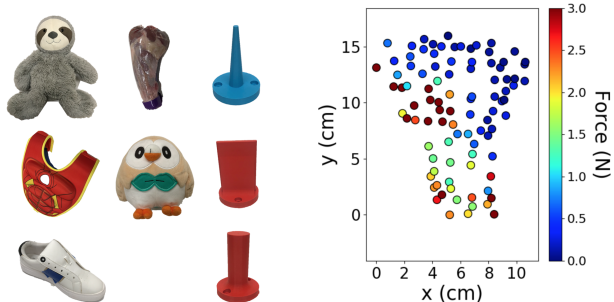
Vest, Lamb, Shoe, and Bird, shown in Fig. 4.2a. Sloth and Bird are stuffed animals, Vest and Shoe are clothing items, and Lamb is a piece of lamb leg consisting of muscle, fat, and bone. Each object has varying stiffness across the object surface. The regions of interest of the objects are the surfaces facing up when the objects are laid flat on a table, with the exception that we use only the face region of the sloth due to poor visual perception on the furry torso.

To build a 3D point cloud of the equilibrium surface, a wrist mounted RGB-D camera is used. Some objects need a few scans from different angles to generate a complete point cloud. Open3D [106] is applied to perform outlier removal and normal estimation.

To capture ground truth force response data, we randomly select a set of locations on the object to be poked with a point probe, shown in Fig. 4.1. During experiments, we collect data on a uniform grid, and randomly select among these locations to be training data, and the rest to be testing data. To complete one probing action at a point x , we align the point probe’s axis with the point’s surface normal. Starting at a user-specified distance from the surface, the probe pushes in the negative normal direction at a constant speed of 2mm/s until reaching a force threshold, which is set to 3-5 N in our experiments depending on the overall stiffness of the testing objects. The amount of displacement and force in the normal direction, measured by the wrist mounted F/T sensor, is recorded at 250 Hz. As a result, our force measurements are quite sparse in the spatial features, while being quite dense in force / displacement. One example of the collected data on the Lamb is shown in Fig. 4.2b.

If both the visual and tactile data have no error, then we would expect the force felt by the point probe tip away from the object surface to be zero and positive otherwise. However, this is often not the case. As shown in Fig. 4.4a, we often observe nonzero forces beginning anywhere from -5 to 5 mm of displacement. This is largely due to error in the calibration and bias in the camera’s depth estimates, and further motivates the use of learning to correlate visual and tactile data.

For validation purposes, we collect second and third sets of poking data with a line probe and a cylindrical probe (capped at 3-14 N, depending on the objects), shown in Fig.4.2a. These are not used for training, but only for evaluation of how well our simulation model can generalize from point probe data. We probe once at each location using the cylinder probe, and three times with the line probe at each location with random orientation changes (about the local probe x-axis). We also record the torque about the EE frame center. The parameters of all three datasets are summarized in Table 1.



(a) Sloth, Vest, Lamb, (b) Ground truth data on Shoe, and Bird objects, Lamb, with color showing force felt at $d = 0.5$ mm. used in testing.

Figure 4.2: The objects and probes used in the experiments. Data is gathered using the point probe, and the simulator generalizes force predictions to the line and cylinder probes.

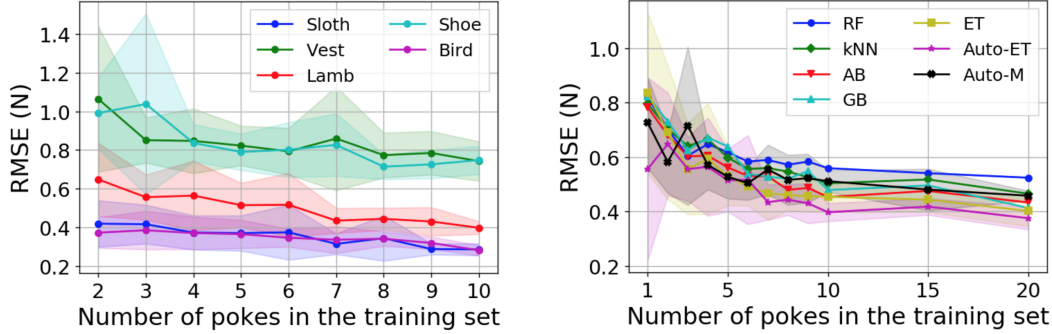
	Point Cloud	Point Probe	Line Probe	Cylinder Probe
Sloth	67,923	121	363	121
Vest	62,318	116	348	116
Lamb	18,059	94	276	94
Shoe	32,408	96	192	96
Bird	41,515	116	315	116

Table 4.1: Statistics of Collected Data on the Testing Objects

4.3.2 Force Response Model Learning

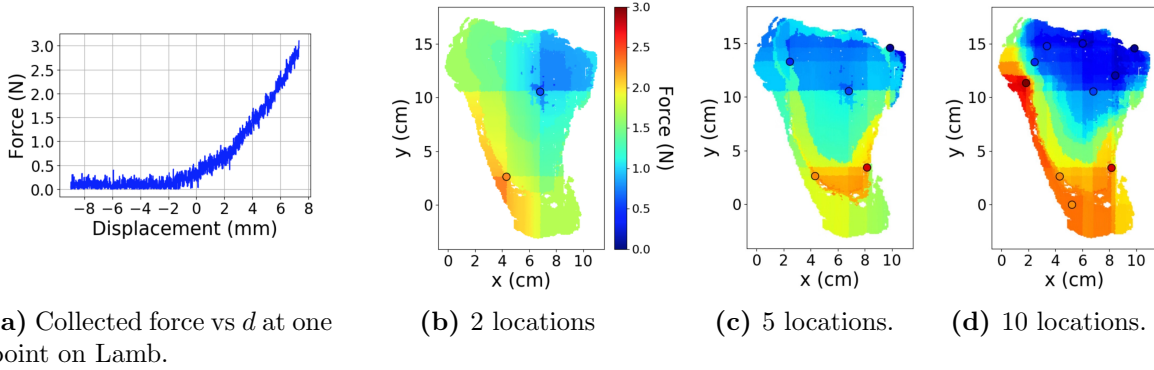
The next step in our pipeline learns to predict force responses at different locations on the object surface. Recent advancements in automated machine learning (AutoML) have greatly reduced the amount of manual effort needed to select a regression model and tune hyperparameters. We use the AutoML [107] package from scikit-learn [108] to perform the regression tasks.

We compare performance of different types of regressors. The AutoML package searches among ensembles of different types of regressors and preprocessors and runs until a user-specified time limit is reached. We can also specify the type of regressors during AutoML training to reduce the number of hyperparameters that need to be optimized, thus improving training speed. Another option is to select an appropriate regressor directly, without relying on AutoML to select the hyperparameters. In particular, we test the following methods: (1) AutoML (Auto-M) trained for 360s (2) AutoML with Extra Trees as the regressor type (Auto-ET) trained for 30s; (3) Adaboost (AB), Gradient Boosting (GB), Random Forest (RF), and Extra Trees (ET), all trained using scikit-learn with default parameters. Fig. 4.3b show the results on the Lamb model. Although single regressors can train a model in less 1s, Auto-ET generally achieves the best results in 30s. If Auto-M were to be given more



(a) Testing error on each object using Auto-ET, with varying training set size. (b) Comparing testing error between regressors, over all objects.

Figure 4.3: Breakdown of learning curves for learning force response models. Regressors are trained over 10 sampled subsets of the ground truth dataset, and plots show RMSE mean and variance across samples.



(a) Collected force vs d at one point on Lamb. (b) 2 locations (c) 5 locations. (d) 10 locations.

Figure 4.4: Collected force-displacement curve at one point and force predictions on the Lamb object, learned with different numbers of poking locations. Poking locations are indicated as circles, colored by the true force. d is set to be 0.5 mm. Because of camera bias, there is already a significant amount of force at this depth. [Best viewed in color]

training time, it could potentially learn a better model, but doing so would require hours or even days of training. We settle on Auto-ET for the rest of the chapter given its balance between speed and accuracy on our dataset. The resulting force predictions that evolve after 2, 5, 10 locations are used as training data for a 0.5 mm displacement on Lamb are shown in Fig. 4.4.

Results with Auto-ET on all 5 testing objects are shown in Fig. 4.3a. The Shoe and the Vest are the hardest to learn because they are not completely elastic and are made of composite materials, resulting in significantly varying stiffness across their surface.

Prediction speed per point varies with the query size, because of query overhead. With 1 point in each query, it takes ≈ 17 ms per point, while with 10,000 points per query, it takes ≈ 0.015 ms per point. Also, prediction speed is not directly related to the number of training points.

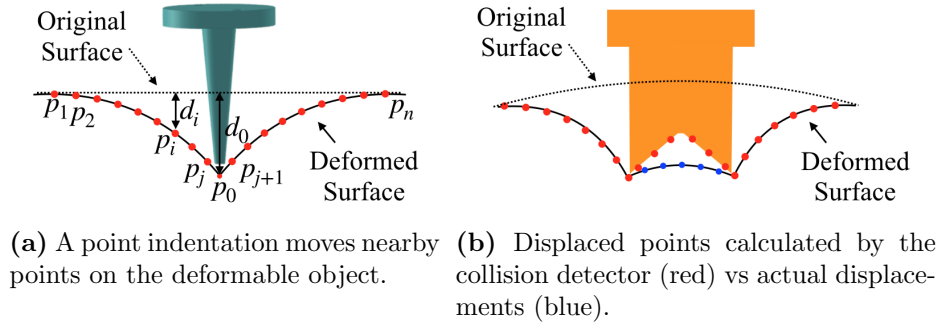


Figure 4.5: Definitions used in the surface deformation model.

4.4 GRAY-BOX SIMULATOR

The second phase of our pipeline estimates reaction forces between the deformable object and a contacting rigid object. The rigid object is represented as a 3D surface mesh M defined in the global frame. The contact detector determines if it is in contact with the deformable object’s equilibrium surface model S , and if so, which points must be displaced to resolve the contact. The contact force solver then computes the 3D reaction force and torque and the deformed shape of the object.

4.4.1 Contact Force Solver

Classic Herztian contact theory predicts that displacement at a surface point on a deformable object will cause neighboring surface points to move in a manner described by a function of distance and material properties (Fig. 4.5a). Hence, we cannot simply sum the forces caused by each independent point displacement, but instead have to account for correlations in displacement. We assume that normal forces and displacements are the dominant effect, and experiments on our object set suggest this assumption is reasonable for normally-displaced objects.

Let d_i denote the *nominal displacement* of a surface point at a contact patch, we assume that it consists of both displacement caused by neighboring point displacement and that actually contributes to the total force, which we call *actual displacement* and denote as \tilde{d}_i . We solve for non-negative actual displacements that are no greater than the nominal displacements. A smaller nominal displacement would indicate penetration, while a negative actual displacement would cause the deformable to apply a negative (pulling) force on the rigid object at that point (Fig. 4.5b).

Nominal Displacement Calculation To calculate nominal displacements of the deformable object’s surface points we use collision detection and estimate penetration depths.

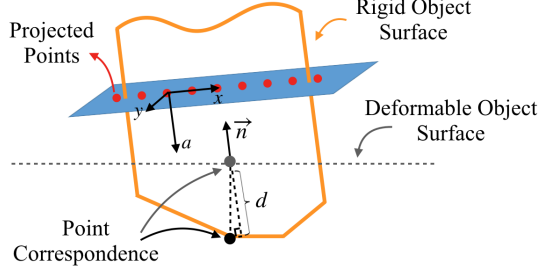


Figure 4.6: An illustration of how the nominal displacements are calculated for a polyhedral rigid object.

To do so, we first discretize the surface of the rigid object by placing equally spaced points on edges and 2D grids on facets based on a user-selected discretization. Via experimentation, we found that simulation results are relatively insensitive to the selection of this discretization, and we use a 3 mm resolution for all experiments.

Next, we find the pairs of rigid surface points and the deformable surface points that will be touching after deformation and the nominal displacement of the deformable surface points. Point correspondence is calculated by projecting all the points to a 2D plane and performing nearest-neighbor searches. We let \mathbf{a} be the approach direction of the rigid object and the normal of the projection plane, shown in Fig. 4.6. To better approximate the shape of the underlying continuous surface, we average the nominal displacements of the the 3 nearest deformable surface points for each rigid surface point. Each nominal displacement is calculated according to

$$d_i = -(\mathbf{p}_{rigid_j} - \mathbf{p}_{deformable_i})^T \cdot \mathbf{n}_{deformable_i}, \quad (4.1)$$

where $\mathbf{p}_{deformable_i}, \mathbf{n}_{deformable_i} \in \mathbb{R}^3$ are the equilibrium surface position of point i and its associated normal. We average the equilibrium positions and nominal displacements of the 3 points to obtain the *estimated equilibrium surface point* and displacement, which we refer to as \mathbf{q} and d . By keeping only the points with $d > 0$, we have N pairs of rigid surface points and its corresponding estimated equilibrium surface points along with nominal displacements.

Force Solving The force solver uses the nominal displacements calculated above combined with an indentation basis function to compute the actual displacements. The shape function is defined as follows. For each pair of points on the deformable object, we define a shape function $g(r) : \mathbb{R} \rightarrow \mathbb{R}$ to describe the correlation of a point indentation on its neighborhood. Here r is the Euclidean distance between 2 points on the equilibrium surface, and we use

the inverse function:

$$g(r) = c/(c + r) \quad (4.2)$$

$g(r)$ describes an inverse relationship between points' relative displacement and equilibrium surface distance, which matches the Hertzian theory. The constant $c > 0$ is picked by observation. However, for future work, we could determine this constant using a camera tracking the shape of deformation during poking. While this constant may actually vary across the object surface, we keep it constant across the object in this chapter.

By calculating the relative distance $r_{i,j}$ between all pairs of deformable points \mathbf{q}_i and \mathbf{q}_j in collision with the rigid object, we obtain a linear complementary problem (LCP) relating nominal and actual displacements:

$$\mathbf{w} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{bmatrix} - \begin{bmatrix} g(r_{1,1}) & g(r_{1,2}) & \dots & g(r_{1,n}) \\ g(r_{2,1}) & g(r_{2,2}) & \dots & g(r_{2,n}) \\ \vdots & \vdots & \ddots & \vdots \\ g(r_{n,1}) & g(r_{n,2}) & \dots & g(r_{n,n}) \end{bmatrix} \begin{bmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \vdots \\ \tilde{d}_n \end{bmatrix} \quad (4.3)$$

$$0 \leq \mathbf{w} \perp [\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_n]^T \geq 0,$$

where the subscripts denote the indices of the deformable surface points in collision. When there are not any duplicated points, i.e. $g(r_{i,j}) \neq 1$ except for $i = j$, this symmetric matrix is invertible. We solve the LCP by iteratively assuming equalities, solving the system of linear equations by matrix inversion, and removing the points with negative actual displacements.

From the actual displacements we sum the forces predicted by the point deformation model to obtain the contact wrench. Given the N_{final} points with positive actual displacements, the total response wrench (F_{total}, τ_{total}) on the object are calculated as follows:

$$\begin{aligned} f_i &= y([q_i, d_i]) - y([q_i, d_i - \tilde{d}_i]) \\ F_{total} &= \sum_{i=1}^{N_{final}} f_i \cdot \mathbf{n}_i \\ \tau_{total} &= \sum_{i=1}^{N_{final}} (\tilde{\mathbf{r}}_j) \times (f_i \cdot \mathbf{n}_i), \end{aligned} \quad (4.4)$$

where $\tilde{\mathbf{r}}_j$ is the vector from the torque center to the corresponding rigid body point \mathbf{p}_{rigid_j} .

To predict the displacement of the entire object, not only the points in contact, we sum up contributions from the shape function over every positive actual displacement.

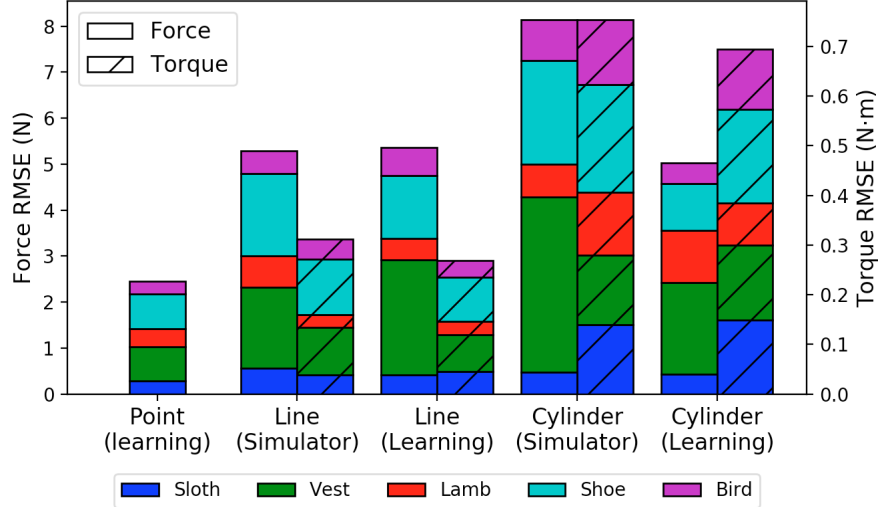
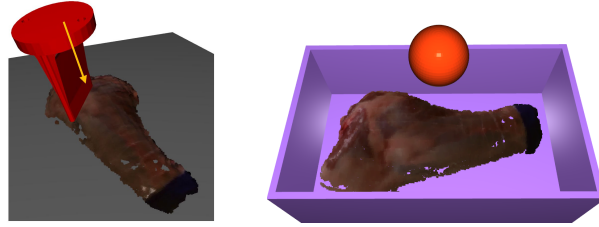


Figure 4.7: Predictions errors of our simulator on novel rigid objects and that of pure data-driven models. Errors on different objects are shown in different colors. We also include the errors of the point models used in predictions as a reference.

4.4.2 Experiments

Throughout this section, we use the point deformation models trained with Auto-ET and data from 10 poking locations. All implementation is performed in Python code with the Numpy package for linear algebra, and all timing is conducted on a desktop PC with an Intel i7 3.8GHz processor.

Simulation accuracy for Line and Cylinder contact Fig. 4.7 shows evaluation of our simulator with line and cylinder probe contact in predicting 3D force and torque vectors. The simulator prediction errors are compared to pure learning using the correct geometry, trained directly on 10 pokes from the ground truth line and cylinder probing data, where Auto-ET is also used as the regressor. We notice that the actual displacements calculated could exceed the range of displacements in the point training data. We simply extrapolate the training data following the estimated force-displacement slope to incorporate all possible queries from the simulator. Overall, our simulator performs slightly worse than pure learning, better on Lamb and Bird, and worse on Vest and Shoe. We believe this is mainly due to the fact that Vest and Shoe are made of composite materials with a mixture of elastic and inelastic components. As a result, shape functions approximate the deformation of these objects sub-optimally and lead to errors in the simulator. The key advantage of our method is that we do not need to re-acquire a new training set, and we can also predict force response for different rolls and pitches of the rigid objects, which are not included as features in the learning-only training set.



(a) Contact with a line probe. (b) Robot packing setup.

Figure 4.8: Simulation experiments, Lamb object.

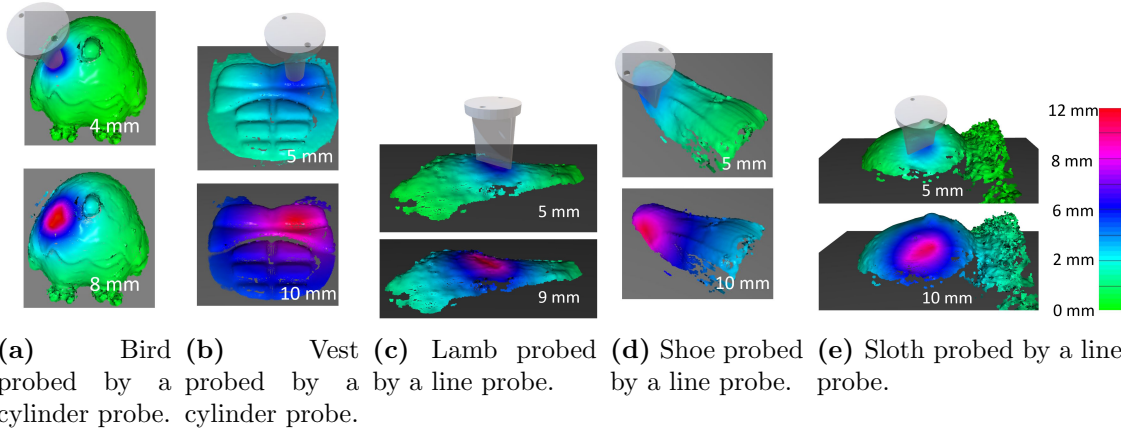
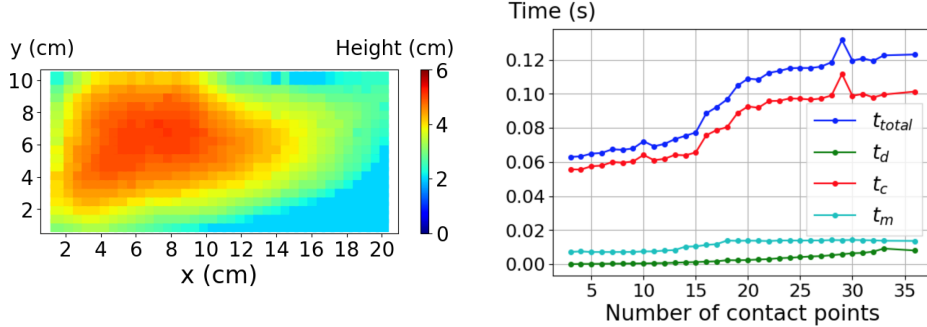


Figure 4.9: Simulation of the line and cylinder probes poking straightly at all 5 experiment objects, with the numbers indicating the amount of probe displacements. The objects are colored with the displacements of the surface points, caused by the probes. [Best viewed in color]

Simulation Visualization We illustrate our simulator by visualizing the simulation of a line or cylinder probe poking straight at different objects, resembling the data collection process in Section III.A. An example is shown in Fig. 4.8a. The simulation frames are displayed in Fig. 4.9. As the probes penetrate deeper, the calculated actual displacements also increase, thus deforming the entire surface more significantly, calculated by the shape function. We can see grooves on the Lamb, the Bird, and the Sloth that fit the probe geometry at deep penetration. The Shoe and the Vest are stiffer objects, with c in the shape function much greater than the other 3 objects. As a result, the grooves are harder to observe.

Application to a Packing Problem We consider a simple packing problem as an illustration of our approach, shown in Fig. 4.8b. Given a rigid box with a deformable object (the Lamb) already packed, we would like to know if the robot can pack another rigid item (a sphere, with diameter = 40 mm and discretized into 150 surface points) into the box. In particular, we would like to know at what positions in the box the sphere can be placed such



(a) The minimum height at which the rigid sphere can be packed under force threshold 2N. (b) Computation time vs number of contacts, showing total time (t_{total}), displacement calculation (t_d), collision detection (t_c), and point model queries (t_m).

Figure 4.10: Results from the packing planning problem.

that its interaction force with the deformable object does not exceed 2 N. We perform a grid search over the x-y-z position of the rigid sphere centroid inside the box, where z starts with the maximum allowed height in the box and finishes the search in the negative z direction once the interaction force limit is exceeded, shown in Fig. 4.8b. A plot of minimum possible height of the sphere on the x-y grid is shown in Fig. 4.10a.

On average in this experiment, a contact reaction force query takes 0.086 s. Each query consists of 0.0738 s of collision detection, 0.010 s of querying the point model, and 0.0017 s of LCP solving by our simulator. The average computation time as a function of the number of detected contact points m is plotted in Fig. 4.10b. The time for LCP increases cubically with m , because it mainly involves inverting a dense matrix. The time for querying the point model stays somewhat constant. There is a lot of room for improvement in speed, including implementation in a compiled language, better collision detection algorithms [109], and parallelization.

4.5 CONCLUSION

The chapter proposed a 2-stage gray-box framework to learn and simulate the force responses of heterogeneous elastic objects. In the first stage, a point deformation model of a testing object is learned via a robot arm poking at the object a few times. In the second stage, a simulator predicts the contact wrench between the deformable object and a novel rigid object by integrating analytic calculation and the learned point model to obtain the expected restoring forces at a series of contact points and summing them.

For future work, we would like to relax several assumptions made. First, learning and

generalizing shear and friction forces with small numbers of touches is an open problem. Perhaps the deformation and force response of a novel deformable object can be bootstrapped by transfer learning from an object that has already been acquired. We would also like to learn the deformation and force response of an object while it is undergoing gross movements, separating the rigid body and elastic components of the movement. This could also be extended to handle plastic deformation. Finally, we would like to enhance our simulator to handle combined rigid and elastic deformation, as well as contact between deformable objects.

CHAPTER 5: AUTOMATED HEART AND LUNG AUSCULTATION IN ROBOTIC PHYSICAL EXAMINATIONS

So far in this thesis, learning is done in a supervised manner, where a small labeled dataset is provided to a machine learning algorithm. This chapter instead presents an active sensing approach where the robot plans information-gathering paths. The specific application is auscultation, where this chapter demonstrates the first implementation of autonomous robotic auscultation of heart and lung sounds. To select auscultation locations that generate high-quality sounds, a Bayesian Optimization (BO) formulation leverages visual anatomical cues to predict where high-quality sounds might be located, while using auditory feedback to adapt to patient-specific anatomical qualities. Sound quality is estimated online using machine learning models trained on a database of heart and lung stethoscope recordings. Experiments on 4 human subjects show that our system autonomously captures heart and lung sounds of similar quality compared to tele-operation by a human trained in clinical auscultation. Surprisingly, one of the subjects exhibited a previously unknown cardiac pathology that was first identified using our robot, which demonstrates the potential utility of autonomous robotic auscultation for health screening.⁴

5.1 INTRODUCTION

Infectious diseases pose substantial risks to healthcare providers and there have been worldwide initiatives to use robots and automation to help combat the COVID-19 pandemic [110]. In particular, performing physical examinations with robots has potential to reducing the risks to healthcare providers by minimizing person-to-person contact. Moreover, robotic health screening can also help promote preventative care and affordable routine checkups, particularly in rural, remote, and low-resource communities.

Motivated by the benefits of automating physical examinations, this chapter explores the task of auscultation, i.e., listening to heart and lung sounds via a stethoscope, which is an important screening procedure that is performed at regular checkups and for patients exhibiting respiratory and cardiac symptoms. This chapter demonstrates the first robotic system capable of performing automated heart and lung auscultation, based on the TRINA robot shown in Fig. 5.1. The design of a newer version of this robot, also name AVATRINA, is discussed in detail in Chapter 8. Prior tele-medicine robotic systems have been built for performing auscultation and echocardiography [111, 112] through tele-operation. Compared

⁴This chapter is reproduced from Yifan Zhu, Alexander Smith, and Kris Hauser, "Automated Heart and Lung Auscultation in Robotic Physical Examinations". In IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 4204-4211, April 2022.

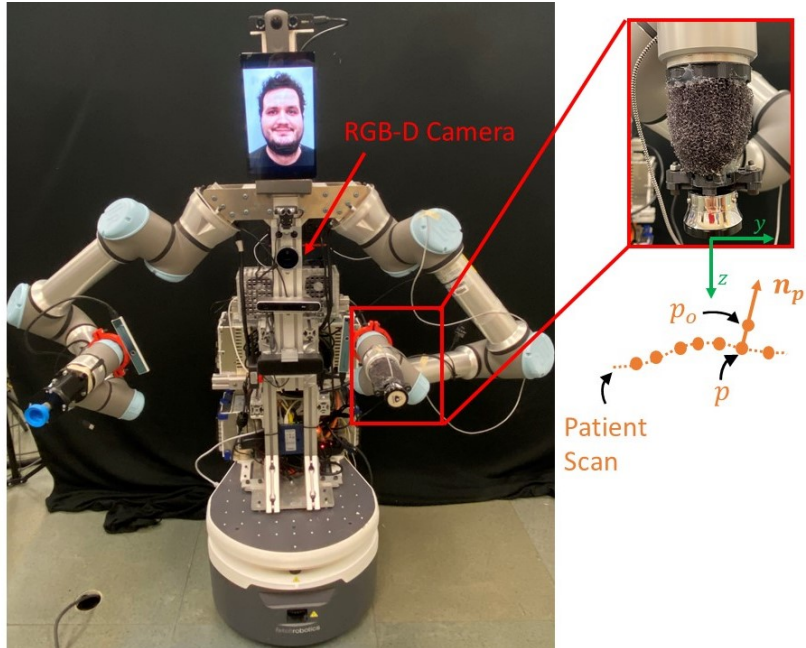


Figure 5.1: The TRINA robot used in the experiments, with the zoomed-in view of the stethoscope on the right. The stethoscope frame is shown in green, and the patient scan point cloud in orange. The design of a newer version of this robot, also name AVATRINA, is discussed in detail in Chapter 8. [Best viewed in color.]

to tele-operation, automated medical exams offer several potential benefits including lower operation time, a shorter learning curve for physicians, and reduced cognitive load and tedium. Auscultation is, however, a challenging process to automate. The quality and interpretability of sounds produced by the anatomical structure of interest at the skin surface is highly variable and depends on the location, pressure, and steadiness of the stethoscope. If the stethoscope is placed at a poor location in relation to the patient’s internal anatomical structures (e.g., over a rib or deep layers of fatty tissue), the sound may be attenuated or muffled in frequencies that are critical for diagnosis. A human doctor uses visual, tactile, and audio feedback as well as anatomical information and diagnostic expertise from prior medical training to localize informative listening locations.

The goal of our system is to record high-quality sounds that a human doctor will listen to, so the robot’s aim is to provide sounds of diagnostic utility rather than to perform diagnosis itself. First, our system captures a 3D point cloud scan of the patient, registers a human body model, estimates the locations of key anatomical landmarks, and produces a prior map of high-quality auscultation locations. It then adopts informative path planning using audial feedback to adaptively search over the region of interest for a high-quality auscultation location. Audial feedback relies on *sound quality estimators* trained on a database of heart and lung stethoscope recordings. To determine the optimal sensing location we formulate a

Bayesian Optimization (BO) problem where the unknown sound quality field is estimated as a semi-parametric residual Gaussian Process (SPAR-GP) model, with a prior map that depends on latent translation offset and sound quality scaling parameters.

Experiments on 4 healthy male human subjects demonstrate that our system performs heart and lung auscultation automatically, and locates sounds of similar diagnostic quality and execution time as tele-operation by a human trained in auscultation in medical school, and trained to use the robot by the investigators. The procedure is reliable and largely required no supervision, except for one subject where a perception error required anatomical landmarks to be input manually before auscultation. Although each subject reported good health and no respiratory or cardiac ailments, in a surprising discovery one subject exhibited a heart murmur that was identified during these experiments. The subject was thereafter referred to a physician for follow-up tests.

5.2 RELATED WORK

5.2.1 Robotic Remote Auscultation and Sonography

Tele-nursing robots consisting of a mobile base and arms have demonstrated a variety of nursing tasks including handovers, vital signs monitoring, disinfection, and auscultation [113, 114, 115, 116]. Robotic technologies employed in the fight against COVID-19 and infectious diseases are reviewed in recent surveys [117, 118].

Giuliani et al. develop a robot and tele-operation user interface for echocardiography [111]. Several works have studied performing robot remote lung sonography using the MGIUS-R3 robotic tele-echography system produced by MGI Tech Co, Ltd. [119, 120, 121, 122]. Marthur et al. develop a semi-autonomous robotic system to perform trauma assessment remotely, where the locations on a patient to be assessed are identified automatically via perception and sonography is performed with tele-operation [123]. Instead of tele-operation, our proposed method performs auscultation fully automatically, which has the potential to reduce physician learning curve, cognitive load, and tedium. On a level of autonomy (LoA) scale from 0 (Full Manual) to 5 (Full Autonomy) [124], our system achieves a level of 4 (High-level Autonomy) because the robot autonomously performs the procedure while an operator monitors and intervenes when necessary.

5.2.2 Informative Path Planning

Our problem falls under the category of informative path planning (IPP), where the robot plans information-gathering paths given a probabilistic model of the quantity of interest. Sensor placement, active learning, and adaptive sampling problems can be seen as special instances of the IPP problem. IPP has been applied to a variety of robotics applications such as object tactile exploration [125], palpation-based tissue abnormalities detection [103, 104, 126], inspection and environment mapping [127, 128, 129].

Most related to ours are the works by Salman et al. [103] and Ayvali et al. [126]. Salman et al. propose both discrete and continuous trajectory search algorithms for robots to search for the boundaries of tissue abnormalities, i.e, high stiffness, through robotic palpation. Gaussian process (GP) regression is used to model the distribution of stiffness. Four methods, including BO, active area search, active level sets estimation, and uncertainty sampling are compared, among which active area search had the best performance. Ayvali et al. also perform robot palpation to detect tissue abnormalities, using GP regression and BO. To inject user prior knowledge of the locations of tissue abnormalities, they add a decaying utility function whose value peaks at the user-provided locations to the acquisition function of BO. In contrast, our work adopts a discrete search BO to the auscultation setting, and introduces two technical contributions to make auscultation more practical. First, visual registration of anatomical landmarks eliminates the need for manual input to specify priors, except as a backup for registration failures, and second, the use of our SPAR-GP model leads to faster convergence than pure residual GP models.

Belonging to the family of semi-parametric regression methods, SPAR-GP performs function approximation with a combination of parametric functions and Gaussian process. In robotics, similar models have been applied to system identification of linear and nonlinear dynamical systems [15, 16, 130], but to our knowledge semi-parametric models have not been adopted in the informative path planning setting.

5.3 METHOD

The overview of our method is shown in Fig. 5.2. In the *offline phase*, we build a dataset of heart and lung stethoscope recordings of humans with labeled sound qualities and train heart and lung sound quality estimators (Sec. 5.3.2). The goal of our system is to generate high (estimated) quality heart and/or lung recordings by placing a stethoscope at anatomically relevant portions of a patient’s chest and back, while keeping the number of auscultation actions small. These estimators are the only feedback used by the system during the online exam; we do not assume human raters are available to provide ground truth.

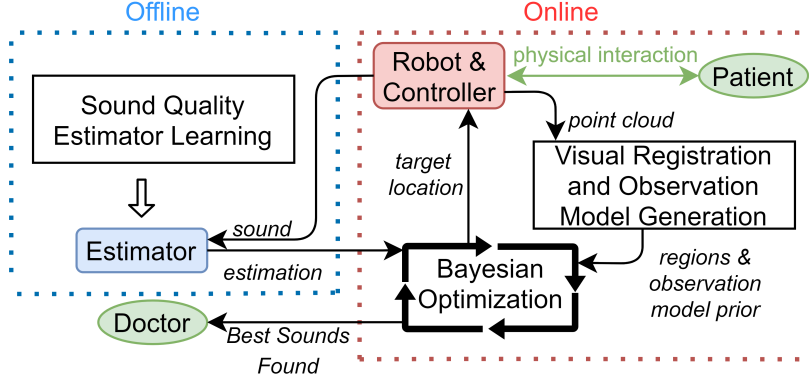


Figure 5.2: The workflow of our method.

During the *online phase*, the robot first performs visual registration of the patient to a reference human model with labeled *clinical auscultation locations*. An *observation model prior* is constructed based on the visual registration (Sec. 5.3.3), which encodes anatomical prior information about expected sound quality. Finally we use BO (Sec. 5.3.1) to select locations to auscultate to find the best sounds utilizing both the learned sound quality estimators and prior information from visual cues. These targets are then used for motion control (Sec. 5.3.4).

5.3.1 BO Formulation

We use BO to search adaptively for an auscultation location that yields a high quality sound within a specified anatomical region. We denote sound quality estimators for heart and lung sounds as $e_s(r)$, where s is an anatomical structure (heart or lung) and r is a stethoscope recording. The BO approach computes a probabilistic estimate of the unknown field $e_s(r(x))$ across the patient surface using a SPAR-GP model, which is a sum of the observation prior and the GP. An acquisition function is optimized to yield the new auscultation location. Given a new observation, the estimate is re-fit to the data and the process repeats until a termination criterion is met.

SPAR-GP A GP models an unknown function f as a collection of random variables $f(x)$ which are jointly Gaussian when evaluated at locations x . A GP is fully specified by its mean function $m(\cdot)$, which we set to 0, and covariance function $k(\cdot, \cdot)$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \quad (5.1)$$

Given n existing observed function values $\bar{y} = [y_1, \dots, y_n]$ at $\bar{x} = [x_1, \dots, x_n]$, GP regression

predicts the function values at new point x^* as a Gaussian distribution:

$$P(f(x^*)|\bar{\mathbf{x}}, \bar{\mathbf{y}}, x^*) \sim \mathcal{N}(\mathbf{k}\mathbf{K}^{-1}\bar{\mathbf{y}}, k(x^*, x^*) - \mathbf{k}\mathbf{K}^{-1}\mathbf{k}^T) \quad (5.2)$$

Here,

$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} + \sigma^2 \mathbf{I} \quad (5.3)$$

$$\mathbf{k} = [k(x^*, x_1), \dots, k(x^*, x_n)], \quad (5.4)$$

where σ is the standard deviation of noise at an observation.

For each anatomical location, the observation model is a sum of a parametric prior mean function $\mu_\theta(x)$ and a GP residual function $f_s(x)$: $e_s(r(x)) \approx \mu_\theta(x) + f_s(x)$. Since the GP models residuals with respect to the prior, we subtract the prior from the sound quality as the GP observations: $y_i = e_s(r(x_i)) - \mu_\theta(x_i)$. Here we further denote the history of the estimated sound qualities as $\bar{\mathbf{e}} = [e_1, \dots, e_n]$.

Although the anatomical reference provides sound quality peaks assuming an average human and perfect registration of anatomical landmarks, the robot's prior should capture the uncertainty in visual registration error and the effect a patient's body type has on the magnitude of the overall sound quality. Therefore, we make the prior μ_θ a parametric function of the latent variables θ representing the translation offset and sound quality scaling, and infer θ from observed sound qualities. In particular, a reference quality map $\mu_o(x)$ is first generated from visual registration and θ is initialized to θ_o , which includes zero translation offset and scaling of 1. The exact composition of the prior mean function is deferred to Sec. 5.3.3. θ is inferred after each reading using the history $\bar{\mathbf{x}}$ and $\bar{\mathbf{e}}$ with maximum a posteriori (MAP) estimation. We use a likelihood function $\mathcal{L}(\theta|\bar{\mathbf{x}}, \bar{\mathbf{e}}) = \prod g(e_i|\mu_\theta(x_i), \sigma^2)$, where $g(\cdot|\mu_\theta(x_i), \sigma^2)$ is the probability density function of the Gaussian distribution $\mathcal{N}(\mu_\theta(x_i), \sigma^2)$. The prior of θ , $h(\theta)$, follows a multivariate Gaussian distribution $\mathcal{N}(\theta_o, \Sigma)$. We solve the MAP estimation problem by maximizing the posterior, using a standard numerical optimization solver:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \mathcal{L}(\theta|\bar{\mathbf{x}}, \bar{\mathbf{e}})h(\theta) \quad (5.5)$$

BO Letting A be a region of interest on the patient surface near structure s , BO aims to solve:

$$\max_{x \in A} e_s(r(x)) \quad (5.6)$$

Since heart and lung sounds need to be listened separately and there are usually multiple parts of the anatomical structures that are of interest, we solve Eqn. 5.6 for each anatomical structure s separately, and for each s sequentially for their regions of interest. We also share the same GP across the entire chest or back of a patient for the same s .

In each iteration of BO, an acquisition function is used to determine the next location to observe the data. Two popular choices are expected improvement (EI) and upper confidence bound (UCB). Let the posterior mean and variance of the GP be $\mu_{\bar{y}}(x), \sigma_{\bar{y}}^2(x)$, then EI is defined as:

$$EI(x) = \mathbb{E}[\max(0, \mu_{\theta}(x) + \mu_{\bar{y}}(x) - e^*)], \quad (5.7)$$

where e^* is the best observed quality so far, and EI can be calculated in closed form [131]. UCB is defined as:

$$UCB(x) = \mu_{\theta}(x) + \mu_{\bar{y}}(x) + \beta\sigma_{\bar{y}}(x), \quad (5.8)$$

where β is an exploration weight that regulates how much bonus is given to uncertainty in the prediction.

The overall algorithm is outlined in Alg. 5.1. For each iteration in each region, a point with the largest acquisition function value is first chosen to be observed, where $\xi(x)$ represents the acquisition function, and the quality is estimated. In particular, A contains sampled points on a patient surface, and the maximization is performed in a brute force fashion by calculating the acquisition function values across all points in A and selecting the maximum. In addition, the same point is observed at most once. Then we add the estimated qualities to the history and update the prior mean function by solving the MAP estimation, after which the residuals for GP are calculated, observations are added, and the GP is re-fit. For the termination criteria, we allow early termination if the estimated quality in a region is above an adequacy threshold for making a diagnosis. We also set the budget N_{max} to a reasonable number of auscultations per region to limit the maximum time of the procedure.

5.3.2 Sound Quality Estimator

Dataset We first collect lung and heart stethoscope recordings of various qualities with TRINA from the researchers on this project. Each recording lasts 10s, and we vary the locations on the subject’s chest, whether or not the subject is wearing clothing, and whether the subject is taking a deep breath. 80 recordings are collected each for heart and lung sounds. The ground truth qualities of the recordings were labeled by 4 medical school students who have undergone auscultation training in a Liaison Committee on Medical Education (LCME)-accredited medical program and have been applying their auscultation skills

Algorithm 5.1: Bayesian Optimization Auscultation

Input: Structure s (heart/lung), prior μ_θ , regions A , max iterations N_{max} ;
Initialize $\bar{\mathbf{x}} = \{\}$, $\bar{\mathbf{y}} = \{\}$, $\bar{\mathbf{e}} = \{\}$;
for all regions A near s **do**
 for $k = 1, \dots, N_{max}$ **do**
 $x_k \leftarrow \arg \max_{x \in A} \xi(x)$;
 if region termination criteria met **then**
 Go to next region;
 else
 Auscultate at x_k , obtain sound quality e_k ;
 Set $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}} \cup \{x_k\}$, $\bar{\mathbf{e}} \leftarrow \bar{\mathbf{e}} \cup \{e_k\}$;
 $\theta \leftarrow \arg \max \mathcal{L}(\theta | \bar{\mathbf{x}}, \bar{\mathbf{e}}) h(\theta)$;
 Set $\bar{\mathbf{y}} \leftarrow \bar{\mathbf{y}} \cup \{e_k - \mu_\theta(x)\}$;
 Re-fit GP;
 end
 end
end
return recordings of max quality in each region;

in clinical settings for at least one year. We note that prior studies suggest that the cardiac examination skills of trained medical school students are no worse and may even be better than those of experienced doctors [132]. Each label is a score between 0-1 with increments of 0.125 based on the rating guidelines provided by Grooby et al., where 0 means no detectable heart or lung signal, and 1 means clear heart or lung sounds with little to no noise [133]. A recording of poor quality either contains weak signals or has noise that obscures the signals. The noise typically comes from the stethoscope contacting a patient and heavy breathing sounds obfuscating heart sounds.

We use intra-class correlation (ICC) [134] to measure the inter-rater reliability of the labels. We obtain the ICC estimates using a mean-rating, absolute-agreement, 2-way mixed-effects model, using the Pingouin package at <https://pingouin-stats.org/>. The calculated ICC estimates are 0.925 and 0.679, respectively for heart and lung, which correspond to excellent and moderate agreement [134]. The average variances for the 4 ratings of each stethoscope recording’s quality are 0.0163 and 0.0400 for heart and lung respectively. The max variances are 0.0938 and 0.151 for heart and lung. The average of the 4 ratings of each recording is used as the quality label. We further augment the dataset with 5 synthetic recordings of pure noise with varying amplitudes and labeled with quality 0, which emulate erroneous conditions like placing a stethoscope on irrelevant parts of the body, rubbing the stethoscope, bumping the stethoscope, and failing to make contact.

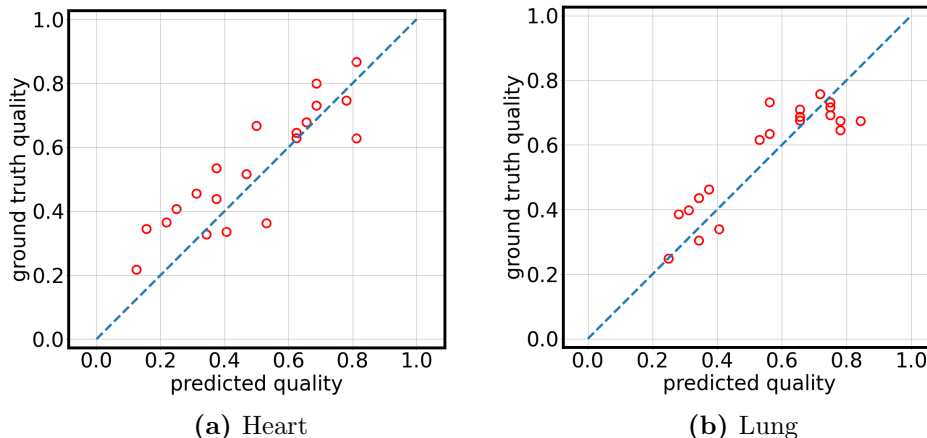


Figure 5.3: Trained heart and lung sound quality estimator predictions on the testing set, where the horizontal axis is the predictions and the vertical axis the ground truth labels.

Feature Extraction We follow the method of Grooby et al. for heart and lung quality classification and modify it for regression. For each sound recording, we apply noise reduction [135], band-pass filter with cutoff frequency 50-250 Hz for heart recordings and 200-1000 Hz for lung recordings, and extract for all sound recordings the top features listed by Grooby et al. In particular, we use the top 5 features for heart sound and the top 2-6 features for lung sound (top 1 feature for lung is not used due to the lack of open-source code). The extracted features are then subsequently used for training the estimators.

Learning We use the TPOT AutoML algorithm [136] to train the heart and lung sound quality estimators that take in the extracted features and predict the sound qualities. We split the dataset and use 25% as testing data. For the TPOT algorithm, we set generations = 100, population size = 100, and set the rest of the hyperparameters as default. We achieve a 0.0945 and 0.0733 mean absolute errors (MAE) on the testing set of heart and lung sounds respectively. The predictions on the testing set are shown in Fig. 5.3.

5.3.3 Visual Registration and Sound Quality Prior

Doctors are trained to auscultate typical points on a patient body as illustrated in Fig. 5.4, both for optimal sound quality and ability to diagnose abnormalities on specific structures of the heart and lung. The goal of visual registration is to locate these points on a patient, both to define regions for auscultation and to obtain a prior observation model. Note that we do not track the patient body after this initial registration is performed and assume that the patient body does not move significantly during auscultation. Small movements are accounted for by the motion controller (Sec. 5.3.4)

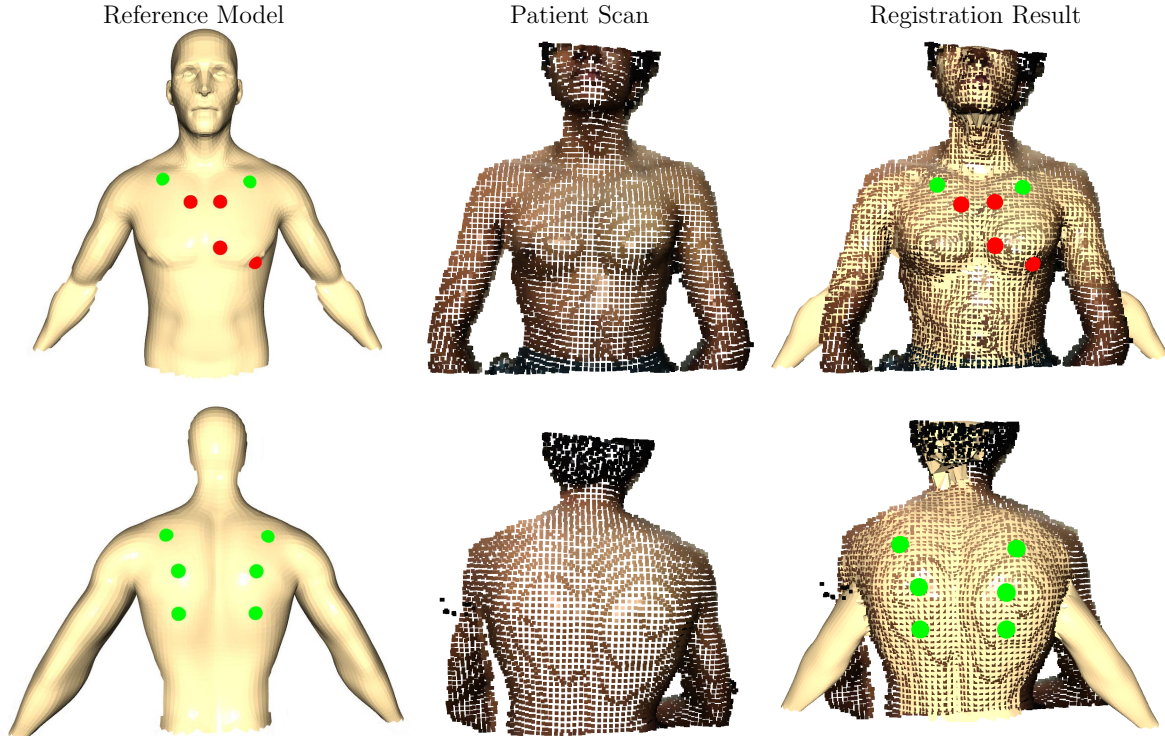


Figure 5.4: Left to right: labeled reference models, subject (“patient”) scan point clouds, and result of nonrigid registration. Auscultation locations for heart are labeled in red, and lungs in green. [Best Viewed in Color.]

Visual Registration We first manually label a reference human mesh model with auscultation locations, which we register onto a 3D point cloud of a patient captured with a RGB-D camera. In all of our experiments, we downsample the captured point cloud with a voxel size of 0.01 m, which strikes a balance between computational time and enough coverage by using a stethoscope whose diameter is approximately 0.02 m. We first perform rigid registration with RANSAC [137] followed by ICP [138] to provide a good initial alignment, then nonrigid registration with nonrigid ICP proposed by Amberg et al. [139], where a local affine deformation model is used. The initial and final results of one case of visual registration is shown in Fig. 5.4. The regions of interest on the patient surface are limited to a certain distance R from the estimated clinical auscultation locations (Fig. 5.5, green circles).

Point Cloud Projection and Sound Quality Prior We project the point cloud onto the coronal plane of a patient, and BO only searches amongst locations on this projected plane. The third dimension (height) of the point cloud is associated with each projected point to de-project selected locations on the plane back into 3D space. As discussed in Sec. 5.3.1, to obtain the prior of the observation model, we first generate a reference quality map $\mu_o(x)$, where we place a negative exponential function at each of the registered auscultation

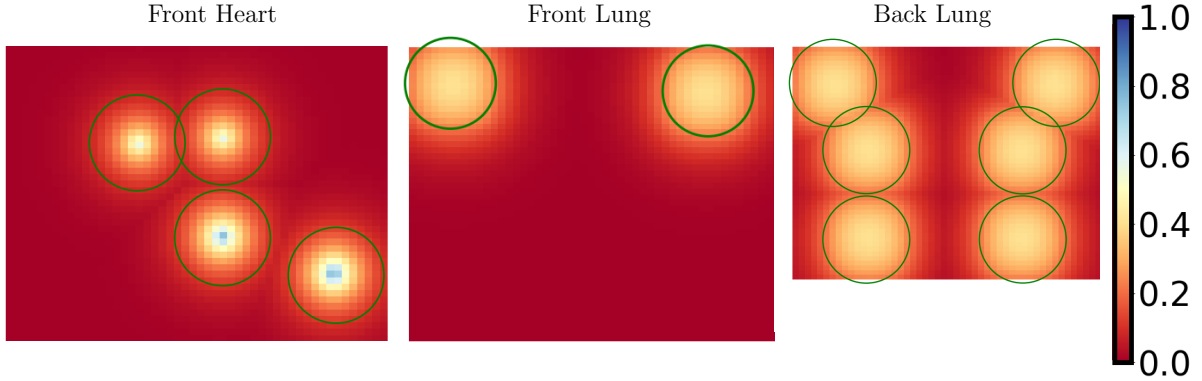


Figure 5.5: Observation model priors for heart and lung on the front of the patient and lung on the back. Each region A is denoted with green circles. [Best viewed in color.]

locations in the projected coronal plane. An example of the initial observation model prior is shown in Fig. 5.5. The latent parameters $\theta = [t_x, t_y, c]$ is initialized to $\theta_o = [0, 0, 1]$, where t_x and t_y are the translation offsets on the projected plane, and c the quality scaling term.

5.3.4 Motion Control

The motion control system generates safe interaction between the stethoscope and the patient, keeping the interaction force at a desired auscultation force F_{aus} . According to both Nowak et al. [140] and our observations in experiments, auscultation force F_{aus} has a small effect on sound qualities. We set $F_{aus} = 7N$, which is both comfortable for human subjects and provides good sound qualities in practice. The robot starts at a manually defined home configuration q_{home} . For a location x selected by BO, we first get the point p on the patient point cloud where x is projected from. To auscultate at p , the robot first moves the stethoscope to an initial position p_o that is a distance $d_o = 0.08$ m from p in the direction of outward surface normal \mathbf{n}_p at p to account for perception uncertainty, shown in Fig. 5.1. In particular, we set the target position of the stethoscope frame to be p_o , with the z -axis aligning with $-\mathbf{n}_p$ and solve for the target robot configuration q_0 with inverse kinematics (IK). The robot moves from q_{home} to q_0 linearly in the Cartesian space, following maximum linear and angular velocities $v_{max} = 0.065$ m/s and $\omega_{max} = 1.0$ rad/s. Starting from p_o , the robot moves the stethoscope in the $-\mathbf{n}_p$ direction at constant speed $s = 0.03$ m/s, during which we also adopt impedance control for enhanced safety, where we utilize the wrist-mounted force-torque sensor on the arm as wrench feedback and control the stethoscope as a mass-spring-damper. The robot keeps moving the stethoscope until the external force in

the \mathbf{n}_p direction reaches F_{aus} , at which point we record the stethoscope audio for 10s and use the learned estimators to estimate the sound quality. After auscultation is finished, the robot first moves back to q_o , waits for BO to give the next auscultation location, and moves to the next initial position with a linear Cartesian movement under velocity limits.

5.4 RESULTS

Throughout the experiments, we use the TRINA robot shown in Fig. 5.1. TRINA is a mobile bimanual manipulator with various visual sensors and swappable end-effectors. In this project, we use a Thinklabs digital stethoscope mounted on the left arm with firm foam as sound insulator. We use the Intel Realsense L515 RGB-D camera on TRINA to capture point clouds. For both the simulation and physical experiments, we use the squared exponential covariance function for GP with length scale = 0.02. We chose this length based on our observations of how sound qualities correlate across the surface of the body.

5.4.1 Simulation Experiment

First we compare the performance of different acquisition functions, and evaluate the efficacy of SPAR-GP. To do this, we compare the results of the BO algorithm under visual registration error and patient overall sound quality variations. We first generate a ground truth heart quality map across the patient chest in a similar fashion as the observation model generation in Sec. 5.3.3, with the same shape as Fig. 5.5 but having the negative exponential functions placed at ground truth locations. Then for each setting of acquisition function and prior observation model, we generate a prior by randomly shifting and scaling the ground truth quality map and run the BO algorithm. In particular, the random shift (x,y sampled independently) is uniform between -0.02m and 0.02m and the random scale is uniform between 0.7 and 1.3. We run this 50 times and compare the average maximum observed qualities across different regions under a given budget N_{max} . Simulations are performed for the heart prior only, giving a total of 4 regions. We disable early termination (i.e., Line 6 of Alg. 5.1) in these experiments.

We set the GP observation noise $\alpha = 0.0417$, and region radius $R = 0.03$ m. The covariance matrix Σ of the prior parameters θ is a diagonal matrix with entries $\sigma_{t_x}^2 = \sigma_{t_y}^2 = 1.33e-4$ (variance of the uniform random distribution of [-0.02,0.02]), and $\sigma_c^2 = 0.03$ (variance of the uniform random distribution of [0.7,1.3]).

Results are summarized in Table I, where we compare 1) acquisition functions EI and UCB with different β parameters under different budgets for each of the regions; 2) BO with

	$N_{max} = 3$			$N_{max} = 10$		
	Zero	Fixed	SPAR	Zero	Fixed	SPAR
EI	0.432	0.434	0.549	0.622	0.620	0.638
UCB, $\beta = 0.5$	0.370	0.428	0.550	0.629	0.624	0.660
UCB, $\beta = 1.0$	0.421	0.423	0.559	0.632	0.628	0.660
UCB, $\beta = 1.5$	0.455	0.420	0.544	0.631	0.624	0.657

Table 5.1: Effects of BO parameters on max sound quality in simulation

	Subject 1	Subject 2	Subject 3	Subject 4
Weight (kg)	95.6	57.4	60.5	120
Height (m)	1.84	1.82	1.81	1.83
BMI	28.2	17.3	18.5	35.8
Age	33	22	21	28

Table 5.2: Statistics of the 4 human subjects

GP under Zero prior, Residual-GP with Fixed prior, and SPAR-GP (SPAR) prior. Numbers indicate the maximum sound quality over each region. SPAR-GP consistently outperforms BO with fixed prior and no prior, particularly when the budget of observations is small. In addition, there is no clear winner among the acquisition functions, so for the subsequent physical experiments we use EI because it is parameter-free.

5.4.2 Physical Auscultation Experiments

We compare BO against two other baselines for a complete auscultation session that includes both heart and lung. In RO baseline, the robot automatically auscultate at the registered auscultation landmarks on the patient. In the DT baseline, a human tele-operates the robot to auscultate the patient, shown in Fig. 5.6. We perform each of the three methods once on 4 healthy male human subjects⁵, whose statistics are listed in Tab. 5.2.

Direct Teleoperation (DT) In the DT condition, an expert tele-operator performs auscultation at the specified regions, and adjusts the position of the stethoscope until he/she judges the sound quality to be good enough for making a diagnosis. The tele-operator is one of the medical school students that labeled the stethoscope recording dataset, and the tele-operator received tele-operation training on TRINA for 3 hrs prior to the experiments.

⁵This human subjects study was fully reviewed and approved by the University of Illinois at Urbana-Champaign Institutional Review Board, with IRB#21849.

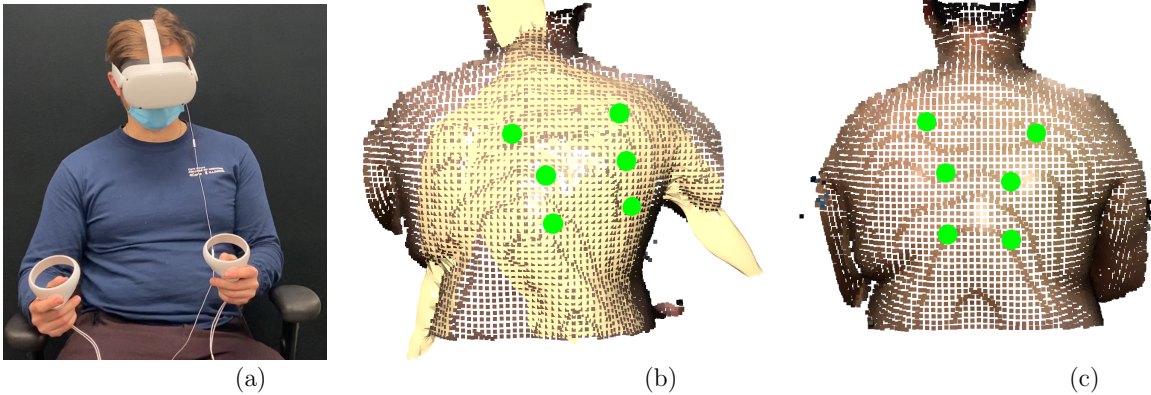


Figure 5.6: (a) Tele-operator performing auscultation using VR headset and controllers. (b) Automatic registration failed on the back of subject 4 due to difference in their body type from the reference model. (c) Manual registration on the back of subject 4.

Tele-operation on TRINA is achieved via the Oculus Quest VR headset and controllers (Fig. 7(a)), where the VR headset streams the stereo camera on TRINA’s head, and the 6D pose of the stethoscope on TRINA is driven in velocity control mode to follow the velocity of one of the VR controllers while a clutch button is depressed.

Bayesian Optimization (BO) For BO, we set the acquisition function to EI, $N_{max} = 4$, GP observation noise $\alpha = 0.0938$ for heart and 0.151 for lung. Σ is a diagonal matrix with entries $\sigma_{t_x}^2 = 2.12e-4$, $\sigma_{t_y}^2 = 9.77e-5$, and $\sigma_c^2 = 0.03$. In addition, We terminate early if sound quality exceeds 0.5. This threshold was judged by the 4 raters as the minimum quality for making a confident diagnosis.

The values for α correspond to the maximum inter-rater variance for each recording in the quality estimator dataset. The values for R and Σ were determined empirically as a function of estimated registration error. We estimate error using a public dataset of complete human scans [141]. We select 30 male human mesh models that cover a wide range of body types, with maximum, minimum, and average body mass index (BMI) 34.9, 17.8, and 26.5, respectively. We manually label the nipples, segment the mesh to emulate a RGB-D scan, and apply visual registration. The registration MAE error of the nipples is 0.0173 m, and the maximum error is 0.0389 m, which we set as R .

Results Tab. 5.3 breaks down the time spent on visual registration (Reg.), the total amount of time spent during the auscultation session (Total), the total number of auscultations (No.), the average of the maximum rated quality in each region (Avg. Max), and the minimum of the maximum rated quality across all regions (Min Max). Note that sound qualities are average ratings given by the the human raters, not values from the estimator.

One example of the final auscultated locations for the heart and lung, and the posterior sound quality estimates are shown in Fig. 5.7.

Subject	Method	Reg. (s)	Total (s)	No.	Avg.	Max	Min	Max
1	BO	171	538	22	0.727	0.438		
	RO	171	370	12	0.648	0.312		
	DT	0	656	19	0.651	0.281		
2	BO	183	496	18	0.716	0.531		
	RO	183	434	12	0.651	0.312		
	DT	0	559	14	0.804	0.531		
3	BO	177	1155	37	0.552	0.375		
	RO	177	442	12	0.484	0		
	DT	0	613	16	0.576	0.25		
4*	BO	148	694	21	0.635	0.469		
	RO	148	495	12	0.599	0.375		
	DT	0	468	14	0.711	0.625		

*: Manual registration was performed on the back of the subject.

Table 5.3: Experimental results on human subjects

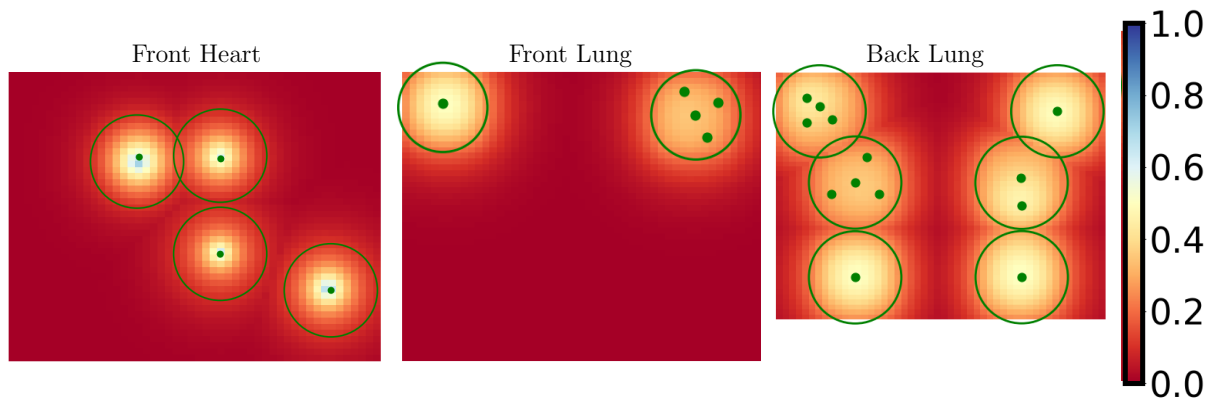


Figure 5.7: Posterior sound quality predictions for heart, front lung, and back lung on Subject 1. Observed locations are marked as green dots.

We use a linear mixed effects model to evaluate the methods, where we set the best estimated sound quality found in each region to be the dependent variable, methods to be fixed effects, and structure s (heart/lung), auscultation region, and subject to be random effects. The results are presented in Tab. 5.4. While the estimated fixed effects coefficients show separations among the methods, they are not strong compared to the confidence intervals.

Method	Fixed Effects		s	Variances of Random Effects		
	β	95% CI		Region	Subject	Residual
BO	0.662	[0.556,0.768]	3.40e-12	1.77e-3	6.03e-3	1.96e-2
RO	0.598	[0.492,0.704]				
DT	0.703	[0.597,0.859]				

Table 5.4: Mixed Effects Model Results (β :fixed effects coefficient, CI: Confidence Interval, s :anatomical structure)

Therefore, we perform a post-hoc pairwise testing using the Tukey’s test [142]. The p-values for the pairs BO-DT, BO-RO, and RO-DT are 0.326, 0.0700, and 0.00110, respectively. In addition, all three random effects have a small contribution to the overall variances, where the residual variance dominate.

To further understand diagnostic utility, we ask the raters to answer a series of yes-no questions regarding the recordings with the best estimated quality in each region. For heart sounds, the key features doctors look for are the first and second heart sound (S1 and S2), while for lung sounds, doctors look for inspiration and expiration. Therefore, for heart, $Q1$ and $Q2$ are whether the raters are able to hear S1 and S2, respectively. For lung, we ask whether they can hear inspiration ($Q1$) and expiration ($Q2$). Then for both heart and lung recordings, $Q3$ asks whether any sounds not originating from the target anatomy could be heard, and $Q4$ asks whether continuous noise or static obfuscated the heart/lung sounds. $Q5$ asks whether they can make a diagnosis based on the recording. The results are shown in Tab. 5.5.

Method	Heart					Lung				
	$Q1$	$Q2$	$Q3$	$Q4$	$Q5$	$Q1$	$Q2$	$Q3$	$Q4$	$Q5$
BO	100	98.4	12.5	26.6	87.5	93.0	86.7	9.38	23.4	74.2
RO	87.5	92.2	25.0	31.3	78.1	89.1	74.2	11.8	24.2	66.4
DT	90.6	95.3	32.8	46.9	85.9	96.9	89.8	6.25	31.3	85.2

Table 5.5: % of Yes responses for diagnostic utility of recordings Higher is better for $Q1$, $Q2$, and $Q5$

5.4.3 Discussion

Although direct teleoperation (DT) produces better sounds than registration-only automated auscultation (RO) ($p = 0.001$), there is no statistically significant difference between

DT and Bayesian optimization (BO) ($p = 0.326$). BO obtained higher minimum sound quality on patients 1 and 3 (Tab. 5.3), and better diagnostic values for heart sounds ($Q5$, Tab. 5.5). BO was also almost always able to generate S1 and S2 heart sounds ($Q1$ and $Q2$, Tab. 5.5). This could indicate greater consistency than a human operator, but more data is needed to test this hypothesis. The evidence suggests that BO achieves higher qualities than RO but is somewhat weak due to the small sample size ($p = 0.070$). BO does achieve a larger minimum quality and better diagnostic values from Tab. 5.3 and Tab. 5.5. As we expected, adaptive sensing helps overcome uncertainty in visual registration and anatomical differences. Overall diagnostic value ($Q5$, Tab. 5.5) roughly matches the results from the mixed effects model, where DT and BO perform comparable on heart sounds but DT outperforms BO on lung sounds, and both outperform RO.

Using BO, the robot takes less execution time than DT on subjects 1 and 2, and similar amount on subject 4. However, on subject 3, BO spent much more time auscultating, mainly due to the poor lung qualities observed on the patient. In fact, the robot ended up using the entire budget N_{max} on all lung locations. Future implementations could skip between regions, and return to past regions if there is sufficient time and need. Moreover, we limited the movement speed of the robot empirically to avoid intimidating the subjects, but further refinement of the motion controller could improve transit times.

For subject 4, the visual registration for the back exam was identified as having poor alignment, shown in Fig. 5.6. The tele-operator decided to reject the automatic registration, and instead manually labeled the auscultation points during the experiment, as shown in Figs. 7(b) and 7(c). This deviation from fully-automatic procedure is indicated by an asterisk in Tab. 5.3. The poor result from the visual registration is likely due to the difference between the body type of the subject (BMI = 35.8) and the reference human model. In future work, visual registration can be improved in various ways, such as adopting multiple reference models for different body types. We also note that raters noticed a moderate amount of noise and static ($Q4$, Tab. 5.5), and we assume this mainly comes from the robot motors because these sounds are not present during manual auscultation. DT was slightly higher than the automated methods, likely due to human tremor.

Finally, a surprising finding during these studies was that both the tele-operator and the raters detected an incidental cardiac pathology on subject 2. During DT, which were performed before BO and RO, the heart sound was described as a systolic III/IV high pitched crescendo-decrescendo murmur localized to the right-upper sternal border. Upon listening with BO and RO, the murmur was no longer present and the auscultation could be described as a normal S1 and S2 with physiologic S2 splitting distinctly noted. This murmur’s transient nature suggests a few possible diagnoses, including aortic stenosis, bicuspid aortic valve, or an

innocent murmur. Subject 2 was recommended to see a physician for further cardiac workup. This is promising since it shows that robotic auscultation can record cardiopulmonary sounds useful in diagnosis.

5.5 CONCLUSION AND FUTURE WORK

This chapter proposed a method that enables a robotic nursing assistant to automatically perform auscultation. To choose sensing locations, we show that using a Bayesian optimization that leverages visual prior information of the clinical auscultation locations outperforms choosing locations according to the prior only, and our system is capable of locating sounds of comparable quality with tele-operation by a human with clinical auscultation expertise.

While the results are promising, our experimental study only includes 4 young human subjects. In future work a larger-scale study is needed to evaluate its diagnostic capability on populations likely to exhibit abnormalities. Furthermore, the system is significantly slower than human doctors, and we intend to accelerate it by making movement faster, and reducing visual registration times, and compare it to manual auscultation. We would also like to improve the robustness of visual registration by using reference human models with varying body types. Finally, we would like to integrate our method with diagnostic algorithms to pre-screen for possible abnormalities.

CHAPTER 6: FEW-SHOT ADAPTATION FOR MANIPULATING GRANULAR MATERIALS AND LEARNING FORCE RESPONSE MODELS OF SOFT OBJECTS

As discussed so far in this thesis, while gray-box learning improves data efficiency, there are cases where the underlying physics is not well understood, making embedding analytical equations based on physics challenging. In addition, gray-box learning methods presented so far always learn from scratch. However, knowledge and priors of the task at hand can potentially improve data efficiency. This chapter explores the use of meta-learning to extract priors from offline data on similar tasks to accelerate online learning. In particular, this chapter proposes a vision-based few-shot meta-learning method with a deep kernel model trained with meta-learning to learn online from very limited experience on novel tasks. It introduces a novel meta-training approach, Deep Meta-Learning with Controlled Deployment Gaps (CoDeGa), that explicitly trains the deep kernel to make predictions robustly under large domain shifts. This chapter applies this method to two applications: few-shot scooping and gray-box learning of deformable object force response models, and shows that the proposed method outperforms non-adaptive methods as well as other state-of-the-art meta-learning methods.⁶

6.1 INTRODUCTION

Deploying robot systems in the real world often faces the challenge of deployment gaps, where the robot systems meet tasks out of the distribution of the training tasks. In such cases, models trained on the training tasks may start to degrade or even completely fail. As a result, it is necessary for the robot to adapt its strategy in an online fashion based on data acquired on the target task. Ideally, the robot should adapt with a small amount of data, i.e., in a few-shot learning setting. *Meta-learning*, also known as learning to learn, is a promising approach to few-shot adaptation in which the on-line learning strategy itself is learned using a set of training tasks [14]. This chapter proposes a vision-based few-shot meta-learning method that is trained to be adaptive and robust to large distribution shifts during deployment, by using a novel meta-training procedure that explicitly trains a deep Gaussian process (GP) model to make predictions robustly under large domain shifts. The deep Gaussian process model employs deep mean functions and deep kernels where the input to the GP kernel is transformed by a neural network. It's important to differentiate

⁶This chapter is adapted from Yifan Zhu*, Pranay Thangeda*, Melkior Ornik, and Kris Hauser, "Few-shot Adaptation for Manipulating Granular Materials Under Domain Shift". In *Robotics:Science and Systems (RSS)*, 2023.

our definition of the term deep GP from its other usage in the literature where it can also refer to deep belief networks constructed from compositions of GP models [143]. Although models similar to our proposed model have been adopted for few-shot adaptation [43], we introduce a new meta-training approach that explicitly trains the kernel to perform well on out-of-distribution tasks. Our Deep Meta-Learning with Controlled Deployment Gaps (CoDeGa) method repeatedly splits the training set into mean-training and kernel-training and learns kernel parameters to minimize loss over the residuals from the mean models. The key idea is that we explicitly control the splitting process to ensure the mean-training and kernel-training sets have maximal domain gaps, and the residuals seen by the kernels based on these splits are more representative of the residuals seen in an out-of-distribution target task. This chapter considers the application of our method to two problems.

The first application is motivated by autonomous lander missions on extraterrestrial bodies that need to sample granular material while coping with domain shift. The terrain compositions and topologies on extraterrestrial planets can be very complex, making it very challenging to embed physics priors. Therefore, this chapter proposes a vision-based scooping system, and employed in a Bayesian Optimization sequential decision-making framework, the proposed method allows the robot to use vision and very little on-line experience to achieve high-quality scooping actions on out-of-distribution terrains, significantly outperforming non-adaptive methods proposed in the excavation literature as well as other state-of-the-art meta-learning methods. Moreover, a dataset of 6,700 executed scoops collected on a diverse set of materials, terrain topography, and compositions is made available for future research in granular material manipulation and meta-learning.

The second application is acquiring force response models of novel heterogeneous deformable objects, which is first introduced in Chapter 4. While the method introduced in Chapter 4 does not leverage any prior knowledge of deformable objects, in this chapter, we apply our few-shot meta-learning method to extract useful priors from offline data on a variety of deformable objects. We evaluate our method on several deformable objects from novel categories and show that it significantly outperforms learning methods that learn from scratch, as is done in Chapter 4.

6.2 RELATED WORK

Here we review literature related to the learning techniques underlying our approach, including meta-learning for Gaussian processes, Bayesian optimization, and few-shot learning.

6.2.1 Meta-learning for Gaussian Process and Bayesian Optimization

When using a Gaussian process (GP) to model an unknown function, knowledge of the distribution from which this function is drawn is required, which is encoded in the GP kernel and other hyperparameters. Usually, this distribution is unknown in practice and all the hyperparameters are estimated with data. However, in the few-shot regime, this estimation can be quite inaccurate with few data points available [144]. One idea to approach this problem is to meta-train the GP on similar tasks [145, 146] offline to find good hyperparameters. In addition, to handle high-dimensional inputs, meta-training deep mean functions and/or deep kernels has also been explored [43, 147, 148].

Bayesian Optimization (BO) is a popular approach for sequential optimization in problems like ours that can be modeled as a contextual bandit. Using GP as the Bayesian statistical model for modeling the objective function is common practice in BO. Meta-learning GP in the context of BO has also been explored before, for both GP [149, 150] and deep mean and kernels [148]. Closest to our approach is the work that meta-learns deep kernels and means for use in BO [148]. Compared to this work, where there are dozens to hundreds of on-line samples, our work focuses on the few-shot regime. In addition, our work deals with real-world high-dimensional inputs and challenging testing scenarios that are drastically different from training scenarios.

6.2.2 Few-shot Learning via Meta-learning

Meta-learning and its application in the few-shot learning scheme have been explored extensively. It involves training models on an offline training dataset consisting of multiple tasks, that can adapt to a novel task using only a few examples. It has been studied in low-dimensional function regression [151, 152, 153], image classification [35, 154, 155, 156], and reinforcement learning tasks [157, 158, 159]. These approaches usually rely on some distance metric in feature space to compare the new examples to the available labeled examples [156, 160, 161, 162], perform few-shot estimation of the underlying density of data [163, 164, 165, 166], or use gradient descent to update a model learned on many related tasks [35, 157, 167, 168]. We apply few-shot learning in the context of deformable object force response model learning and decision-making for granular material manipulation. However, these meta-learning methods do not necessarily work well right out of the box for these problems especially with large domain shifts, which we will show in the experiments for granular material manipulation. Hence we propose CoDeGa, a novel few-shot meta-learning method for deep GP that allows a robot to achieve high-quality scooping actions in out-of-distribution terrains faster than state-of-the-art methods.

6.3 PROPOSED METHOD

We consider a model that maps observations and actions x to output y . During training, the robot has access to an *offline* prior experience, which consists of a set of M tasks $\{T_1, \dots, T_M\}$, and a training dataset $D_i = \{(x^j, y^j) | j = 1, \dots, N_i\}$ for each task $i = 1, \dots, M$. For convenience of notation, we separate the training datasets into sequences of dependent variables $D_i^y = \{(y_i^j) | j = 1, \dots, N_i\}$ and independent variables $D_i^x = \{(x_i^j) | j = 1, \dots, N_i\}$.

Our method proposes to use a deep kernel model with a deep mean, which is meta-trained by our Deep Meta-Learning with Controlled Deployment Gaps (CoDeGa) method to perform well under simulated deployment gaps extracted from the *offline* prior experience. We will first describe our proposed model, and then training the model with CoDeGa.

6.3.1 Deep Gaussian Process Model

A GP models function f as a collection of random variables $f(x)$ which are jointly Gaussian when evaluated at locations x [144]. A GP is fully specified by its mean function $m(\cdot)$ and kernel $k(\cdot, \cdot)$, which is the covariance function:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')). \quad (6.1)$$

Given n existing observed function values $\mathbf{y} = [y_1, \dots, y_n]^T$ at $\mathbf{x} = [x_1, \dots, x_n]^T$, GP regression predicts the function values at new point x^* as a Gaussian distribution:

$$P(y^* | \mathbf{x}, \mathbf{y}, x^*) \sim \mathcal{N}(m(x^*) + \mathbf{k}\mathbf{K}^{-1}\bar{\mathbf{y}}, k(x^*, x^*) - \mathbf{k}\mathbf{K}^{-1}\mathbf{k}^T). \quad (6.2)$$

Here,

$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix} + \sigma_n^2 \mathbf{I}, \quad (6.3)$$

$$\mathbf{k} = [k(x^*, x_1), \dots, k(x^*, x_n)], \quad (6.4)$$

$$\bar{\mathbf{y}} = [y_1 - m(x^*), \dots, y_n - m(x^*)]^T, \quad (6.5)$$

where σ_n is the standard deviation of noise at an observation and $\bar{\mathbf{y}}$ is the residual. A typical choice for the mean function is a constant mean and the radial basis function kernel (RBF) is a popular kernel of choice [144]. The mean constant, kernel function parameters, and σ_n can be hand-picked if there is prior knowledge of f . In practice, doing so is usually not

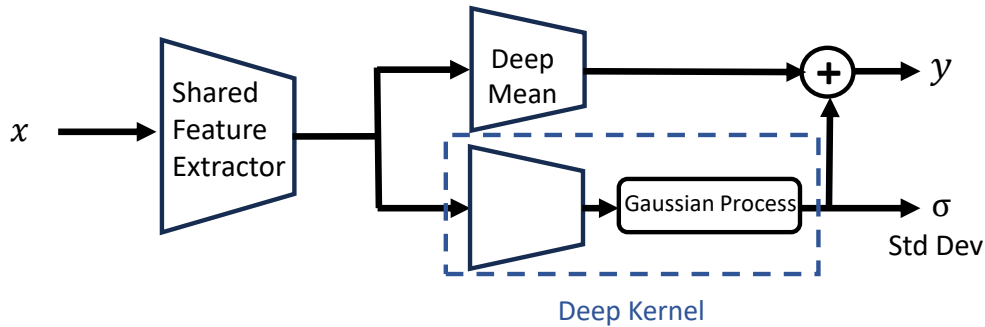


Figure 6.1: Model architecture. The deep kernel and the deep mean share a common feature extractor. σ is the standard deviation.

possible and they are estimated from data with type-II maximum likelihood by minimizing the negative log marginal likelihood (NLML):

$$\begin{aligned}
 -\log P(\mathbf{y}|\mathbf{x}, \theta) &= \frac{1}{2} \log |\mathbf{K} + \sigma_n^2 \mathbf{I}| \\
 &+ \frac{1}{2} (\mathbf{y} - m(\mathbf{x}))^T (\mathbf{K} + \sigma_n^2 \mathbf{I}) (\mathbf{y} - m(\mathbf{x})) \\
 &+ c,
 \end{aligned} \tag{6.6}$$

where θ denotes all the parameters to be determined and c is a constant.

Deep kernels enhance kernels with neural networks to be more scalable and expressive [42]. For deep kernels, an input vector is mapped to a latent vector using a neural network before going into the kernel function $k(g_\theta(\cdot), g_\theta(\cdot))$, where $g_\theta(\cdot)$ is a neural network with weights θ . Deep kernels for the few-shot adaptation setting have been proposed [43] previously. Deep kernels were also extended to use deep mean functions $m_\theta(\cdot)$ [147]. Our proposed deep GP contains both a deep kernel and a deep mean. The model architecture is shown Fig. 6.1. The kernel and mean function share the same feature extractor, which is different depending on the application’s observation and action input type.

6.3.2 Meta-learning with Controlled Deployment Gaps

The neural network parameters and the kernel parameters of a deep GP can be jointly trained over the entire training set with the same NLML loss as Eqn. 6.6, where θ contains the neural network parameters.

However, this approach does not typically train kernels that are well-tuned to individual tasks because it aggregates the data from all tasks together. Instead, meta-training may be realized with stochastic gradient descent with each batch containing the data for a single

task, i.e. minimizing the aggregate loss $\sum_i -\log P(D_i^y|D_i^x, \theta)$, where D_i^y and D_i^x are the target variables and input variables of a single task. We will refer to this approach as Deep Kernel and Mean Transfer (DKMT) which has been proposed in the meta-learning literature [147, 148].

DKMT has potential problems with out-of-distribution tasks. During training, the residuals seen by the kernels are residuals of the deep mean on the training tasks, which could be very different compared to the residuals on tasks out of distribution of the training tasks. As our experiments for scooping will show, this feature leads to the kernels being poorly calibrated in the worse case. Another potential issue is the over-fitting of the deep mean function. The first two terms of NLML in Eqn. 6.6 are often referred to as the *complexity penalty* and *data fit* terms, where the *complexity penalty* regularizes the deep kernels [144]. However, there is no regularization of the deep mean function. As a result, the deep mean can potentially overfit on all the training data, so the residuals will be close to zero.

CoDeGa addresses these issues by encouraging the residuals seen in kernel training to be representative of the residuals seen in out-of-distribution tasks with novel characteristics. In the application of granular media manipulation, these characteristics are the granular media material, and in the application of deformable object force response model learning, they are object categories. The idea is to separate the training tasks into a mean training set and a kernel training set, where each set contains different characteristics from one another. Then, the mean is trained on the mean training set to minimize error and the GP is trained on the residuals of the mean model on the kernel training set. These residuals are representative of those during deployment because they are characterized by a deployment gap of novel characteristics. However, this limits the amount of training data available for the mean and kernel. Therefore, we repeat this process similarly to k -fold cross-validation, in which each fold has a separate mean model trained on the mean split for that fold, and then the residuals for that model on the kernel split are used to define the kernel loss for that fold. A common kernel is trained using losses aggregated across folds. Finally, the mean model is trained again on all data.

The overall CoDeGa training procedure is as follows. Denote the set of all training characteristics as A and the set of training datasets as $D = \{D_1, \dots, D_M\}$.

1. Split A into K folds A_1, \dots, A_K .
2. Initialize an empty residual training dataset E .
3. For each fold A_k :

- (a) Split D into a kernel training set S_{kernel}^k and a mean training set S_{mean}^k . S_{kernel}^k

corresponds to tasks that contain any characteristic in A_k . S_{mean}^k consists of $D \setminus S_{mean}^k$.

- (b) Train feature extractor weights θ_f^k and deep mean weights θ_m^k using a mean squared error (MSE) loss on all the data in S_{mean}^k . Let these weights be denoted θ^k .
 - (c) For each dataset $D_i \in S_{kernel}^k$, collect the residuals of the mean model $\tilde{r}_i^j = r_i^j - m_{\theta^k}(x_i^j)$, $j = 1, \dots, N_i$. Construct the inputs $\hat{D}_i^x = D_i^x$ and the outputs $\hat{D}_i^y = \{\tilde{r}_i^j \mid j = 1, \dots, N_i\}$ to predict the residuals. $E \leftarrow E \cup \hat{D}_i$.
4. Train deep kernel parameters θ_k with database E , i.e. minimizing the aggregate NLL loss $-\sum_i \log P(\hat{D}_i^y | \hat{D}_i^x, \theta)$. We use stochastic gradient descent where each batch contains all samples in one task. (Note that the associated features extractor weights for each task need to be loaded and fixed at the start of each batch training.)
 5. Train θ_f and θ_m from scratch using standard supervised learning on all data in the offline database.

If there are known features of tasks that are correlated with the latent task variables α , instead of manually defining the characteristics for the splits, one approach would be to perform clustering in the feature space to identify large splits. We hope to explore these avenues in future work.

6.4 GRANULAR MEDIA MANIPULATION

Terrain sampling is a key component of scientific exploration of planets and other extraterrestrial bodies [169]. However, Earth-bound teleoperation, as typically done in existing landers, faces intermittent and delayed communication that incurs latency of minutes or even hours in the case of long-duration interference. Autonomous sampling, in which the robot interprets sensor signals and makes decisions on where and how to sample, could drastically increase the efficiency of exploration. However, realizing autonomous sampling is challenging due to uncertainty in terrain material properties, composition, appearance, and geometry, limits in onboard computation, and a limited sampling budget.

This section is inspired by proposed NASA missions to send autonomous landers to Europa and Enceladus to collect and analyze terrain samples to explore whether these bodies exhibit conditions that could support extraterrestrial life [169]. However, the composition of the icy regolith is largely unknown and could range from snow-like granules, to flat ice, to complex terrain formations. Although a lander may be tuned to sample well on Earth-bound terrain

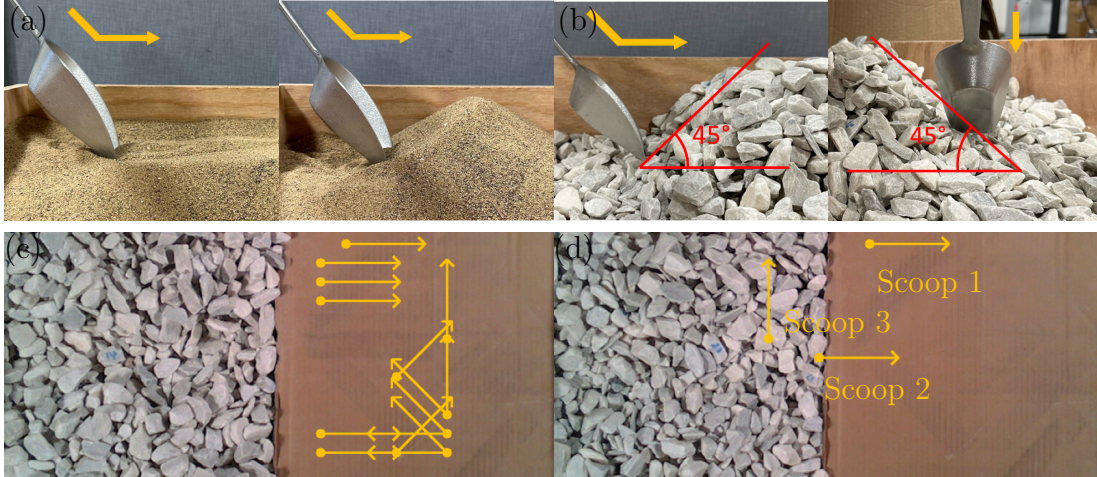


Figure 6.2: (a) Scooping sand towards a slope results in a larger scooped volume than scooping a flat surface, but (b) scooping gravel towards a slope (left) often results in jamming due to interlocking between particles. Instead, scooping perpendicularly to a slope (right) causes a large volume of rocks to fall into the scoop without excessive effort. (c) On a terrain with a novel non-scoopable material (right), a non-adaptive strategy predicts high volumes on this out-of-distribution material and only selects to scoop there (yellow arrows). (d) Our adaptive strategy adjusts based on the feedback to explore novel and uncertain regions [Best viewed in color].

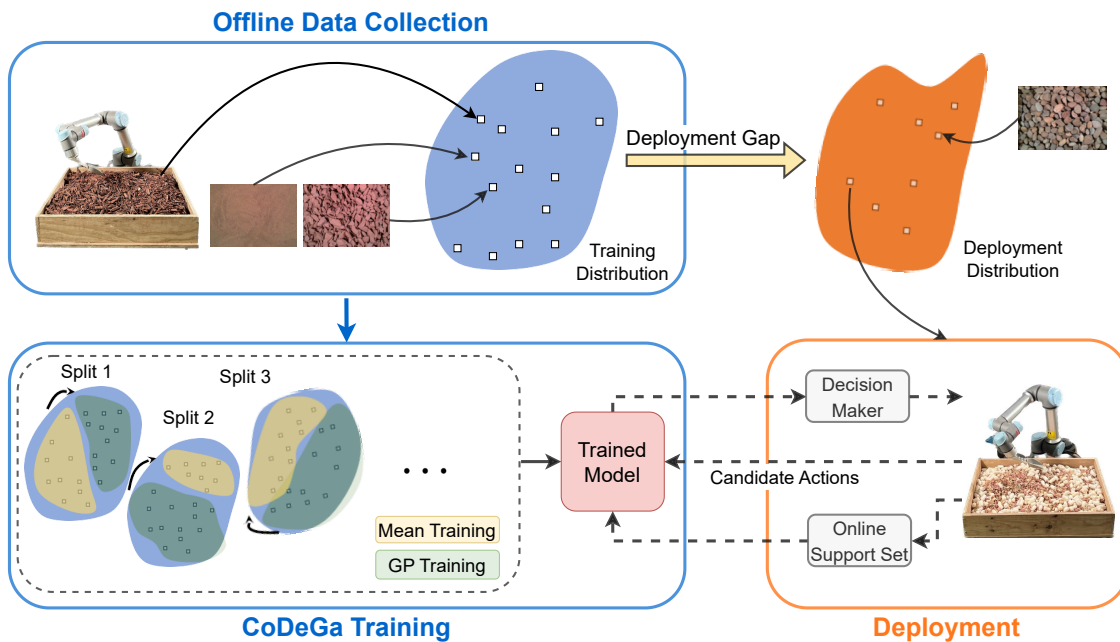


Figure 6.3: Method overview in the context of the scooping problem. Our proposed deep GP model is trained on the offline database with CoDeGa, which repeatedly splits the training set into mean-training and kernel-training and learns kernel parameters to minimize the residuals from the mean models. In deployment, the decision-maker uses the trained model and adapts it to the data acquired online (Support Set).

simulants, it will inevitably face a *deployment gap* when operating on an extraterrestrial body. This work proposes a learning-based approach that explicitly addresses such gaps.

Specifically, we study a simplified but analogous problem of scooping in which the goal is to collect high-volume samples from a target terrain with a limited budget of attempts. Different terrain compositions and shapes require very different scooping strategies, as shown in Fig. 6.2. For instance, while scooping against a slope on sand leads to a larger scooped volume, doing so on a bed of gravel often results in excessive contact forces because the gravel can interlock and jam. Although machine learning models can be trained using terrain simulants to predict scooping outcomes [170], these models will have worse prediction accuracy when the target terrain is *out of distribution* of the training terrains.

During scooping, our model takes an RGB-D image and parameters of a scooping action as input and predicts the mean and variance of the scooped volume. It is trained by executing 5,100 scoops on a variety of terrains with different compositions and materials. For decision-making, we use a Bayesian optimization framework that chooses an action that maximizes an acquisition function that balances the scoop volume prediction and its uncertainty. Our experiments evaluate the proposed method on out-of-distribution terrains that have drastically different appearances and/or material properties than the training terrains. Our method allows the robot to achieve high-volume scooping actions in out-of-distribution terrains faster than state-of-the-art deep kernel and conditional neural processes models when used in Bayesian Optimization. Moreover, it significantly outperforms non-adaptive methods such as those proposed in the granular material manipulation literature. To encourage further research in these domains, we release our dataset of 6,700 executed scoops collected on a diverse set of materials and compositions. ⁷.

6.4.1 Related Work

Our work is related to granular material manipulation and is to our knowledge the first approach to integrate vision input and manipulation outcomes to adapt to out-of-distribution terrains. Granular materials occur in a variety of real-world robotic applications, including food preparation, construction, and outdoor navigation. Many granular material manipulation tasks have been explored, including scooping [171], excavation [172], pushing [173], grasping [174], untangling [175] and locomotion [176, 177]. Most related to our work are scooping and excavation, which are connected but operate on different scales. The task proposed by Schenck et al. focuses on manipulating a granular terrain to a certain shape [171] by learning a predictive function of terrain shape change given an action. An optimization-

⁷Dataset download link: <https://drillaway.github.io/scooping-dataset.html>

based method is proposed by Yang et al. to generate excavation trajectories to excavate desired volumes of soil based on the intersection volume between the digging bucket swept volume and the terrain [178]. Dadhich et al. propose to use imitation-learning for rock excavation by wheel loaders, given expert demonstrations of [179]. All of these past works are developed on a single type of material. In contrast to these methods, our work directly addresses the large deployment gaps that are likely to be found in extraterrestrial terrain sampling. We explore the use of kernel learning methods trained on offline data combined with Bayesian Optimization to achieve adaptive behavior with little experience of the target terrain.

6.4.2 Problem Formulation

We formulate the scooping problem as a sequential decision-making task where the robot, in each *episode*, observes the terrain RGB-D image $o \in \mathcal{O}$, uses a *scooping policy* to apply $a \in \mathcal{A}(o)$ where $\mathcal{A}(o)$ is a discrete set of parameterized, observation-dependent scooping motions. The reward $r \in \mathcal{R}$ of a scoop is the scooped volume.

Presented with a target terrain T_* , the robot’s goal is to find a scoop whose reward is above a threshold B . In planetary missions, for example, B could be the minimal volume of materials needed to perform an analysis. During the n -th episode, the robot knows the history of scoops on this terrain $H = \{(o^j, a^j, r^j) \mid j = 1, \dots, n - 1\}$, which we also refer to as the *on-line support set*. Note that the support set only contains samples of low quality, i.e. below B , because otherwise the goal would already have been achieved.

The robot has access to an *offline* prior scooping experience, which consists of a set of M terrains $\{T_1, \dots, T_M\}$, and a training dataset $D_i = \{(o^j, a^j, r^j) \mid j = 1, \dots, N_i\}$ of past scoops and their rewards for each terrain $i = 1, \dots, M$.

For a terrain, we suppose a *latent variable* α characterizes its composition, material properties, and topography, which are only indirectly observed. Let α_* characterize T_* and α_i characterize T_i for $i = 1, \dots, M$. Moreover, the observation is dependent on the latent variable, and an action’s reward $r \equiv r(\alpha, a)$ is also an unknown function of the action and latent variable. Standard supervised learning applied to model $r \approx f(o, a)$ will work well when α_* is within the distribution of training terrains, and α_* is uniquely determined by the observation o or the reward is not strongly related to unobservable latent characteristics. However, when T_* is out of distribution or the observation o leaves ambiguity about latent aspects of the terrain that affect the reward, the performance of the learned model will degrade.

As a result, on-line learning from H has the potential to help the robot perform better on T_* . Meta-learning attempts to model the dependence of the reward or optimal policy on α ,

either with explicit representations of α (e.g., conditional neural processes [180]) or implicit ones (e.g., kernel methods [43], which are used here).

6.4.3 Scooping Problem Description

This section describes the scooping problem in more detail. We use the setup shown in Fig. 6.4, which includes a UR5e arm with a scoop mounted on the end-effector, an overhead Intel RealSense L515 RGB-D camera, and a scooping tray that is approximately 0.9 m x 0.6 m x 0.2 m. A terrain is defined as a unique *composition* of one or more *materials*, where a material is composed of particles with consistent geometry and physical properties. We consider a variety of materials and compositions for the offline database. The materials used in this work are listed in Tab. 6.1. The offline database contains materials Sand, Pebbles, Slates, Gravel, Paper Balls, Corn, Shredded Cardboard, and Mulch. The testing terrains also include Rock, Packing Peanuts, Cardboard Sheet and Bedding, which significantly differ from the offline materials in terms of appearance, geometry, density, and surface properties. The terrain compositions used are listed in Tab. 6.2. The offline database contains the Single, Mixture, and Partition compositions, while the testing set also contains the Layers composition. On terrains with the Layers composition, observations do not directly reflect the composition of the terrain, and on-line experience is needed to infer it. All terrains are constructed manually, with varying surface features (e.g. slopes, ridges, etc.) with a maximum elevation of about 0.2 m and a maximum slope of 30°. Some terrain examples are demonstrated in Fig. 6.5. We also observe that the scooping outcomes show high variance because many terrain properties are not directly observable, such as the arrangement and geometry of the particles beneath the surface.

A *scoop action* is a parameterized trajectory for a scoop end effector that is tracked by an impedance controller. We follow the common practice in the excavation literature [170, 181] to define a scooping trajectory, shown in Fig. 6.6, where the scoop has a roll angle of 0 and stays in a plane throughout the trajectory. The scoop starts the trajectory at a location p , penetrates the substrate at the attack angle α to a penetration depth of d , drags the scoop in a straight line for length l to collect material, closes the scoop to an angle β , and lifts the scoop with a lifting height h . We assume that the scoop always starts scooping at the terrain surface, which can be calculated from the depth image. The impedance controller of the end-effector is configured with stiffness parameters b .

To reduce the action space, we manually tuned the parameters that have smaller effects on the scooping outcome, fixing the attacking angle α at 135°, the dragging length l at 0.06 m, the closing angle β at 190°, and the lifting height h at 0.02 m. In addition, we set



Figure 6.4: The experimental setup.



Figure 6.5: Example terrains illustrating different compositions, materials, and topography. Top left: Partition of Sand and Corn. Top right: Mixture of Mulch and Gravel. Bottom left: Single of Pebbles. Bottom right: Layer of Shredded Cardboard (right), Packing Peanuts (left, top layer), and Slates (left, bottom layer).

Sand [†] : fine play sand, $\ll 1$ mm		Pebbles [†] : rocks, 0.8 – 1.0 cm	
Slate [†] : flat rocks, 2.0–4.0 cm		Gravel [†] : rocks, 1.5–3.0 cm	
Paper Balls [†] : crumpled paper, 4.0 – 6.0 cm		Corn [†] : dry corn kernels, 0.3–0.7 cm	
Shredded Cardboard [†] : cardboard, 1.0 – 8.0 cm		Mulch [†] : red wood landscape mulch	
Rock : rocks, 5.0 – 8.0 cm		Packing Peanuts : white packing peanuts, 2 x 4 cm	
Cardboard Sheet : flat cardboard sheet		Bedding : small animal bedding, wood shavings, 0.2 - 3 cm	

Table 6.1: Materials used in each terrain, with approximate grain sizes where applicable. Images show U.S. quarter coin for scale. Training set denoted with [†]; testing set consists of all materials and novel material combinations.

two options for the impedance controller stiffness b , corresponding to soft and hard stiffness, where the linear spring constants are 250 N/m and 750 N/m and the torsion spring constants are 6 Nm/rad and 20 Nm/rad, respectively. Therefore, the action is specified by the starting x , y position and yaw angle of the scoop, the scooping depth d , and stiffness b .

To measure the scooped volume, the scoop is moved to a fixed known pose, after which a height map within the perimeter of the scoop is obtained from the depth image. The volume is then calculated by integrating the difference between this height map and the height map

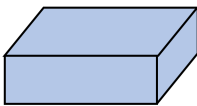
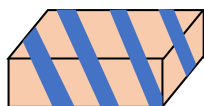
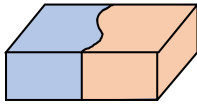
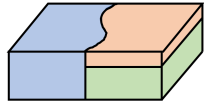
Single[†] : A single material		Mixture[†] : Uniform mixture of two materials	
Partition[†] : two materials that each occupy a partition		Layers : Two partitions but with two layers of different materials in one partition	

Table 6.2: Terrain compositions. Training set denoted with [†]; testing set consists of all compositions.

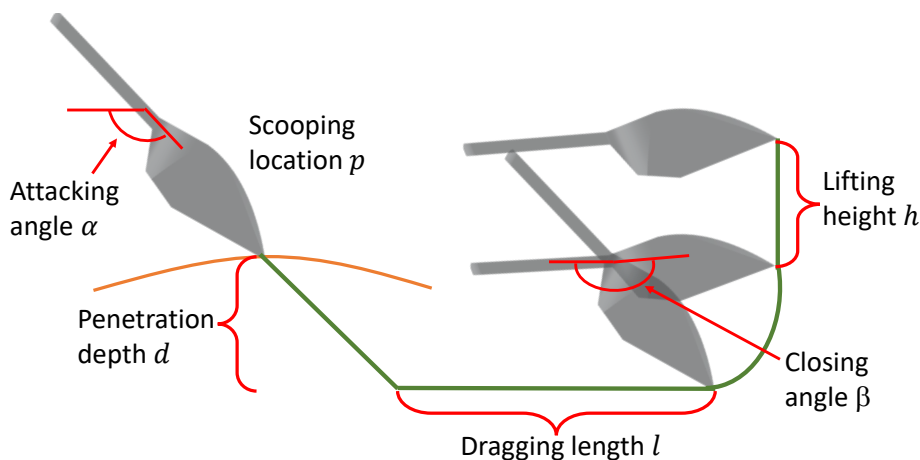


Figure 6.6: Parametrized reference scooping trajectory.

of an empty scoop at the same pose collected beforehand.

The offline database contains data on 51 terrains, all with unique combinations of materials and compositions. Out of these terrains, 8 are Single, 25 are Partition, and 18 are Mixture. The materials used are randomly selected from the training materials. 100 random scoops are collected on each terrain, sampled uniformly with random x , y positions in the terrain tray, random yaw angle from a set of 8 discretized yaw angles, 45° apart, random depth in the range of 0.03 m to 0.08 m, and random stiffness (either “hard” or “soft”). Sometimes trajectory planning of the robot manipulator for a sampled scoop can fail due to kinematic constraints. If so, the scoop action is discarded and sampling continues until planning is successful. The average scooped volume across the offline database is 31.3 cm^3 , and the maximum volume is 260.8 cm^3 . The distribution of scooped volumes in the database is shown in Fig. 6.7.

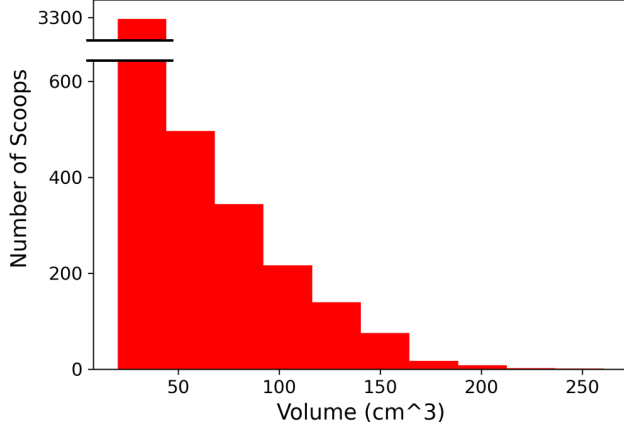


Figure 6.7: Collected data distribution. The maximum scoop volume is 260.8 cm^3 , while the average is 31.3 cm^3 .

6.4.4 Method

For the tasks of scooping, the specific architecture of the deep GP model is shown in Fig. 6.8. It is then trained by the CoDeGa method, introduced in the previous section, where it splits the terrains based on the terrain materials, regardless of the composition.

To use a reward model in the scooping sequential decision-making problem, the decision-maker maximizes a score $s(o, a)$ over the action a : $\pi(o) = \arg \max_{\mathcal{A}(o)} s(o, a)$. A greedy optimizer would use the mean as the score, $s(o, a) = m(o, a, H)$ where $m(o, a, H) = E[r|r \sim p(R|o, a, H)]$, but this does not adequately explore actions for which the prediction is uncertain. Instead, a Bayesian optimizer uses an acquisition function that also takes uncertainty into account. For example, the upper confidence bound (UCB) method defines the acquisition function $s_{UCB}(o, a) = m(o, a, H) + \gamma \cdot \sigma(o, a, H)$, where $\sigma(o, a, H) = \text{Var}[r|r \sim p(R|o, a, H)]^{1/2}$ is the standard deviation of the prediction and $\gamma > 0$ is a parameter that encourages the agent to explore actions whose outcomes are more uncertain.

6.4.5 Experiments and Results

Testing Tasks We evaluate our method on 16 test terrains that contain out-of-distribution materials and compositions. We introduce 4 new materials, which are Rock, Packing Peanuts, Cardboard Sheet, and Bedding, described in Tab. 6.1. In addition to the compositions during training, we also consider the Layers composition, described in Tab. 6.2. Note that the Cardboard Sheet material is not scoopable. For each of the Single, Partition, Mixture, and Layers compositions, we consider 4 terrains. The 4 Single terrains are created with each of the 4 new testing materials. Material combinations on terrains with the Mixture, Partition, and Layers compositions are randomly generated but with the constraints that 1) each of

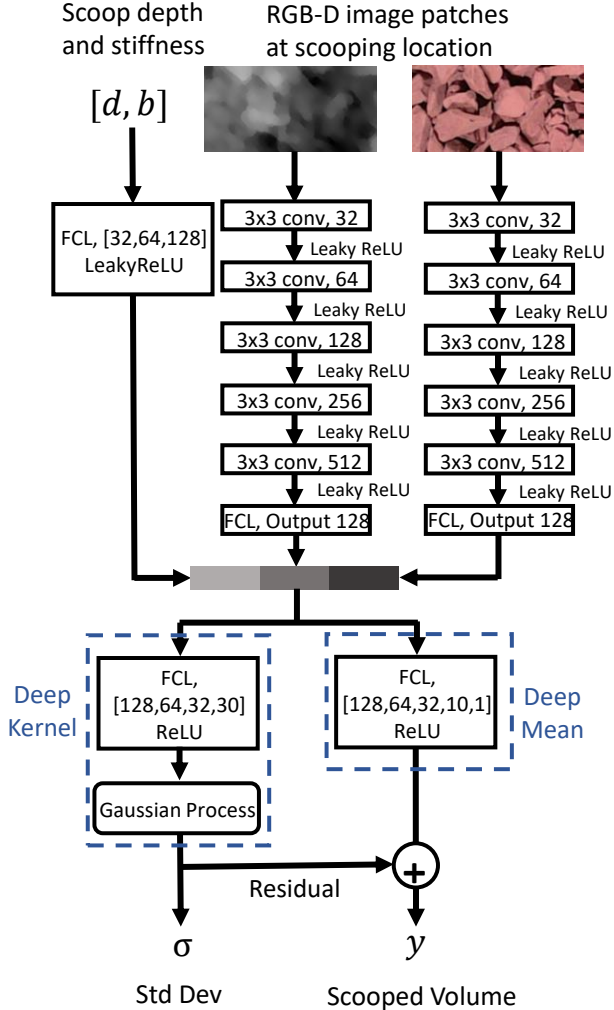


Figure 6.8: Model architecture. The deep kernel and the deep mean share a common feature extractor. “FCL” denotes fully-connected layers, with the output layer sizes and the activation function specified. “conv” denotes convolution layers with filter size and number specified. A stride of 2 is used for all convolutional filters. The radial basis function kernel is used for GP.

the new materials is selected at least once; 2) each terrain contains at least 1 new material. We exclude Cardboard Sheet from Mixture since it is physically impossible to create.

Model Training PyTorch [182] and GPyTorch [183] are used to implement the neural networks and GP. The Adam optimizer is used for training. Learning rates of $5e-3$ and $1e-2$ are used for the training of the deep mean and deep kernel, respectively. These values are hand-picked by inspecting the training loss and are not carefully tuned. For training the mean, 10% of the training data is used for validation and early stopping based on the validation loss with patience of 5 is used to select the training epochs. Early stopping with a patience of 5 based on the training loss is used to select the training epochs for the deep kernel. We also apply data augmentation, where for both mean and kernel training, random

vertical flips of the images are used since flipping vertically would not change the predicted volume. For mean training, random hue jitter and random depth noise are also applied. The training process takes less than 30 minutes on a hardware setup consisting of an i7-9800x CPU, a 2080Ti GPU, and 64GB of RAM.

Simulated Experiments We perform 2 types of simulated experiments, *prediction accuracy* and *simulated deployment*, on a static test database to evaluate the performance of our methods against the state-of-the-art. The test database consists of 100 randomly chosen scoops on each of the 16 testing terrains

For prediction accuracy, we are evaluating how well each model predicts scoop volume in the k -shot setting. 80 samples from each terrain’s data are first randomly drawn to form the query set. The support set is randomly drawn from the remaining 20 samples. The prediction accuracy in terms of mean absolute error (MAE) of the model on the query set given the support set is evaluated.

For simulated deployment, we evaluate how the model’s prediction accuracy impacts adaptive decision-making performance. In this experiment, we implement a policy that only selects from the 100 actions in the dataset for the given test terrain, and the robot receives the corresponding reward observed in the dataset. A trial begins by observing a single RGB-D image as input, and the agent executes the policy until the sample reward is above a threshold B . B is customized for a given terrain and is defined as the 5th largest reward in that terrain’s dataset. The Single Cardboard Sheet terrain is excluded in these experiments because it is not scoopable.

We compare our method against two state-of-the-art meta-learning methods and one non-adaptive baseline. The first method is DKMT [147], with the same deep network architecture. A learning rate of 1e-2 and early stopping with patience of 5 based on the training loss is used to select the epochs. The second method is conditional neural processes (CNP) [180], which is a non-kernel-based approach that learns a task representation using the support set and conditions the prediction on the query set on the learned task representation. Neural networks of similar width and depth compared to our proposed neural network architecture are used. During each epoch in meta-training, each task is randomly split into a support set of 5 samples and a query set of 95 samples. The training loss is the NLML on the query set on each task given the support set. Early stopping with a patience of 5 based on the validation loss on 6 randomly sampled validation tasks is used to select the training epoch. For the 0-shot case in CNP, a zero task representation is used. The last baseline is a non-adaptive baseline that only uses the deep mean function of our model.

The results are summarized in Tab. 6.3 and Tab. 6.4. Each model is trained 3 times with

Method	0-shot	5-shot	10-shot	0-shot*	5-shot*	10-shot*
CoDeGa	27.4	24.7	23.8	68.4	61.3	60.8
DKMT [147]	25.8	22.1	21.3	103.4	83.6	80.1
CNP [180]	25.7	25.1	25.0	101.4	100.4	100.5
Non-adaptive	27.4	27.4	27.4	68.4	68.4	68.4

Table 6.3: Average Prediction MAE. Averages taken across all test terrains and across 3 random seeds. * indicates MAE over the 5 samples of largest volume in the query set, averaged across all test terrains.

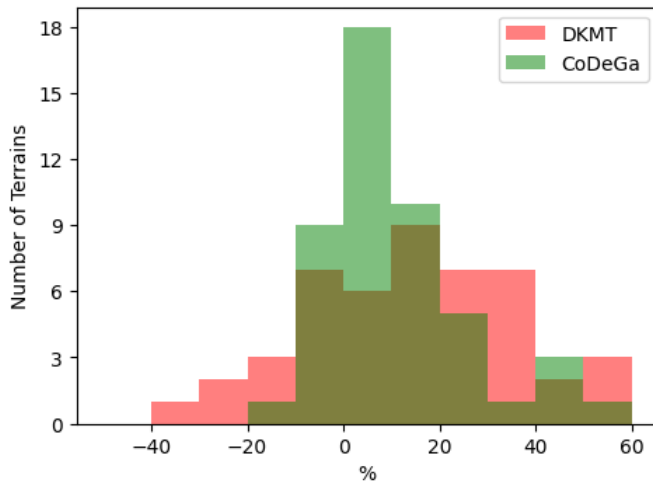


Figure 6.9: Distribution of MAE reduction from 0-shot to 10-shot support sets (10% indicates that the 10-shot MAE is 90% of the 0-shot MAE), over all testing terrains, and aggregated over 3 trained models.

different random seeds and average results across all tasks aggregated over 3 random seeds are reported. For prediction accuracy, in addition to the average MAE on all query data, we also report the average MAE of the 5 largest samples in the query set of each terrain because it is important to predict well on the “good samples” in a reward-maximizing decision-making setting. To compare DKMT and CoDeGa on a finer scale, we plot the distribution of MAE percentage reduction of each terrain from 0-shot to 10-shot support sets on all test terrains, aggregated over all 3 trained models, in Fig. 6.9. For simulated deployment, the adaptive methods are used by a UCB decision maker with $\gamma = 2$, while the non-adaptive baseline sorts actions by the reward predicted by the mean model and greedily proceeds down the list.

From Tab. 6.3, we observe that DKMT and CoDeGa have similar MAE reduction from 0-shot to 10-shot on average, with DKMT slightly outperforming it. However, CoDeGa outperforms DKMT in terms of prediction accuracy on the good samples. In addition, from Fig. 6.9, we find that DKMT exhibits a high variance, even *degrading* significantly in performance for some terrains from 0-shot to 10-shot adaptation. CoDeGa outperforms

Method	Avg. Attempts	Max. Attempts
CoDeGa	5.2	28
DKMT [147]	6.9	50
CNP [180]	9.6	40
Non-adaptive	8.3	57

Table 6.4: Simulated deployment results. Average and Max are taken across all testing terrains excluding the Single Cardboard Sheet terrain, and across 3 random seeds.

DKMT on the simulated deployment experiments. On the Single Rocks testing terrain where DKMT suffers the largest degradation, BO with the DKMT model takes as many as 44 attempts to reach the threshold in for one of the random seeds. This is due to incorrect correlations between low-quality support set samples and samples that are potentially of high quality. Both CoDeGa and DKMT outperform CNP and the non-adaptive approach significantly on both prediction accuracy and simulated deployment.

Physical experiments Lastly, we evaluate the real-world performance of our method in physical deployments. Here, the robot has a larger action set, it executes the scooping sequence as determined by the optimizer, and each action introduces terrain shifting for the subsequent action, so the RGB-D image is re-captured after every scoop. Policies are deployed on the same 15 testing terrains as the simulated experiments and with the same termination threshold B . For each trial a budget of 20 attempts is enforced, beyond which the trial is considered a failure.

The action set is a uniform grid over the action parameters, with 15 x positions (3 cm grid size), 12 y positions (2 cm grid size), 8 yaw angles, 4 scooping depths, and 2 stiffness, totaling 11520 actions. If robot trajectory planning fails for a scooping action, the next action that has the highest score is selected until planning succeeds.

We compare our proposed method (Ours) to the Non-Adaptive baseline, i.e. only the deep mean, and a volume-maximizing (Vol-Max) policy, where the action is chosen to maximize the intersection between the scoop’s swept volume and the terrain following a strategy proposed recently in the excavation literature [178]. We note that Vol-Max also does not adapt. Ours uses a UCB decision maker with $\gamma = 2$ and the CoDeGa model, while Vol-Max and Non-Adaptive use a greedy decision maker.

Each method is run on each terrain three times. Ours and Non-Adaptive are tested with three models trained with different random seeds, while Vol-Max is simply tested 3 times. When deploying the policies on a testing terrain, the terrain is manually reset at the start of each deployment so that surface features are consistent across trials. Note that slight terrain

Method	Avg. Attempts	Max. Attempts	Success Rate
Ours	3.1	16	100%
Vol-Max	7.3	20	91.1%
Non-adaptive	6.2	20	84.4%

Table 6.5: Results from physical experiments over 3 trials per method on 15 testing terrains. The robot is limited to 20 attempts.

variations are introduced naturally during the reset.

The average and max number of attempts before termination and success rates for all methods are reported in Tab. 6.5. Our method outperforms the other two baselines significantly, achieving a 100% success rate. We show two representative trials for each of the three methods in Fig. 6.10. On the Partition with Gravel and Cardboard Sheet terrain, the deep mean function predicts higher volumes on the Cardboard Sheet, but our method is able to quickly adapt. Vol-Max also succeeds, but requires more attempts since maximizing volume is not the most optimal policy to scoop Gravel. Non-adaptive causes the decision-maker to repeatedly select scoops on the Cardboard Sheet and eventually fail. On the second terrain, Layers with Packing Peanuts over Slates and Shredded Cardboard, the deep mean function predicts higher volumes on Packing Peanuts, but due to the layer of Slates underneath the scoop jams easily. Our method adapts to this observation in a few attempts. Vol-Max is able to perform similarly well as our method because Shredded Cardboard has more prominent terrain features, resulting in large intersection volumes. As a result, it always selects to scoop on Shredded Cardboard but takes 9 attempts to obtain high volumes because Vol-Max ignores the arrangement of granular particles, which has a big effect on the scoop outcome. Non-Adaptive takes many samples on Packing Peanuts, but eventually stops because the Slates become exposed, and Slates are in the training database and predicted to yield low volume.

Although our method achieves a 100% success rate, it is possible to have arbitrary materials, e.g. material with rainbow colors, where the kernels correlate support data samples poorly. In such cases, it is desirable to adapt the kernels online and we leave this to future work.

6.5 DEFORMABLE OBJECT FORCE RESPONSE MODEL LEARNING

In this section, we revisit the problem of learning point interaction models of deformable objects, first introduced in Chapter 4. Previously, learning the point interaction model is done purely in an online fashion with AutoML [184] without any prior knowledge of the

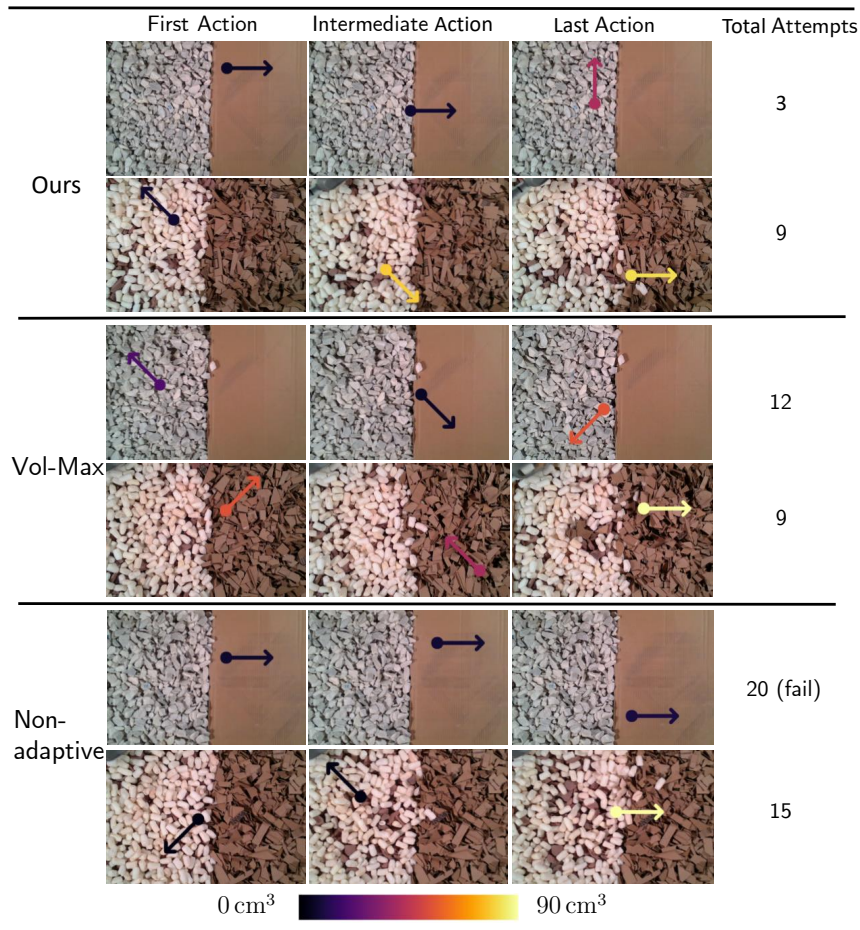


Figure 6.10: Example physical trials on two terrains comparing our method, the Non-Adaptive baseline, and Vol-Max. The terrain on the top is Partition with Gravel and Cardboard Sheet, while that on the bottom is Layers with Packing Peanuts over Slates on the left and Shredded Cardboard on the right. The arrow indicates executed action while the color indicates the scooped volume.

learning problem. In this section, we demonstrate that by leveraging offline training data on a variety of deformable objects using CoDeGa, online learning adapts to out-of-distribution deformable objects better.

6.5.1 Problem Description

The problem setup follows that introduced in Chapter 4.3.1. The main difference is that now the machine learning model has access to prior poking experience on different objects. We briefly describe the problem here. Data collection is performed with the same procedure as that introduced in Chapter 4.3.1. For each deformable object, the equilibrium surface model is captured, which is a set of S of points $x \in \mathbb{R}^9$, with each point

$x = [p_x, p_y, p_z, r, g, b, n_x, n_y, n_z]$, representing position, color, and (outward) normal, respectively. The point interaction model is denoted as $f = y(x, d)$, where $f \in \mathbb{R}$ is force magnitude in the normal direction, x is a point in S , and $d \in \mathbb{R}$ is displacement. Letting $\mathbf{n} = [n_x, n_y, n_z] \in \mathbb{R}^3$ denote the normal in x and $\mathbf{p} = [p_x, p_y, p_z] \in \mathbb{R}^3$ the position, this map gives the force a point probe would feel in the direction of \mathbf{n} when the point is moved to position $\mathbf{p} - d \cdot \mathbf{n}$.

Poking data is then collected at sampled surface points by moving the point probe in the negative normal position at each point while recording force in the normal direction at 200 Hz until a maximum force is reached, which is set to be 7 - 11 N, depending on the overall stiffness of each object (please refer to Chapter 4.3.1 for details). Sampling is done by using voxel downsampling with a voxel size of 1.5 cm to 3 cm, depending on the overall size of the object.

Consequently, the dataset of each object includes the surface point cloud S , a set of N points poked: $X = \{x_1, \dots, x_N\}$, and for each point poked, 8 uniformly sampled displacement-force pairs (d, f) . The objects in the offline training set and online testing set are shown in Fig 6.11, with a total of 21 objects coming from 7 distinct categories with varying visual and tactile properties.

In addition, we also collect data of the objects interacting with other novel rigid objects for evaluating the accuracy of the simulator. In particular, we use the same line probe as shown in Fig. 4.2a in Chapter 4. We probe with the line probe once at each of the same poking locations as that for the point probe with random orientation (about the normal) while recording the 6D wrench until the force magnitude reaches the maximum force, being 11 - 15 N depending on the object. Similar to the point probing data, the displacement-wrench data is also uniformly downsampled to 8 points for each probing action.

6.5.2 Experiments

We use the AutoML [184] method used in Chapter 4 as the baseline. For the deep kernel model, we adopt a very similar architecture compared to that used for scooping. The main difference is that instead of using CNN to encode local RGBD images, we use a PointNet++ [185] for encoding the entire object point cloud, with default settings. The input point cloud S is randomly downsampled to 2048 points, and each point only includes the position and color information. The action input for the model is the position of the probed point and the displacement, and the output is the mean and standard deviation of the felt force in the normal direction.

We evaluate the performance of CoDeGa with force prediction mean absolute error (MAE)



Figure 6.11: The training (blue box) and testing (orange box) objects used in the point interaction model learning experiment with CoDeGa. The boxes group objects from the same category together. The categories include stuffed objects, toys with rubber, dog toys, hand bags, sports equipment, shoes, and boxes (order from left to right and top to bottom). [Best viewed in color.]

in the k -shot setting on the testing objects, after training on all training objects. The controlled deployment gap used here is the object category.

Results We use the same hyperparameters for training the deep kernel as those for scooping. CoDeGa is trained 3 times with different random seeds. The mean absolute error (MAE) for different numbers of shots for CoDeGa and AutoML is shown in Fig 6.12. The MAE reported for AutoML is the average across 4 testing objects while that reported for CoDeGa is averaged across the aggregated tasks over 3 random seeds. The standard deviation is calculated from the 3 average MAE of 4 testing objects from 3 random seeds. CoDeGa is able to outperform AutoML significantly, especially when the number of shots is small. A qualitative comparison of CoDeGa and AutoML predictions on a tissue box is shown in Fig. 6.13. From the ground truth values, it is obvious that there is some calibration error in the RGBD camera extrinsics. Our method is able to better capture both the bias in the camera calibration and the fact that the edges and corners of the tissue box are stiffer.

We further combine the learned point model with the analytical simulator for simulating the contact wrench of the testing objects with a line probe. The force and torque error magnitudes using the point models trained by CoDeGa and AutoML with different numbers of shots averaged across the 4 testing objects are shown in Fig. 6.14. A medium-performing random seed is used for CoDeGa. These results show similar patterns to the results for point model learning, where CoDeGa also improves the analytical simulator over AutoML. These results also show that CoDeGa can complement gray-box learning.

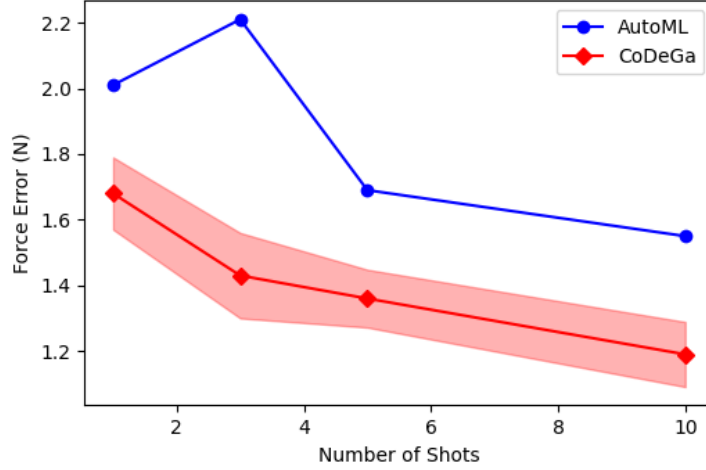


Figure 6.12: The point interaction model learning results. The force MAE reported for AutoML is the average across 4 testing objects while that reported for CoDeGa is averaged across the aggregated tasks over 3 random seeds. The shaded area represents the standard deviation, which is calculated from the 3 average MAE of 4 testing objects from 3 random seeds. [Best viewed in color.]

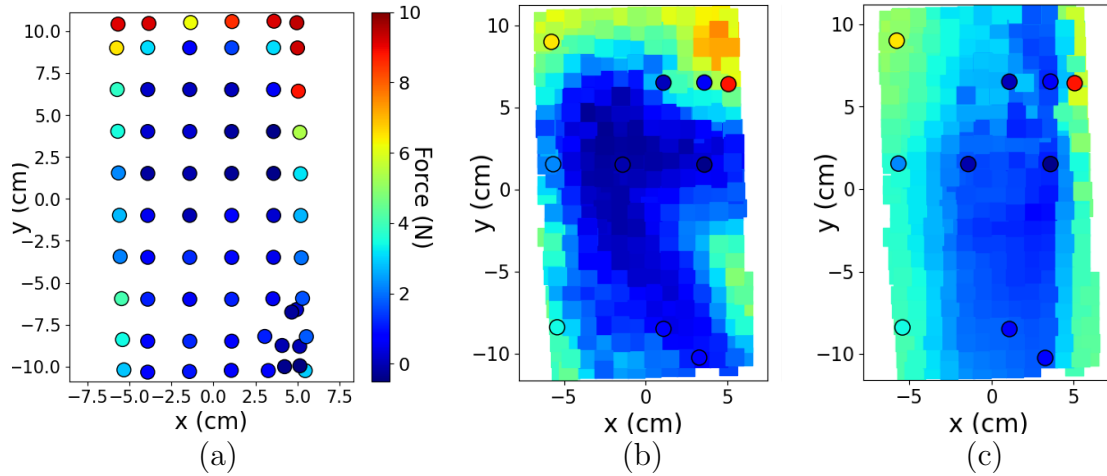


Figure 6.13: The ground truth and predicted force felt by a point probe at displacement = 3 mm on the rectangular tissue box object. The poked locations are indicated as circles, colored by the true force. The poked locations in (b) and (c) are the support data. (a) The ground truth force felt at the locations in the dataset. (b) The predicted forces by our method. (c) The predicted force by AutoML. Our method is able to better capture both the bias in the camera extrinsic calibration and the fact that the edges and corners of the tissue box is stiffer. [Best viewed in color.]

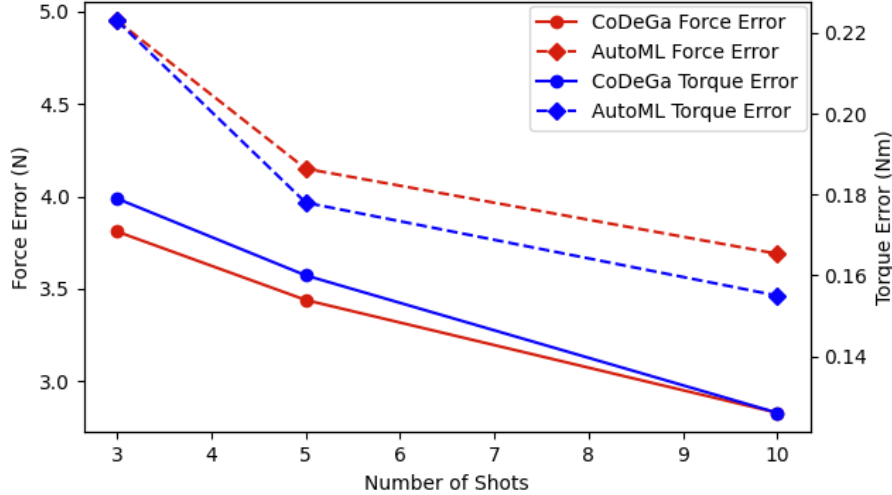


Figure 6.14: Results for the simulated contact wrench with the line probe using point models learned by CoDeGa and AutoML with different numbers of shots, averaged across all pokes for all testing objects. A medium-performing seed of CoDeGa is used. The force and torque error magnitudes are separately reported. [Best viewed in color.]

6.6 CONCLUSION AND FUTURE WORK

This chapter introduced a novel method for vision-based few-shot meta-learning. This chapter also demonstrates two applications of this method. For granular material scooping under domain shift, we demonstrate in real-world experiments that our proposed approach quickly achieves large scoop volumes on terrains that are drastically different from those seen in training, and significantly outperforms non-adaptive methods. In the future, we would like to explore more complex rewards, such as the outcome of a scientific assay from a sample analysis instrument. In addition, the force and torque the scoop experiences when executing a scoop action could be very informative about the underlying terrain and we would like to condition the scooping policy on these as well. Finally, our current method has no control over the controller that tracks the scooping trajectory other than the stiffness parameter. We would like to investigate feedback controllers that can react and adapt to terrain in real-time during scooping to further facilitate adaptation.

We also demonstrate the performance of CoDeGa for the point force response model learning of deformable objects, where it is able to significantly outperform pure online learning. This shows evidence that CoDeGa can be general to other robotics tasks, and complement gray-box learning.

CHAPTER 7: EXCAVATION OF FRAGMENTED ROCKS WITH MULTI-MODAL MODEL-BASED REINFORCEMENT LEARNING

While Chapter 6 presents an algorithm to plan for open-loop reference granular manipulation trajectories, this chapter presents a controller for the excavation of a single type of fragmented rocks given a reference trajectory. This chapter adopts a multi-modal model-based reinforcement learning (MBRL) approach to the excavation of fragmented rocks, which are very challenging to model due to their highly variable sizes and geometries, and visual occlusions. A multi-modal recurrent neural network (RNN) learns the dynamics of bucket-terrain interaction from a small physical dataset, with a discrete set of motion primitives encoded with domain knowledge as the action space. Then a model predictive controller (MPC) tracks a global reference path using multi-modal feedback. We show that our RNN-based dynamics function achieves lower prediction errors compared to a feed-forward neural network baseline, and the MPC is able to significantly outperform manually designed strategies on such a challenging task.⁸

7.1 INTRODUCTION

Excavators are versatile earth-moving machines that are widely used in mining, agriculture, and construction. Excavator operation requires highly skilled operators and oftentimes happens in hazardous environments that could cause injuries [186]. This has led to an increasing amount of effort in excavation automation [179], but one of the key challenges of excavation automation is that it is impossible to model the machine-earth interaction accurately [179], especially for complex, non-homogeneous terrains such as fragmented rocks.

In this work, we focus on the problem of excavation of fragmented rocks, and a real-world example is shown in Fig 7.1. Compared to homogeneous materials such as sand, the interaction between an excavator bucket and a pile of fragmented rocks is very challenging to model because the rocks are highly variable in sizes and shapes, and they are only visible from the surface. When following an arbitrary digging trajectory, an excavator bucket tends to get stuck because *jamming* among the rocks occurs frequently and generates large interaction forces. When human operators excavate fragment rocks, they need to find gaps between rocks to insert the bucket and continuously adjust and react to the change in the visual scene, interaction forces, and even engine noises of the machines. Another strategy they use when the bucket gets stuck is to wiggle the bucket to disturb and loosen up the rocks. However, this

⁸This chapter is reproduced from Yifan Zhu, Liyang Wang, and Liangjun Zhang, "Excavation of Fragmented Rocks with Multi-modal Model-based Reinforcement Learning". In IEEE International Conference on Intelligent Robots and Systems (IROS) 2022. This work is done during my internship at Baidu, USA.



Figure 7.1: Fragmented rocks excavation using an excavator.

type of strategy is challenging to discover automatically by a planner. Previous works have adopted admittance control [187, 188] and trajectory learning [189] for wheel loaders, but can only handle limited scenarios. Recently, model-free reinforcement learning approaches have been applied to finding excavation policies for wheel loaders [190] and excavators [191], but the training is done in simulation due to the requirement of large data. The bottleneck here is that accurate and efficient simulators are currently not available for complex terrains such as fragmented rocks.

We aim to address these challenges associated with excavation of fragmented rocks, including the difficulty in modeling terrain-bucket interactions and acquiring a large dataset, and the requirement for dexterous excavation skills. Towards equipping autonomous excavators with similar capabilities as humans, we propose a multi-modal model-based reinforcement learning (MBRL) approach to fragmented rocks excavation, that enables data-efficient learning using a small dataset of physical data and significantly outperforms manually tuned excavation strategies.

In our method, the excavation domain knowledge is encoded into a discrete set of primitive motions, and a recurrent neural network (RNN) learns the dynamics of bucket-terrain interaction with visual, tactile, and proprioceptive modalities using data collected in physical experiments. During online excavation, a global reference excavation path is given, which is tracked with a model predictive controller (MPC). The MPC follows the path whenever possible to maximize the excavated mass, and deviates from the path to avoid excessive contact forces that lead to jamming. We show that our RNN-based model significantly outperforms a feed-forward neural network baseline. We demonstrate from extensive experiments that our system is capable of completing excavations with high success rates of avoiding jamming, and outperforming manually designed strategies in tracking error and excavated mass.

7.2 LITERATURE REVIEW

7.2.1 Excavation Automation

There has been plenty of work in control [192, 193, 194], trajectory planning [178, 195], and task-level planning [196, 197, 198] for autonomous excavators and earth-moving machines. However, the majority of these works focus on excavation of homogeneous materials, such as sand and soil, and can not be directly applied to fragmented rocks. Here we review works on the excavation of fragmented rocks.

Control-based methods have been proposed for fragmented rocks excavation. Dobson et al. adopt an admittance control framework for fragmented rocks excavation by wheel loaders [187]. Fernando et al. extend this work by adapting the admittance control parameters online with iterative learning [188]. Compared to control-based approaches, our method performs local planning that gives rise to more versatile and optimal behaviors.

Dadhich et al. adopt imitation learning for wheel loaders by simply performing regression on expert demonstrations [189], but this approach could only handle simple scenarios. Model-free [190, 191] and model-based [170] RL approaches have been adopted in this problem domain as well, but the learning is performed in simulation, which deviates from reality. For example, when applying trajectories planned with a planner based on simulation data, failures due to jamming have been reported [170]. Our proposed method enables data-efficient learning from physical data and incorporates multi-modal sensory information to avoid jamming.

7.2.2 Contact-rich Manipulation with Multi-modal Observations

Humans are capable of seamlessly fusing multiple sensory modalities including visual, tactile, and proprioceptive information while performing contact-rich manipulation tasks, and it is a natural strategy for robots as well because each of these modalities is able to provide unique information and complement each other. In the recent years, it has become increasingly common to consider multi-modal observations for contact-rich manipulation tasks. Kappler et al. propose a method that would learn a set of manipulation skill with multi-modal sensory input, formulated as dynamic motion primitives, and store them in a manipulation graph used for planning [199]. Fazeli et al. demonstrate a robot that leverages hierarchical Bayesian representation with multi-sensory observations to play the game of Jenga. End-to-end learning has also been adopted [200], and Lee et al. utilize auxiliary goals to learn a latent state representation that fuses all modalities [201]. In addition, imitation learning [202, 203] and inverse RL [204] approaches have also been adopted for

contact-rich manipulation tasks with multi-modal observations. However, the majority of these methods focus on relatively simple tasks such as peg-in-hole, Jenga, and wiping, while we tackle a very challenging problem of fragmented rocks excavation.

7.2.3 Model-based Reinforcement Learning

MBRL approaches have been surveyed and compared recently in a survey [205] and a benchmark [206]. Compared to model-free methods, a model-based approach is more data-efficient and makes it possible to directly learn from real-world data, which is usually expensive to collect for robotics tasks. Compared to most of the MBRL methods for control and planning [206], our method adopts a discrete action space with primitives designed with domain knowledge, and uses discrete planning methods for the MPC.

7.3 PROBLEM DEFINITION

In the real world, e.g. construction sites, automation of excavators aims to maximize the amount of materials excavated per unit time. In the case of fragmented rocks, the excavator would execute a trajectory that allows the digging bucket to collect as much material as possible, while avoiding getting stuck since that would require the excavator to abort the current trajectory. In this work, we simplify the problem by giving a reference path that is designed to excavate a large volume if followed precisely, and aim to plan actions to follow the path, using multi-modal feedback. Ideally, the executed trajectory would follow the reference path when possible, but also deviate from the path to avoid getting stuck due to large contact forces.

We use a 7-DOF Franka Panda arm and irregular wooden blocks to emulate an excavation scene of fragmented rocks. The excavation setup and the frame definition are in shown in Fig. 7.2, where we use an overhead RGB camera and mount a digging bucket to the robot arm end-effector. We note that although as explained in the next paragraph, the emulated excavator has 4 DOFs, the same as an actual excavator, the geometries of the configuration space are different. However, this would not affect the generality of the proposed method itself. We represent the state of the robot and the environment at time t as $s_t \in \mathcal{S} = \mathcal{P} \times \mathbb{R}^3 \times \mathcal{I}$, which includes the bucket pose, the contact force at the bucket and an RGB image from the overhead camera. Since an excavator arm only has 4 degrees of freedom, we restrict the robot’s bucket pose to only the 3 translations and rotation around the world y-axis, i.e. $\mathcal{P} = \mathbb{R}^3 \times SO(2)$. For notation convenience, we also represent the state without the visual information at time t as $q_t \in \mathcal{Q} = \mathcal{P} \times \mathbb{R}^3$.

When an excavator bucket encounters large resistive forces from the substrate, a human operator often adopts a strategy to wiggle the bucket to disturb and loosen the rocks. We take advantage of such domain knowledge and instead of using continuous controls as the action space, we define a set of 9 discrete action primitives \mathcal{A} as the action space. The primitives include Cartesian movements and rotation of the bucket, and wiggling of the bucket:

- Cartesian movement of the bucket in both the positive and negative x-, y-, and z-directions in the world frame. Each movement is 1.5 cm in length with orientation fixed, executed at 1 cm/s.
- Positive and negative bucket orientation change in the y-axis for 0.1 rad (5.7°) at 1 rad/s.
- Wiggling of the bucket in the x-axis for 4 seconds, where the displacements follow a sine wave with magnitude 2 cm and frequency 1 Hz.

We empirically chose these actions and their parameters for our excavation scene. In the future, it would be beneficial to investigate how to systematically choose the action parameters based on the size of the fragmented rocks. To execute an action a_i , a joint-space trajectory is first generated using inverse kinematics (IK) and sent to the joint impedance controller of the Franka arm with impedance set at 100 Nm/rad.

Throughout this project, we follow the common practice in the excavation literature and divide a reference path into 4 phase: digging, dragging, closing, and lifting [170, 181]. We assume that the bucket goes toward the base of the robot, and to simplify the problem without the loss of generality, we only consider trajectory in the x-z plane of the robot. With these assumptions, the reference path is defined as $f_{ref} : [0, 1] \rightarrow SE(2)$, an example is shown in Fig. 7.3. Simply tracking such an trajectory on fragmented rocks often results in the bucket getting jammed. In our experiment setup, due to the use of a safe collaborative robot, the robot controller would take over and issue an halt to the robot when a large contact force is detected, which we use as a criterion for whether the robot has jammed or not throughout the chapter.

Throughout the chapter, due to the use of a discrete action space, a trajectory of length T is defined as T actions, along with the $T + 1$ states with or without visual information before and after executing the action, and we ignore the states of the robot and the environment while the primitive action is being executed.

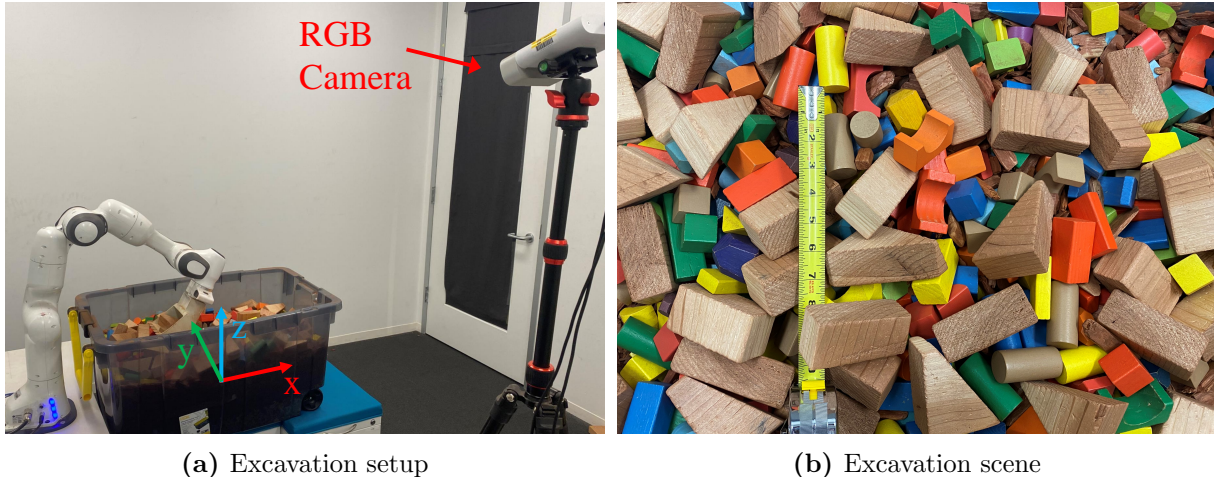


Figure 7.2: Excavation setup with the world frame labeled and the excavation scene with scale.

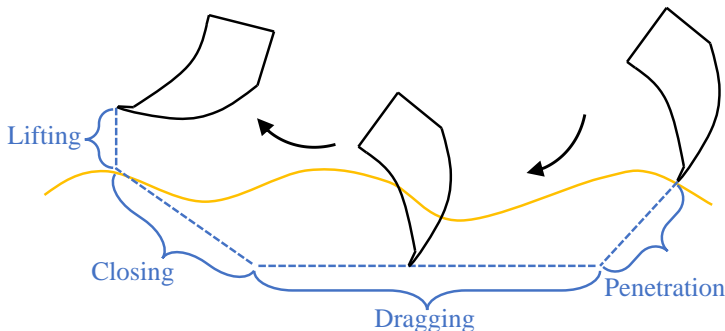


Figure 7.3: Reference digging trajectory.

7.4 DYNAMICS LEARNING

We adopt a MBRL formulation, where we first collect a dataset of trajectories offline on the physical setup, train a dynamics function for a sequence of actions, and use an MPC to track a reference path online. We shall discuss dynamics learning in this section, and MPC in Sec. 7.5.

Since only the surface of a pile of rocks is visible, we hypothesize that in order to accurately predict the excavation dynamics, a robot needs not only the current state of the system, but also state history, which is shown to be true through experiments in Sec. 7.6. Therefore, the goal of dynamics learning is to, at time t , predict a horizon k of future states without visual information $\tilde{q}_t = [q_{t+1}, \dots, q_{t+k}]$ given a history of states of length l : $\bar{s}_t = [s_{t-l+1}, \dots, s_t]$, and future actions $\tilde{a}_t = [a_t, \dots, a_{t+k-1}]$, where each a_i is an integer. Since this is a sequence-to-sequence prediction task, we propose to use an RNN to learn such a dynamics function, which encodes visual, proprioceptive, and tactile modalities.

The entire architecture is shown in Fig. 7.4, where the sizes of the fully-connected layers

are annotated. Each of the three modalities in each state is first encoded into a single latent feature. The RGB image is encoded with a ResNet50 convolutional neural network (CNN) model [207]. The bucket pose and contact wrench are encoded with separate fully-connected layers. These three modalities of encoded features are then concatenated into a single vector that passes through another fully-connected layer to a feature vector z . A sequence of length l of z vectors is then input into the RNN encoder.

Actions are one-hot encoded and they pass through a fully-connected layer before inputting into the decoder. The dimension of the one-hot encoding is 10, which is the size of the action space plus a null action. Separate fully-connected layers are used to obtain the pose and force from the decoder latent state. We apply batch normalization and the leaky ReLU activation function [208] to each fully-connected layer. In addition, We use a gated recurrent unit (GRU) [209] RNN model with additive attention [210]. Compared to a typical sequence-to-sequence machine translation, the main difference is that we have control inputs in the RNN decoder.

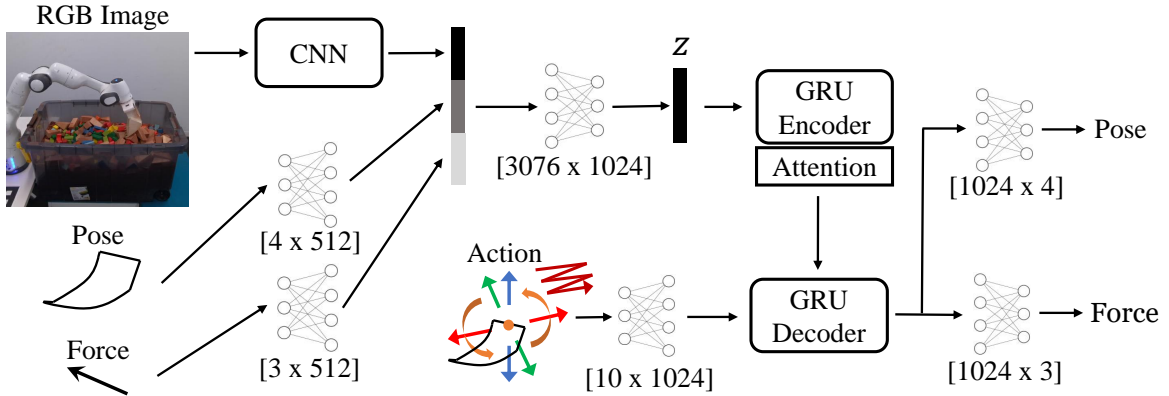


Figure 7.4: Dynamics learning architecture.

We collect training data by executing random and manually designed trajectories while recording visual, proprioceptive, and tactile data, detailed in Fig. 7.6.1. The trajectories are post-processed into training data points, where the i -th data point is denoted as $(\bar{s}_t^{(i)}, \tilde{a}_t^{(i)}, \tilde{q}_t^{(i)})$, where $\tilde{q}_t^{(i)}$ is the prediction target. In addition, we pad the start of each trajectory with $l-1$ null actions and copies of the first state in the trajectory. Let the dynamics function be $\hat{q}_t^{(i)} = f(\bar{s}_t, \tilde{a}_t)$, where $\hat{q}_t^{(i)}$ is the predicted future states, then for N data points, we minimize the following loss:

$$\sum_{i=0}^{N-1} \sum_{j=0}^{k-1} \left\| c^T (\hat{q}_t^{(i)}[j] - \tilde{q}_t^{(i)}[j]) \right\|_2^2, \quad (7.1)$$

where c is a weight vector, and we give the bucket pose a weight of 10 and the wrench a

weight of 1 .

7.5 PLANNING

We adopt the MPC scheme to reactively plan actions to track a reference path, where the planner repeatedly plan a trajectory of length k based on a cost function and execute the first action. This allows the planner to adjust to feedback when applying each action and be robust to dynamics prediction error. The cost function we aim to minimize is a weighted sum of three terms:

$$\begin{aligned}
 J(\tilde{a}_t, \tilde{q}_t) = & \\
 w_1 \cdot g_{prog}(f_{ref}, \tilde{q}_t) + w_2 \cdot g_{track}(f_{ref}, \tilde{q}_t) + w_3 \cdot g_{col}(\tilde{q}_t), & \tag{7.2}
 \end{aligned}$$

where g_{prog} measures the progress along the reference path, g_{track} weighs the tracking performance of the planned trajectory, g_{col} calculates the penalty given to large contact forces. We set w_1, w_2 , and w_3 to 0.8, -1.05, and -1/1500, respectively, which are tuned empirically. The goal is to maximize g_{prog} while minimizing g_{track} and g_{col} . The exact definitions of these 3 terms are introduced in Sec. 7.5.2.

The planning problem is then defined as:

$$\begin{aligned}
 \operatorname{argmax}_{\tilde{a}_t} \quad & J(\tilde{a}_t, \tilde{q}_t) \\
 \text{s.t.} \quad & \tilde{q}_t[\cdot] \in \mathcal{Q}, \tilde{a}_t[\cdot] \in \mathcal{A}, \\
 & \tilde{q}_t = f(\bar{s}_t, \tilde{a}_t),
 \end{aligned} \tag{7.3}$$

where \mathcal{Q} is the space of feasible robot configurations and $f(\cdot, \cdot)$ is the dynamics function.

7.5.1 Planning Algorithm

Due to the discrete nature of the actions, we consider 3 well-known methods for discrete planning, including brute-force search, random shooting (RS), and Monte Carlo tree search (MCTS). In brute-force search, each possible trajectory is evaluated and the optimal trajectory is selected. However, this method is limited by the planning horizon since the number of possible trajectories increases exponentially with the prediction horizon. For RS, a fixed number of random trajectories are sampled and evaluated. This number is the computation budget. MCTS is a heuristic search algorithm, where a search tree is iteratively expanded by balancing exploration and exploitation while each node is assigned a utility that is the average of the rewards, i.e. negative costs, of randomly sampled future trajectories starting

from this node. In our implementation, a fixed number of random trajectories are evaluated at each expansion step of MCTS. The computation budget for MCTS equals this fixed number multiplied by the total number of expansion steps.

7.5.2 Excavation Specific Details

As defined earlier, q_t contains both the bucket pose and contact force, which we denote as $p = [x, y, z, \alpha]$ and $F = [F_x, F_y, F_z]$. For a bucket pose p , we evaluate its progress along the reference path by first finding the closest point $p_{closest}$ on the path in terms of Euclidean distance, ignoring the angular component. Then the progress $\epsilon \in [0, 1]$ is calculated as the path length, considering only the position, from the start of the path to $p_{closest}$ divided by the entire length of the path. g_{prog} simply averages the progresses for the k future bucket poses.

To calculate g_{track} , for a pose p , we alternatively calculate its distance to the reference path by calculating the Euclidean distance between p and $p_{closest}$ including the angle, where the position and angle use meter and radian as units, and the angles are weighted by 0.15. g_{track} could be defined as the average of the distances of the k future poses. However, due to the granularity of the discrete actions, we allow small deviations from the reference trajectory and use the average of the adjusted distances, where each adjusted distance scales the portion of the original distance that is below a threshold of 0.015 by 0.2.

g_{col} is used to penalize large contact forces, and is an average of k future contact force penalties, where for each contact force, zero penalty is given when the contact force is below a threshold of 17.5 N, and is penalized with a coefficient of 1 over the threshold.

Finally, since large contact forces during lifting rarely occur, the reference path does not contain the lifting phase. When the robot finishes closing, we simply execute lifting by sending open-loop commands. We also note that the parameters described in this section are empirically determined.

7.6 RESULTS

7.6.1 Data Collection

One goal of data collection is to obtain data that cover a wide distribution. To do this, we adopt 3 strategies, including applying random actions, following scripted policies, and following scripted policies with stochasticity.

For random actions, the robot randomly executes the 9 actions with equal probability,

with boundaries on the position and orientation of the digging bucket to ensure safety. We consider 3 types of scripted policies. In the first policy, denoted as "Follow", the robot executes the action that minimizes its distance from a moving reference point on the reference path, where the distance is defined as that of g_{track} without scaling in 7.5.2. In particular, once the distance in the x position between the bucket and the reference point is smaller than 0.04, the reference point moves along the trajectory for 2 cm. Therefore, this policy does not consider the wiggling action. The second policy (denoted as "Closed-loop") is the same as Follow, but wiggles every 2 actions. The third policy (denoted as "Open-loop") is an open-loop version of Closed-loop, where the entire action sequence along the path is first planned in a similar fashion by assuming perfect execution of the actions, also with wiggling every 2 other actions.

To introduce randomness into the scripted policies, we use a modified version of Follow. Instead of picking actions in a deterministic way, we assign nonzero probabilities to select each of the actions and bias the action that would minimize the distance to the reference point.

Using these strategies, we collect a total of about 300 trajectories, and split them into training, validation, and testing data with a ratio of 8:1:1.

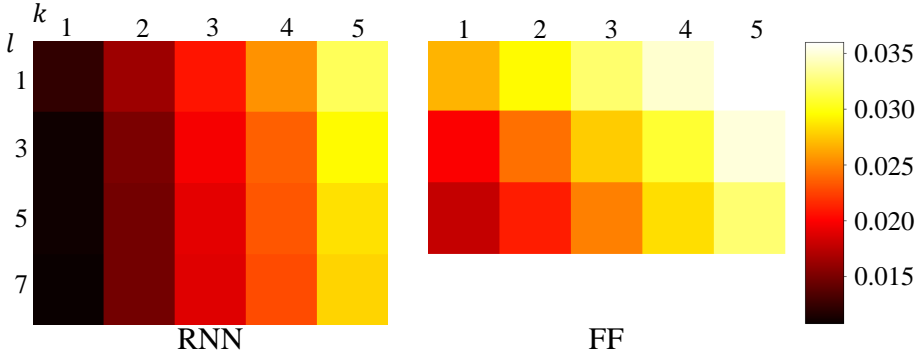


Figure 7.5: Loss of RNN and FF with different history lengths and prediction horizons on the testing data. [Best viewed in color.]

7.6.2 Dynamics Learning

We compare our method (denoted as RNN) against a feed-forward neural network baseline (denoted as FF). FF also uses the same multi-modal encoding layers as RNN to obtain the history features. The model concatenates the history features and the future controls, and predict the next k states all at once. Note that we cannot adopt the scheme where we predict only the next state, move the window of history forward one time step and predict again

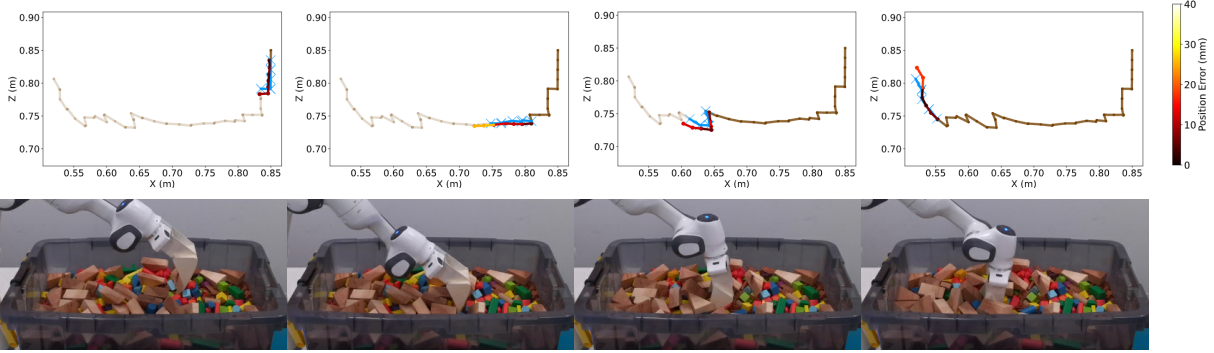


Figure 7.6: Four snapshots of position prediction errors along a testing trajectory with RNN. In every picture of the top row, the brown line represents the x-z position of the entire trajectory, with future trajectory shown with transparency. The blue crosses are the ground-truth positions for the 7 future states. The colored line are the 7 positions predicted by the learned dynamics, with colors representing position error. The bottom row contains the corresponding excavation scene. [Best viewed in color.]

such that a variable number of future states can be predicted. This is because we would not be able to predict the future images. For the architecture, we use a total of 5 fully-connected layers of sizes $(l+k) \times 512$, $(l+k) \times 1024$, $(l+k) \times 512$, $k \times 512$, and $k \times 64$. We also use batch normalization and the leaky ReLU activation function on each layer.

For RNN, we train a model with history length $l = 7$ and prediction horizon $k = 7$. For the FF baseline, we train three models with the same prediction horizon of 5, and history lengths of 1, 3, and 5. We do not train FF with l or $k = 7$ due to neural network model size and computer hardware limits.

One difficulty in predicting a horizon of future states with RNN is that errors in predictions accumulate and could blow up. This is a problem for the initial stage of training where the 1-step prediction error is big. Therefore, we perform RNN training in 2 stages, similar to the strategy used by [211]. In stage 1, we train the entire pipeline with a prediction horizon length of 1. In stage 2, we train with the full horizon length k . For both FF training and stage 1 of RNN training, we set the initial learning rate to $1e-4$, with an exponential decay rate of 0.7 every 10 epochs. We train with a total of 200 epochs with the Adam optimizer and batch size of 16, and set gradient clipping at 1. For stage 2 of RNN training, we use the same settings, except for the initial learning rate of $1e-5$ and a total of 100 epochs.

For our method, and both baselines, we compare the loss on the testing dataset, for different history length and prediction horizon, shown in Fig. 7.5. The trend in the loss on the rows and columns of the grid shows that a longer history length leads to a smaller prediction error and prediction errors increase with prediction horizon, for both RNN and FF. In addition, RNN outperforms FF consistently, while using fewer neural network parameters (excluding the multi-modal encoding layers, RNN is approximately 1/3 of the size of FF

Method	Testing Loss
all modalities	0.02191
all modalities w/ pretrained CNN	0.02606
no vision	0.02217
no tactile	0.02356
no vision nor tactile	0.02370

Table 7.1: Dynamics Ablation Studies

with $l = 1$). To give a sense of the scale of the prediction error, an average prediction error of 4.5 mm in position, 0.18 rad in angle, and 0.10 N in force is equal to a loss of about 0.3. Finally, we demonstrate the dynamics prediction and its quality along a trajectory in Fig. 7.6.

Ablation studies In the ablation studies, we intend to answer whether the visual and tactile modalities help improve state prediction. We also look into whether we could use pretrained visual features on other tasks to improve training efficiency. We train separate RNN models for each of the methods listed in Tab. 7.1 for $l = k = 7$, and report the loss on the testing dataset. For pretrained visual features, we use pretrained RNN weights on the ImageNet dataset [212]. The results show that the visual features learned on the ImageNet are not suitable for this task. At the same time, each modality decreases the loss, although it seems that the vision plays a smaller role than tactile information. This could indicate that the surface of the object pile is less useful. However, we suspect that this is more likely due to the fact that end-to-end learning does not lead to the best visual features, and we would like to pretrain the visual features with auxiliary tasks designed for excavation in the future.

Method	% of Jam	Pos. Err. (cm)	Ang. Err. (°)	Weight/Action (g)
Follow	50% / 60% / 100%	0.61 / 0.64 / -	2.2 / 2.2 / -	1.00 / 2.10 / 0.0
Closed-loop	10% / 20% / 60%	1.5 / 1.3 / 1.4	9.1 / 8.5 / 7.7	0.202 / 0.688 / 1.55
Open-loop	10% / 10% / 30%	1.2 / 1.4 / 2.0	5.4 / 6.4 / 6.5	0.341 / 0.669 / 1.40
BF, $k = 3$	0% / 0% / 20%	1.6 / 2.0 / 1.6	2.9 / 2.8 / 2.5	0.971 / 1.35 / 1.47
BF, $k = 5$	10% / 10% / 20%	1.6 / 1.7 / 2.2	2.1 / 2.4 / 2.7	0.954 / 2.37 / 3.13
RS, $k = 5$	30% / 10% / 30%	1.5 / 2.0 / 2.4	2.3 / 2.6 / 3.1	1.67 / 2.29 / 2.53
RS, $k = 7$	10% / 10% / 10%	1.4 / 1.9 / 2.8	2.9 / 3.2 / 3.5	1.65 / 2.64 / 2.90
MCTS, $k = 5$	10% / 10% / 20%	1.4 / 1.8 / 1.9	3.2 / 3.1 / 2.7	1.34 / 2.23 / 2.77
MCTS, $k = 7$	40% / 10% / 20%	1.2 / 2.3 / 2.0	3.3 / 3.3 / 3.2	1.84 / 1.45 / 3.11

Table 7.2: Excavation Experiments for Shallow/Medium/Deep Trajectories (BF: Brute-force)



Figure 7.7: The top and bottom rows show two example trajectories executed with MCTS, $k = 5$, tracking the deep trajectory. In the top row, a few objects are successfully excavated, while the excavation is aborted due to large contact forces in the bottom row.



Figure 7.8: Two cases that led to jamming. On the left, big blocks prevented penetration. On the right, the digging bucket teeth caught a red concave object. [Best viewed in color.]

7.6.3 Excavation Experiments

For the excavation experiments, in addition to evaluating the performance of the 3 different planners, we look into whether using wiggling helps excavation, and whether using a longer prediction horizon improves planner performance despite the larger errors in prediction. We compare the planners against 3 baselines, Follow, Closed-loop, and Open-loop, defined in Sec. 7.6.1. We use three previously unseen trajectories to evaluate the methods. Relative to the surface of the fragmented rocks pile, the trajectories are shallow, medium, and deep in depth, with the deepest trajectory about 8-10 cm beneath the surface during the dragging phase. We repeat 10 trials on each of the 3 trajectories, and we manually randomize and reset the object pile after each trial. If the robot gets jammed during any of the trials, we abort the current trial. Throughout the experiments in this subsection, we use the same

dynamics model architecture with the best performance, which is RNN with all modalities. Due to the small size of the dataset, we train a final model on both the training and testing data, and stop training when the validation error is minimized. We use a history length $l = 7$ for all the experiments.

We use 4 metrics to compare the performance of different planners, which are the number of trajectories that lead to jamming, the average position and angle tracking errors for the trajectories that do not jam, and the amount of material removed per action. The first three metrics are directly related to the terms in the cost function for tracking error and large contact forces. Since the ultimate goal in the real world is to maximize the amount of materials removed per excavator action, we also use the weight per action metric. For the trajectories that lead to jamming, we simply count the actions that have been executed before jamming happens, and consider the excavated weight as 0. The results are shown in Tab. 7.2. We experiment with prediction horizons of 3 and 5 for brute-force, and do not experiment with 7 due to computational tractability. For RS and MCTS, we experiment with prediction horizons of 5 and 7, all with a computation budget of 20,000 simulations. We show in Fig. 7.7 two excavation trajectories, one with successfully excavated objects and one that jammed, by MCTS with $k = 5$ tracking the deep trajectory.

Comparing Follow against the other two baselines, it is obvious that the use of wiggling helps reduce large contact forces, without which excavation always fails for a deep trajectory. All of our planners, with different settings, are able to significantly outperform the baselines. The results for brute-force demonstrate that using a prediction horizon of 5 instead of 3 improves excavation performance. Using an even longer prediction horizon of 7 shows different results for RS and MCTS, where RS is improved and MCTS is slightly worse. Overall, brute-force with $k = 5$, RS with $k = 7$, and MCTS with $k = 5$ achieve the best performance, but with brute-force using a greater amount of computation (59,049 compared to 20,000 simulations). We expected MCTS to outperform RS, a simple strategy, but we do not see such results from the experiments. We think that for the task of excavation, during the search in MCTS, the node with the lowest average cost might not be along the best trajectory. For example, an action of going down when the contact force is large, could lead to large costs on average for all possible future trajectories due to penalty for contact forces. However, the optimal trajectory might require the action of going down first.

One advantage of the method is data efficiency, where using only a total of about 300 trajectories result in policies that perform significant better than manually designed strategies. The data efficiency of method mainly comes from a model-based formulation and a well-designed action set of motion primitives where we encode domain knowledge.

We show two common cases that lead to jamming in Fig. 7.8, including a big object on the

way of penetration and a concave object that catches the digging bucket teeth and results in excessive contact forces. In the first case, our planner is unable to plan a local trajectory to get around it, due to limitation in the planning horizon as a longer planning horizon leads to increasingly inaccurate dynamics predictions. We believe that, in this case, a re-planning of the global reference path is necessary and we plan to investigate this in the future.

7.7 CONCLUSION

In conclusion, we propose a multi-modal MBRL approach for the task of fragmented rocks excavation. With a discrete action space of excavation primitives encoded with domain knowledge and the multi-modal RNN-based dynamics architecture, we show that we can learn the dynamics function from a small real-world dataset reasonably well to be used in planning. In addition, our MPC is able to significantly outperform manually designed strategies for tracking a global reference path, while brute-force, RS, and MCTS do not differ significantly in performance from each other.

In the future, we would like to extend our work to combine the re-planning of global reference path when our local planner is struggling, like the planner introduced in Chapter 6. The results also show that the visual features improve prediction accuracy marginally, and we would like to investigate better learning of the visual features, possibly by pretraining them with auxiliary tasks. Finally, we would like to implement our method on real excavators. One foreseeable challenge is the amount of training data needed compared to a controlled experiment in this chapter, and we intend to collaborate with industrial partners to scale up data collection.

CHAPTER 8: IMMERSIVE COMMODITY TELEPRESENCE WITH THE TRINA ROBOT AVATAR

This chapter takes a different angle at solving the challenges of completing contact-rich tasks in the real world. The idea is to leverage human intelligence through tele-operation of a robot avatar. Immersive robot avatars have the potential to aid and replace humans in a variety of applications such as telemedicine and search-and-rescue operations, reducing the need for travel and the risk to people working in dangerous environments. Many challenges, such as kinematic differences between people and robots, reduced perceptual feedback, and communication latency, currently limit how well robot avatars can achieve full immersion. This chapter presents AVATRINA, a teleoperated robot designed to address some of these concerns and maximize the operator’s capabilities while using a commodity light-weight human-machine interface. Team AVATRINA took 4th place at the recent \$10 million ANA Avatar XPRIZE competition, which required contestants to design avatar systems that could be controlled by novice operators to complete various manipulation, navigation, and social interaction tasks.

The original paper this chapter is reproduced from details the components of AVATRINA and the design process that contributed to our success at the competition. However, the paper is the collective effort of a large team and the paper contains sections that I have not contributed significantly. Instead, I will describe my efforts in the design and control of the manipulators of the TRINA system and the experience and lessons learned from competing in the ANA Avatar XPRIZE competition.⁹

8.1 INTRODUCTION

Telepresence [213], or telexistence [214], aims to enable a human operator to feel as though they are actually present in a remote robot’s environment through immersive vision and rich haptic feedback. Teleoperation of mobile manipulators has the potential to aid a wide variety of applications including telemedicine [20, 215, 216], search-and-rescue operations [217, 218], and remote environment exploration [219, 220, 221], by leveraging the perception and planning capabilities of humans. However, telepresence can be hindered by multiple factors such as kinematic differences between the robot and operator [222], reduced perceptual feedback, and network latency [223]. In addition, the cost of the robot and teleoperation hardware

⁹This chapter is reproduced from Joao Marques*, Patrick Naughton*, Jing-Chen Peng*, Yifan Zhu*, James Seungbum Nam, Qianxi Kong, Xuanpu Zhang, Aman Penmetcha, Ruifan Ji, Nairen Fu, Vignesh Ravibaskar, Ryan Yan, Neil Malhotra and Kris Hauser, "Immersive Commodity Telepresence with the TRINA Robot Avatar ". Under review for the International Journal of Social Robots.

limits the accessibility of telepresence robots. Exoskeletons, which are commonly used operator hardware for teleoperation [224] in research labs and industrial settings [225, 226], are not only costly but also require careful setup and calibration to use.

This chapter presents AVATRINA (AVAtar Tele-Robotic Intelligent Nursing Assistant), shown in Fig. 8.1, an immersive avatar robot operated with low-cost commodity hardware. The original paper that this chapter is reproduced from includes a complete design of the system, including its optimized kinematics, sensing and perception systems, and user interfaces, which creates an immersive telepresence experience, where the main contributions are:

- An immersive, novice-friendly teleoperation system with human-like manipulation, communication, and sensing capabilities, controlled by lightweight commodity operator hardware.
- A set of task-oriented metrics for optimizing robot hardware design for immersive teleoperation, complementary to traditional workspace analysis.
- Human subjects studies to validate the design of our perception system, showing that matching the distance between the robot’s eye cameras to the operator’s interpupillary distance and providing more degrees of freedom with which the operator can move the robot’s head can improve depth perception during teleoperation.¹⁰

In this chapter, however, I will only focus on the components that I have contributed significantly, which include the design and control of the manipulators.

Team AVATRINA, a collaboration between the University of Illinois and vRotors¹¹, built AVATRINA to compete in the ANA Avatar XPRIZE competition¹² finals. This competition aimed to accelerate the development of robot avatar technologies, improving the quality and variety of haptic sensing and rendering devices, and promoting fundamental research in system integration, networking, and virtual reality to create responsive, immersive, and intuitive telepresence systems. Our team achieved 4th place and was among the 4 teams that completed all 10 tasks at the competition¹³. In this chapter, I will also share the insights and experience gained from participating in the ANA Avatar XPRIZE competition and on how to design an immersive Avatar system.

¹⁰While this part is included in the original paper this chapter is reproduced from, I personally did not contribute significantly to this part, and thus this part is omitted in this thesis due to the length of this chapter.

¹¹<https://www.vrotors.com/>

¹²<https://www.xprize.org/prizes/avatar>

¹³<https://youtu.be/lOnV1Go6Op0?t=28364>

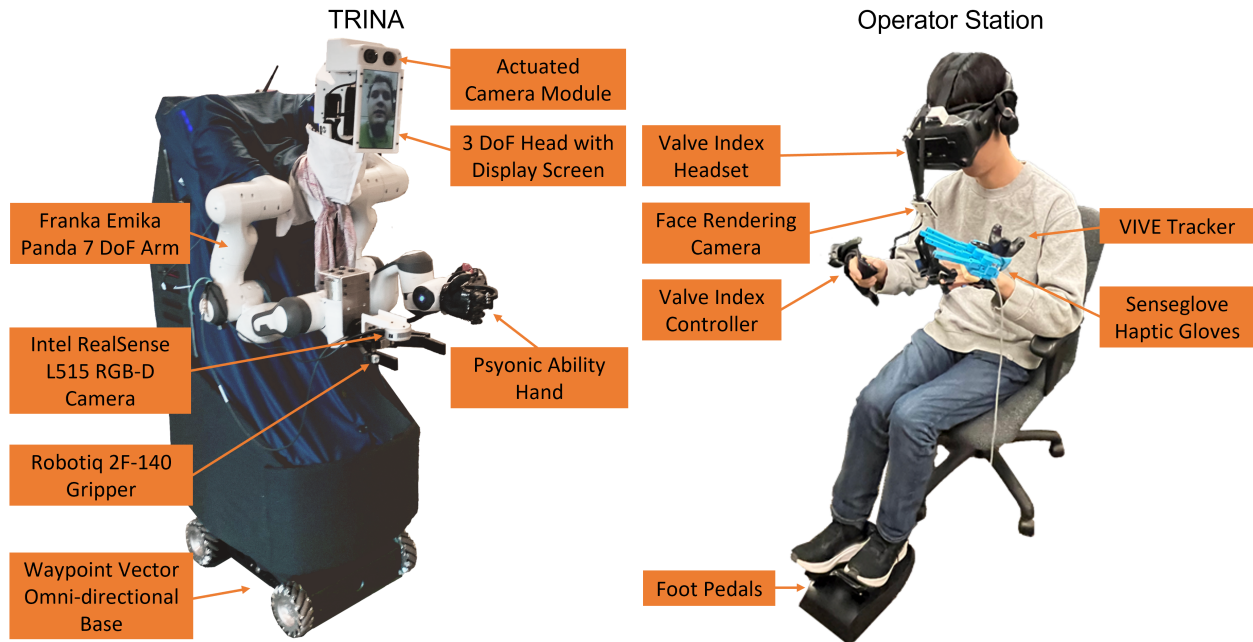


Figure 8.1: An overview of the AVATRINA robot and operator station. The robot is configured at the “home” configuration. [Best viewed in color.]

8.2 RELATED WORK

Avatar embodiments can take many forms depending on their intended applications. Many of the avatars presented during the DARPA robotics challenge [227] opted for humanoid bipedal robots with an alternative wheeled locomotion mode for faster traversal of flat ground and more stable environmental interaction. Two examples are team IHMC’s entry [228] and KAIST’s DRC-HUBO+ [229]. Other notable designs include RoboSimian’s quadruped design [230] and Team NimbRo’s wheeled centaur-like platform Momaro [231].

These robots were designed to operate in challenging locomotion scenarios and were primarily focused on efficient task completion under adversarial networking and environmental conditions. As such, their UI was focused on redundancy and robustness, with some interfaces requiring up to seven operators to man a single robot [231]. Furthermore, these robots fulfilled no social function and thus lacked expressive features for communication with remote individuals.

Avatars are not restricted to enabling humans to perform remote or dangerous activities. They can also be used to help people with disabilities expand their independence in performing activities of daily living and working [232, 233]. Being designed for indoor use, however, imposes other important constraints on these robots, such as maximum size, weight and cost, and introduces different manipulation requirements [234].

Yet other platforms focus on operator immersion and on the avatar’s socially expressive

capabilities. The TELESAR VI platform [235], for instance, follows a legged humanoid design with human proportions, but forgoes practical arm payloads and robot mobility in favor of enabling expressive arm, torso, finger and hand motion. Its interface also prioritizes immersion, using a VR Head Mounted Display (HMD) and haptic gloves which render finger forces, vibrations, and temperatures sensed from embedded sensors in the anthropomorphic avatar hand.

The ANA Avatar XPRIZE finals was a culmination of efforts to unify these research directions into socially capable, task efficient, and immersive robot avatars. Since the competition did not require challenging environment traversal, most teams opted for wheeled robot bases [236, 237, 238]. Indeed, as noted by [237], avatar embodiments that relied solely on legged locomotion all irrecoverably fell during the competition. Furthermore, as social tasks were an integral part of this competition, most embodiments paid special attention to properly rendering the operator’s face and voice.

Out of the 4 teams that completed all 10 tasks in the competition, 3 used VR headsets for immersive operation (NimbRo [236], Pollen Robotics and AVATRINA [238]), while Team Northeastern’s [237] system used an ultra-widescreen monitor for visualization. The teams that used VR employed different strategies to reconstruct the operator’s face without the headset on the robot, while Team Northeastern did not have to address this problem.

The user interfaces of the top 3 teams all used some form of exoskeleton for force haptic transmission to the arms and fingers. NimbRo’s consisted of a rig with additional robot manipulators mounted to the operator’s arms to render force, Team Northeastern leveraged mirrored hydraulic mechanisms between operator and avatar for rendering arm and finger forces, and Pollen Robotics used a 1 DoF elbow mounted exoskeleton to provide arm force feedback. AVATRINA was the only team to exclusively use vibratory and visual cues for rendering arm forces to the operator and complete the full course.

Furthermore, these systems used different methods for remote texture sensing - Pollen Robotics and NimbRo used acoustic based sensors [239], while Team Northeastern leveraged both contact acoustics and sensed vibrations and forces on the hydraulic actuators for identifying surfaces [237]. Our system captured a high-resolution heightmap of surfaces and used it to render distinct vibratory and auditory cues for each surface without requiring direct contact.

8.3 SYSTEM OVERVIEW

This section describes overall design goals for the system and an overview of the major components. Individual components will be described in following sections.

8.3.1 System Design Goals

AVATRINA is designed to be a socially-capable robot that can be controlled by a human *operator* to navigate and manipulate objects in remote environments designed for humans. The social capabilities should enable the operator to interact naturally with other human *recipients* in the remote environment. It should be easy to operate for novice users, accessible to a wide number of operators, and ergonomic enough for long-term usage. We emphasize that ergonomic and lightweight user interfaces are necessary to enable longer-term applications of teleoperation, such as tele-work and data gathering for imitation learning. Lightweight user interfaces improve scalability to many users and the likelihood of adoption. Moreover, AVATRINA should be a stable, maintainable platform to enable reproducible research in tele-nursing, telerobotics, and mobile manipulation for many years.

Our system design goals include:

1. The robot should have similar manipulation capabilities to humans, with human-like kinematics and load capabilities.
2. The operator station should be comfortable, ergonomic, and lightweight, requiring minimal setup.
3. The operator interface should be intuitive and immersive.
4. Recipients in the remote environment should feel the operator’s presence.
5. The robot should use as many off-the-shelf parts as possible for ease of construction, maintenance, and reproduction.
6. The robot should have easily reconfigurable end-effectors and sensors to support different applications.

8.3.2 Overall System Components

The overall system consists of the AVATRINA robot and the *operator station*, summarized in Fig. 8.1.

The robot is a bimanual mobile manipulator consisting of two robot arms (Franka Emika Panda), an omnidirectional mobile base (Waypoint Vector), and a custom 3 DoF anthropomorphic head with a custom adjustable baseline stereo camera. Other system specifications are listed in Tab. 8.1. We support different end-effectors, including the 6 DoF anthropomorphic Psyonic Ability hand, the Robotiq 2F-140 and 2F-85 parallel jaw grippers, the Franka hand, and the Righthand Reflex 3-fingered hand. The default configuration has the Psyonic

hand on the left arm and a 2F-140 Robotiq gripper on the right arm. The robot is similar in size to an adult, standing at 1.85 m tall and 0.7 m wide, and has a forward reach of 0.5 m. Four ultrasonic rangefinders around the perimeter of the robot provide enhanced situational awareness. In the default configuration, there is an RGB-D sensor mounted at the wrist of the right arm for close-range inspection and texture sensing. TRINA can operate for about 2 hours on its onboard battery. The robot is connected to the Internet via WiFi and can connect to an operator station via a handshake configured through designated servers.

The operator station is designed to consist of commodity hardware, so that the robot can potentially be connected to any operator with Internet access. It consists of an desktop PC, a VR head-mounted display (HMD) and controllers, foot pedals, and the SenseGlove haptic gloves (whose specifications are noted in Tab. 8.1). We support various commercial VR products such as the Valve Index, HTC Vive Pro, and Meta Quest 2. The poses of the HMD, controllers, and gloves are tracked by a Vive tracking system. In the default setup (Fig. 8.1), the operator wears the Valve Index headset, a SenseGlove on their left hand, and holds a Valve Index controller with their right hand. Fig. 8.2 shows how these devices are mapped to AVATRINA’s different components to give the operator control over AVATRINA’s motion.

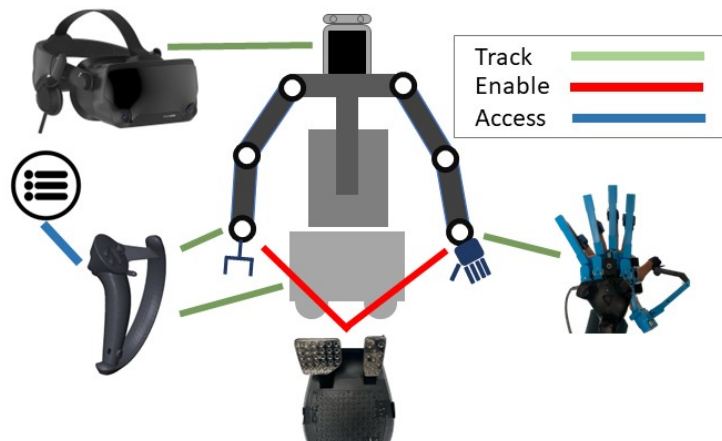


Figure 8.2: Mapping between operator control devices and AVATRINA’s movement. AVATRINA’s head tracks the VR headset and provides stereo camera feedback. The right VR controller controls the right gripper, and the right joystick commands the base translation and rotation. The left anthropomorphic Psyonic gripper tracks the SenseGlove. The foot pedals enable the movements of the end-effectors. The right controller can also bring up a menu of different semi-autonomous functionalities. [Best viewed in color.]

Overall	
Length x Width x Height	1.0 m x 0.68 m x 1.75 m
Weight	157 kg
Manipulation and Locomotion	
Payload	3 kg (including gripper)
Forward Reach	0.5 m
Max Speed	0.2 m/s
Sensors	
RGB-D Sensors	Configurable, default: 1
Stereo Camera	1
Birds-Eye Cameras	3
Ultrasonic Rangefinders	3
Miscellaneous	
Estimated Battery Life	2 hrs
Battery Capacity	1534 Wh
Onboard Compute	
Main Computer	
CPU	AMD Ryzen 9 5900X
GPU	NVIDIA RTX 3060 12 GB
RAM	64 GB DDR4 3200 MT/s
OS	Ubuntu 20.04
Control Computer	
CPU	AMD Ryzen 7 5800U
RAM	16 GB DDR4 3200 MT/s
OS	Ubuntu 20.04 (PREEMPT_RT)
Operator Station Compute	
CPU	AMD Ryzen 9 5900X
GPU	NVIDIA RTX 3090
RAM	32 GB DDR4 3200 MT/s
OS	Windows 11
Operator Station Hardware	
Head-Mounted Display	Valve Index Headset
Right Hand Tracking	Valve Index Controller
Left Hand Tracking	Vive Tracker and SenseGlove
Arm Activation	Thrustmaster F430 Pedals
Facial Rendering Camera	ELP-USBFHD01M-L170

Table 8.1: AVATRINA Canonical Specifications.

8.4 MANIPULATORS

AVATRINA both interacts with and is controlled by people, so the design criteria for its manipulators differ substantially from typical criteria for industrial robots. The operator is primarily concerned with the visibility of AVATRINA’s hands and the intuitiveness of the

arm control, while the recipient desires a compliant and predictable robot with which they can safely interact. This section describes how we designed the robot’s hardware and control software to meet these criteria.

8.4.1 Kinematic Design Optimization

In designing AVATRINA’s manipulation capabilities, we considered several criteria related to the mounting of the arms to the torso, including:

- The arms should have a large dexterous workspace, and be able to track the operator smoothly through that workspace.
- The arms should have human-like kinematics to make their behavior predictable for the operator and recipients.
- During manipulation tasks, the robot should minimize occlusion of the object being manipulated.
- The robot’s width should be less than 800 mm, to enable passing through the Americans with Disabilities Act (ADA) compliant doors¹⁴.

These criteria are highly coupled and often conflict with one another, making it difficult even for experienced engineers to reason about them through intuition alone. We therefore employed a *co-design* approach, using computational methods to optimize free design parameters (arm and hand mounting) while simultaneously considering the control software in the evaluation loop [240, 241].

The design parameters θ include the arm mounting $SE(3)$ transform and the 2 gripper mounting $SE(3)$ transforms. We assumed the arms will be mounted symmetrically, so we only optimize one shoulder mount and reflect it to obtain the other. To further simplify the problem, we first optimize the arm mounting transforms, then fine-tune the gripper transforms.

We propose four metrics to capture our design goals: 1) the robot’s ability to track the operator, 2) visibility of the grippers, 3) shoulder width, and 4) human-likeness. We manually proposed a set of promising design parameters θ , shown in Fig. 8.3 guided primarily by manufacturability constraints, and evaluated these metrics for each candidate design.

¹⁴<https://www.access-board.gov/ada/>

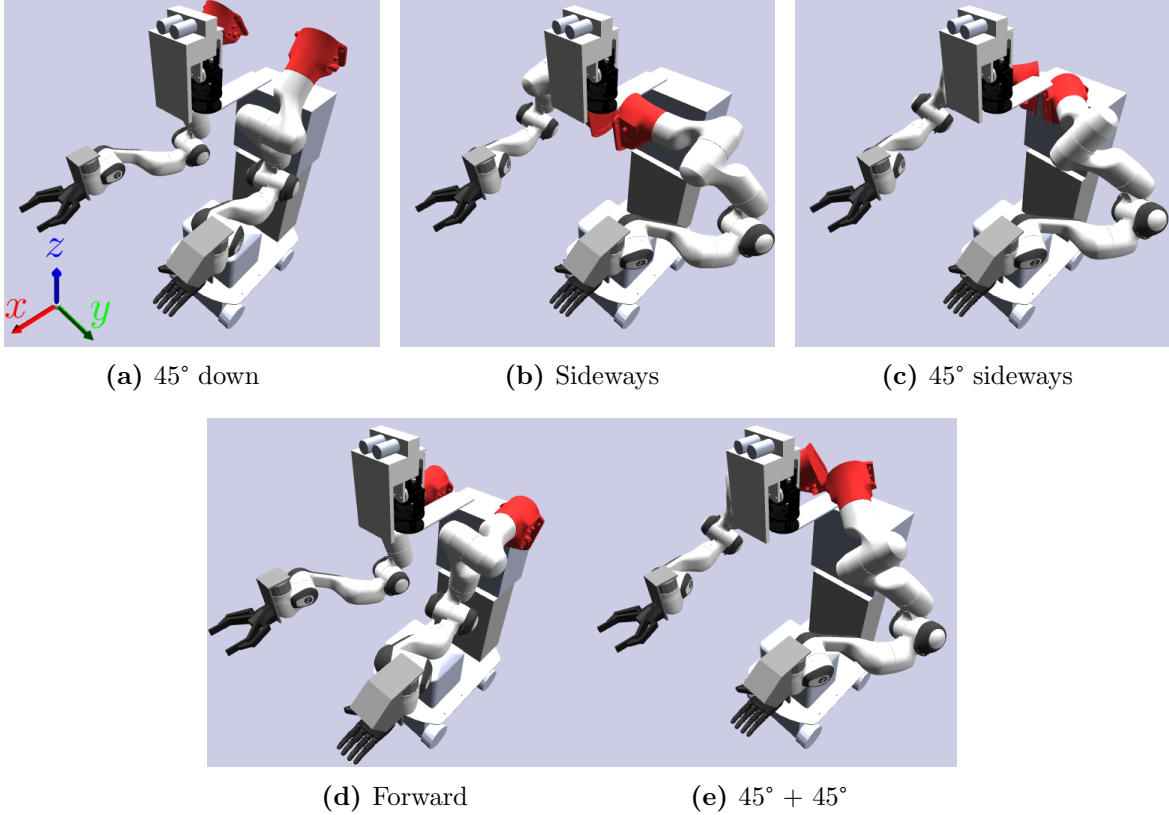


Figure 8.3: Candidate arm mounting poses. Shoulder attachment points are highlighted in red. [Best viewed in color.]

Design Metrics To evaluate our designs in context, we first record a person teleoperating a floating gripper in simulation to cover the robot workspace for representative manipulation tasks, such as tabletop manipulation (Fig. 8.4). Ideally the robot should be able to match these trajectories with its end-effector (EE). Each trajectory is given by $\mathbf{T} = (T_1, T_2, \dots, T_N)$, a list of N $SE(3)$ transforms expressed in the robot base frame.

For a candidate set of mounting parameters θ we can then simulate tracking the trajectory using the robot’s arm controller (Fig. 8.4.2). Each subsequent robot configuration q_i is derived by repeatedly applying the controller on the simulated robot state at time $i - 1$ to bring the EE towards T_i . The controller biases each joint towards the center of its limits to aid in redundancy resolution. The resulting joint trajectory $\hat{\mathbf{Q}}(\theta) = (q_1, \dots, q_N)$, is thus a function of θ . For ease of notation we also denote the simulated end-effector trajectory as $\hat{\mathbf{T}}(\theta) = (\hat{T}_1, \dots, \hat{T}_N)$.

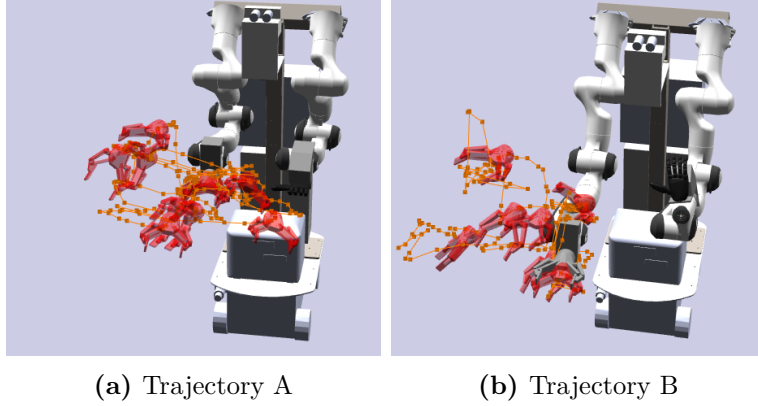


Figure 8.4: Trajectories used in kinematic optimization and inverse kinematics (IK) evaluation. Trajectory A is mostly within the robot’s expected workspace; Trajectory B extends further to the right, testing the robot’s ability to reach farther poses and recover from commands that exceed workspace limits. [Best viewed in color.]

The operator tracking error (OTE) metric M_t is defined as:

$$\begin{aligned}
 e_t([R|p], [\hat{R}|\hat{p}]) &= \mathbf{1}(\|p - \hat{p}\|_2 > \epsilon_x \\
 &\quad \text{or } \angle(R^{-1}\hat{R}) > \epsilon_\theta), \\
 M_t(\theta) &= \frac{1}{N} \sum_{i=1}^N e_t(T_i, \hat{T}_i(\theta)).
 \end{aligned} \tag{8.1}$$

Here $\angle(\cdot)$ indicates the angular deviation of a rotation matrix from the identity. The metric counts the number of time steps that tracking error exceeds thresholds in either position or orientation. We use this rather than the sum of pose errors, since small errors will not be noticed but tracking loss tends to confuse and frustrate the operator. We set the tolerances to $\epsilon_x = 0.05$ m and $\epsilon_\theta = 0.3$ rad.

The visibility metric is the fraction of the grippers not occluded by the arms as seen from the robot’s camera pose as the robot moves along $\hat{\mathbf{Q}}$. The camera pose is defined as the midpoint between the robot’s two eye cameras while the head points forward. For each simulated pose, we render the camera image to find the number of gripper pixels visible, $\hat{v}_g(\hat{q})$, and the number of total gripper pixels that would be visible if the arms were not present, $v_g(\hat{T})$. The occlusion metric is defined as:

$$M_v(\theta) = \frac{1}{N} \sum_{i=1}^N \frac{\hat{v}_g(\hat{q}_i)}{v_g(\hat{T}_i)} \tag{8.2}$$

The shoulder width metric M_w is defined by the maximum horizontal extents of the arms in the most compact home configuration. The human-likeness metric M_l is a subjective

measure, which is the average human-likeness score (on a scale of 1-5; higher is better) given by lab members after examining the robot’s appearance and behavior in simulation.

Mount	Trajectory A				Trajectory B	
	$M_w \downarrow$	$M_l \uparrow$	$M_t \downarrow$	$M_v \uparrow$	$M_t \downarrow$	$M_v \uparrow$
45° down	68 cm	4.00	0.13%	39.19%	42.76%	31.39%
Forwards	68 cm	2.25	14.88%	37.75%	55.45%	31.71%
Sideways	91 cm	3.25	29.60%	49.17%	43.33%	38.86%
45° sideways	79 cm	3.25	10.92%	41.26%	29.82%	35.02%
45° + 45°	79 cm	2.2	0.07%	36.30%	38.17%	32.71%

Table 8.2: Arm mounting evaluation. Lower width (M_w) and tracking error (M_t) is better; higher human likeness (M_l) and visibility (M_v) is better.

Results First, we compared five different arm mounting configurations and evaluated them based on M_w , M_l , M_t , and M_v . The Low mount (Fig. 8.5) was chosen arbitrarily for this evaluation. Results are shown in Tab. 8.2. We chose 45° down by prioritizing width, human likeness, and tracking error in the workspace in front of the robot (Trajectory A), while maintaining acceptable error in out-of-range reaching tasks (Trajectory B) and comparable visibility with other mountings that maintain good tracking.

We then compared different gripper mounting configurations (shown in Fig. 8.5) using M_t and M_v . For the right hand, we observed little difference between the low and high mounted grippers, as seen in Tab. 8.3, while the end mount exhibits especially poor tracking performance. This is because when oriented forward for tabletop manipulation tasks, the gripper places two of the wrist joints in near-singularity, causing many tracking failures. Ultimately, we chose the low mount to reduce wrist interference with obstacles when manipulating objects on a table. For both hands, the tracking metrics indicate that we should choose the higher mount. However, we noticed that this mount would result in collisions with the environment during tabletop manipulation tasks, a failure mode not captured by our metrics. Therefore, we chose the low mount despite it having slightly worse metrics. For future work, this indicates that metrics should incorporate more context of the robot’s tasks, such as expected environmental constraints.

8.4.2 Arm Control

AVATRINA’s arms are to track target end effector poses. Our arm control algorithm consists of two levels. On the higher level, an inverse kinematics (IK) solver computes

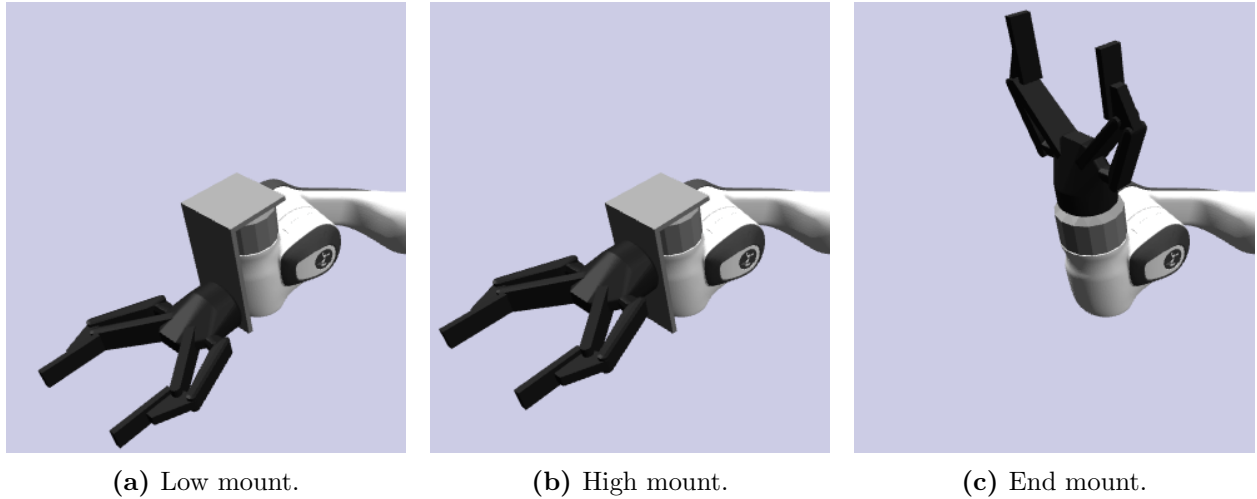


Figure 8.5: Candidate gripper mounts.

Side	Mount	Trajectory A		Trajectory B	
		$M_t \downarrow$	$M_v \uparrow$	$M_t \downarrow$	$M_v \uparrow$
Right	low	1.02%	0.566	28.41%	0.570
	high	0.22%	0.520	24.59%	0.549
	end	76.64%	0.343	82.59%	0.422
Left	low	0.14%	0.448	27.44%	0.711
	high	0.08%	0.365	7.77%	0.617
	end	44.86%	0.214	36.54%	0.322

Table 8.3: Gripper mounting evaluation results.

desired joint positions q_{desired} to track the target end effector pose T_{target} at 125 Hz for each arm independently. q_{desired} is then sent to the lower level controller if there is no collision. On the lower level, we compute joint torques τ_j to track q_{desired} at 1 kHz while also allowing for compliant motion to ensure safety during contacts.

Inverse Kinematics for Teleoperation Real-time tracking with IK is a challenging problem due to discontinuities, singularities, and local minima, which can cause abrupt movement or loss of tracking. To deal with this problem, we explored two different approaches to IK solving: modifying the target poses to ensure dexterity [242], and using a quadratic program IK solver that enforces smooth behavior [243].

In the first approach, we apply the measure of manipulability (MoM) method proposed by [242] that modifies the target transform sent to a standard Newton’s-method-based IK solver [244] such that the measure of manipulability of each arm remains above a threshold. The measure of manipulability (M_m) is given by

$$M_m(q) = \sqrt{\det \mathbf{J}\mathbf{J}^T}. \quad (8.3)$$

where $\mathbf{J} \equiv \mathbf{J}(q)$ refers to the Jacobian of the arm relating its 7 joint velocities to the end effector angular and translational velocity. M_m is positive semi-definite, reaching zero only when the robot arm is in singularity.

MoM prevents M_m from decreasing below a small threshold by 1) computing a correction motion that zeros out any components of the commanded motion (in task space) that decrease M_m if the arm is at or below this threshold, and at the same time, 2) introducing a small motion that pushes the robot in the direction of increasing M_m . We refer to the original paper [242] for more details.

The second approach uses a quadratic program (QP) solver based on the resolved-rate controller described in [243]. At every time step, a step in joint space δq is found as the solution to a constrained optimization problem where the objective function consists of tracking accuracy, joint velocity penalty, and a joint angle bias for redundancy resolution. The constraints encode the arm’s joint limits:

$$\begin{aligned}
\underset{\delta q}{\operatorname{argmin}} \quad & \|\delta r_{\text{target}} - \mathbf{J}\delta q\|_{W_r}^2 \\
& + \|\delta q\|_{W_q}^2 \\
& + \|q + \delta q - q_{\text{bias}}\|_{W_b}^2
\end{aligned} \tag{8.4}$$

$$\text{subject to } q_{\min} \leq q + \delta q \leq q_{\max},$$

where $\|r\|_W^2 = r^T W r$ is the weighted squared norm of a vector. This optimization problem is solved with a generic convex optimization problem solver [245] in real-time. In our implementation, we empirically chose $W_r = I$, $W_q = 0.02I$, and $W_b = 0.0005I$.

For both approaches, we considered the following bias configurations:

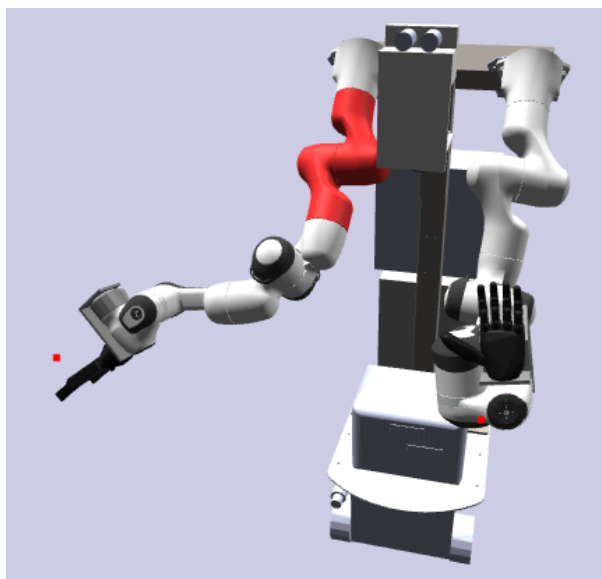
- Neutral bias: Bias solvers towards the middle of the arm’s joint limits.
- Shoulder angle bias: We developed a heuristic for the shoulder angle q_{shoulder} of the robot based on the target end effector transform, which guides how far outwards the elbow should swing. The IK solver biases the robot’s shoulder angle towards this value.

We evaluated the two different approaches for solving IK under different bias configurations by comparing the tracking metric M_t . The optimized robot design discussed in Sec. 8.4.1 and the same two trajectories (shown in Fig. 8.4) were used for all the experiments in this section. The results are reported in Tab. 8.4. QP generally provides lower tracking error, while MoM achieves faster computation. Tab. 8.4 also shows that the shoulder heuristic joint biasing generally improves tracking performance without sacrificing computation speed. One reason for this is that seeking more “natural-looking” elbow poses tends to help the robot avoid inverting the shoulder and elbow joints, as seen in Fig. 8.6, preventing the arm from getting stuck in regions of low dexterity. This is especially true for QP, which can allow the arm to enter singular configurations to achieve better tracking in the short term, at the expense of losing tracking at later points in the trajectory. We used biased MoM during the XPRIZE finals competition to ensure fast tracking, whereas the biased QP method is used in this chapter’s design evaluation (Sec. 8.4.1).

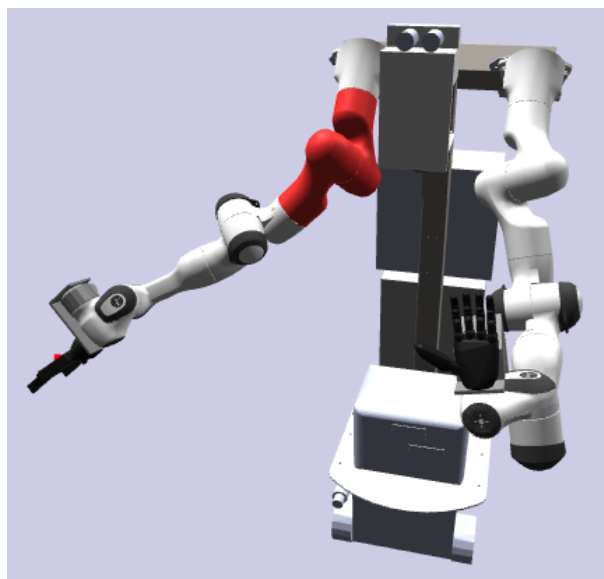
Low level torque control To track desired joint positions (q_{desired}) coming from the inverse kinematics controller, the low-level controller computes joint torques τ_{target} , which are sent to the robot’s low level controller at 1 kHz. The raw commands coming from the high level controller are passed through a complementary filter, and target joint velocities

Solving Method	Bias	Trajectory A		Trajectory B	
		$M_t \downarrow$	Time (ms) \downarrow	$M_t \downarrow$	Time (ms) \downarrow
Measure of Manipulability	Neutral	11.17%	0.41	32.60%	1.00
Quadratic Programming	Neutral	0.13%	1.32	44.47%	2.72
Measure of Manipulability	Shoulder angle	2.12%	0.40	33.60%	0.91
Quadratic Programming	Shoulder angle	1.22%	1.37	28.42%	2.97

Table 8.4: Tracking error rate for IK solver and bias heuristics, on two trajectories. Time is average time per loop iteration.



(a) IK with neutral bias “picks” the wrong direction for the shoulder.



(b) IK with heuristic bias keeps the shoulder angle on the correct side.

Figure 8.6: The specific failure case in trajectory B accounting for the poor performance of IK solving without heuristic biasing. See Tab. 8.4 for full results. [Best viewed in color.]

are computed as the velocity of the filtered target position using a finite difference method. τ_{target} is computed by

$$\begin{aligned}\tau_{pd} &= k_p \Delta q + k_d \Delta \dot{q} + \text{FF}(q, \dot{q}), \\ \bar{\tau}_{\text{target}} &= \text{plimit}(\tau_{pd}, \dot{q}) + c_q(q) + c_{\dot{q}}(\dot{q}), \\ \tau_{\text{target}} &= \text{clamp}(\bar{\tau}_{\text{target}}, \tau_{\text{max}}, \tau_{\text{min}}),\end{aligned}\tag{8.5}$$

where it uses a PD controller with the feedforward terms while respecting joint position, joint velocity, power, and torque limits. Here $\text{FF}(q, \dot{q})$ is the feedforward term, computed based on the dynamics model of the robot (provided by the manufacturer) to compensate for gravity and Coriolis terms. To account for joint position and velocity limits, we use a quadratic control barrier function, which imposes a smooth increase in resistance as the joint is pushed towards its position (resp. velocity) limits:

$$c_x(x) = \begin{cases} -K \left(\frac{x_{\text{max}} - x}{\epsilon_x} \right)^2 & \text{if } x > x_{\text{max}} - \epsilon_x \\ K \left(\frac{x_{\text{min}} - x}{\epsilon_x} \right)^2 & \text{if } x < x_{\text{min}} + \epsilon_x \\ 0 & \text{otherwise,} \end{cases}\tag{8.6}$$

where the parameters $K, \epsilon_x, x_{\text{max}}$, and x_{min} can be tuned based on the quantity being limited.

The Franka Emika Panda arm also has built-in joint power and torque limits for safety reasons. Violating these limits causes the arm to engage a protective stop, causing the operator to lose control of the arm temporarily. To prevent this, we add power and torque limits to our controller. The power limit is implemented as

$$\text{plimit}(\tau, \dot{q}) = \begin{cases} p_{\text{max}}/\dot{q} & \text{if } \tau \dot{q} > p_{\text{max}} \\ \tau & \text{otherwise,} \end{cases}\tag{8.7}$$

and the torque limit clamps between minimum and maximum torques. Here, \dot{q}_{target} is estimated by taking a finite difference derivative of q_{target} .

8.5 HAND CONTROL

In this section, we describe how the operator controls and receives force feedback from AVATRINA’s end effectors. The operator controls AVATRINA’s parallel-jaw gripper by pulling a trigger on their right-hand VR controller. Control of the anthropomorphic Psyonic Ability hand is more complex, requiring retargeting from the SenseGlove’s sensing to the

hand’s actuating capabilities.

8.5.1 Hand Motion Retargeting

The goal of hand motion retargeting is to map the motion of a high-DoF (21) human hand to a low-DoF (6) anthropomorphic gripper. The SenseGlove tracks 4 degrees of freedom (three in flexion, and one “splay”) for each finger. The full hand motion retargeting problem is defined as finding a mapping G between the 20 SenseGlove readings q_{glove} to 6 gripper motor positions, $q_{gripper}$, shown in Fig. 8.7. Note that the gripper uses a transmission mechanism so that a single motor command bends both joints in each finger in a synergistic manner.

Methods We seek a mapping of the form

$$G : (\mathbb{R}^4 \rightarrow \mathbb{R})^4 \times (\mathbb{R}^4 \rightarrow \mathbb{R}^2), \quad (8.8)$$

where each of the four fingers maps from four SenseGlove DoFs to one Psyonic DoF, while the thumb maps from four SenseGlove DoFs to two thumb DoFs.

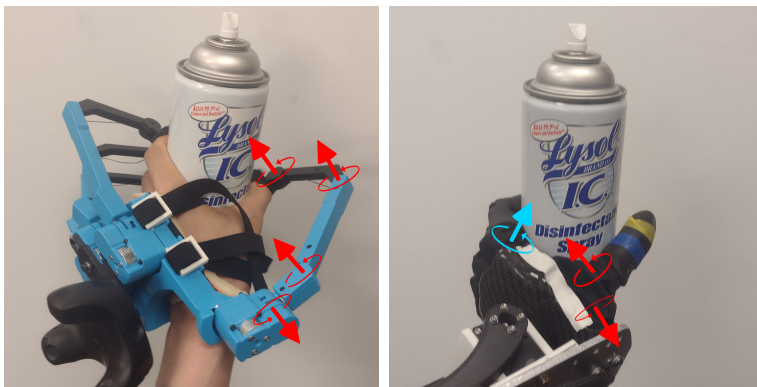


Figure 8.7: Hand Motion Retargeting. Each finger is measured with 4 DoFs on the SenseGlove (left), but the Psyonic gripper (right) only has 2 DoFs for the thumb (red) and 1 DoF per every other finger (blue). [Best viewed in color.]

To find this mapping, we designed a set of 15 gripper poses, shown in Fig. 8.8, that cover the Psyonic’s workspace and important grasps such as key and power grasps. To adapt to each operator’s hand, a calibration procedure is performed. The operator first mimics the gripper poses while wearing the SenseGlove and the corresponding SenseGlove readings are recorded. Not all fingers are considered for each calibration pose, since they may be difficult to imitate or irrelevant to the current calibration step. The mapping G is then learned by

supervised learning to minimize calibration error. Note that we learn a mapping for each finger and thumb individually.

The gripper poses used were designed to match the human poses visually; however this may not capture the operator’s intent. For example, for a pinch grasp on a thin object, the operator may want to pinch harder on an object, but once the operator’s fingers touch, the SenseGlove is not able to detect an intent to apply extra force. To compensate for this, we tweaked the target gripper joint angles to form a set of “biased” calibration poses in order to increase grasp success. Fig. 8.9 demonstrates how joint biasing is set up.

Results We evaluate the calibration performance on a different set of 4 grasps, shown in Fig. 8.10. First, we hand designed Psyonic poses for the four grasps. Then, each operator is instructed to physically perform each grasp while wearing the SenseGlove, and the SenseGlove joint angles are recorded. Finally, the recorded SenseGlove angles are transferred to the Psyonic using the mapping calibrated for the operator. These angles are compared against the hand-designed poses using two metrics: the mean absolute error (MAE) of the joint angles, and the grasp success score. The grasp success score is defined by using the Psyonic to pick up the physical object and repeatedly shaking the object with increasing intensity, as shown in Fig. 8.11. The number of shaking rounds before the object is dropped, up to 5, is recorded as the grasp success score. Two shaking trials are conducted for each grasp.

Biasing	Model	Grasp Success Score				Error (rad)
		Pinch 1	Pinch 2	Cube	Power	
manual	None	3.50	5.00	4.00	5.00	-
unbiased	GP	2.67	5.00	3.00	5.00	0.317
unbiased	KRidge	2.17	4.67	2.00	5.00	0.217
biased	GP	4.17	4.83	3.33	5.00	0.402
biased	KRidge	3.50	5.00	3.50	5.00	0.282

Table 8.5: Gripper calibration evaluation on four grasp attempts. Scoring represents the number of shaking cycles survived, averaged across six trials with three different operators. Error is grasp pose MAE averaged over all joints.

We experiment with two different regressors: Gaussian process regression and kernel ridge regression [184], and test each regressor using the unbiased and biased calibration poses. The calibration results for 3 researchers on this project are shown in Tab. 8.5. Overall, the two regressors perform similarly in grasp success, and both of them are helped by biasing the calibration poses. In terms of MAE, kernel ridge regression performs better than Gaussian

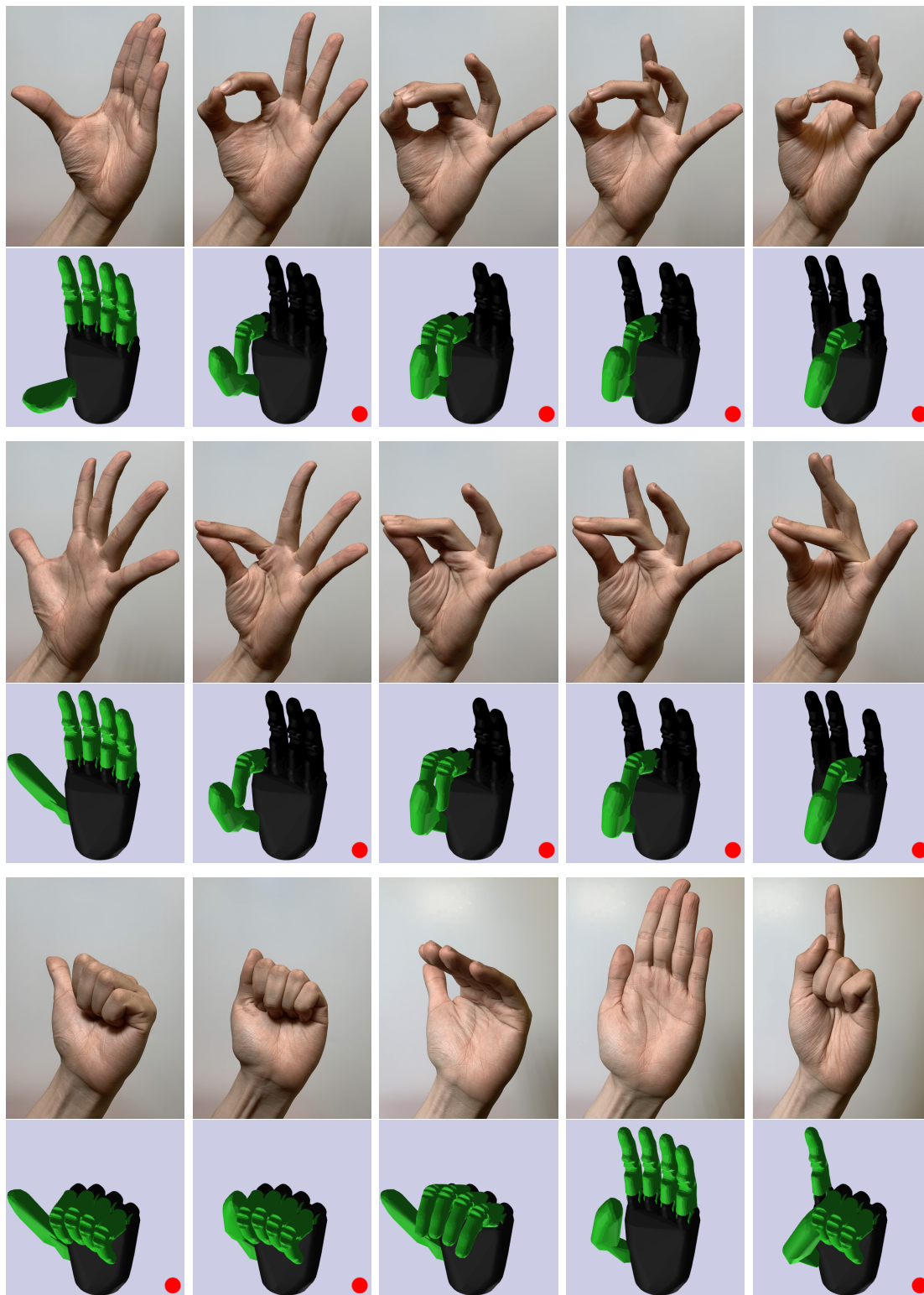


Figure 8.8: The hand calibration poses used. Human hand and target Psionic pose are shown. Green highlight shows fingers that are considered for the given calibration pose. A red dot indicates that joint angle biasing was applied to the pose. [Best viewed in color.]



Figure 8.9: Example for how the joint angles were biased for hand calibration. Bias effect is exaggerated for clarity. [Best viewed in color.]

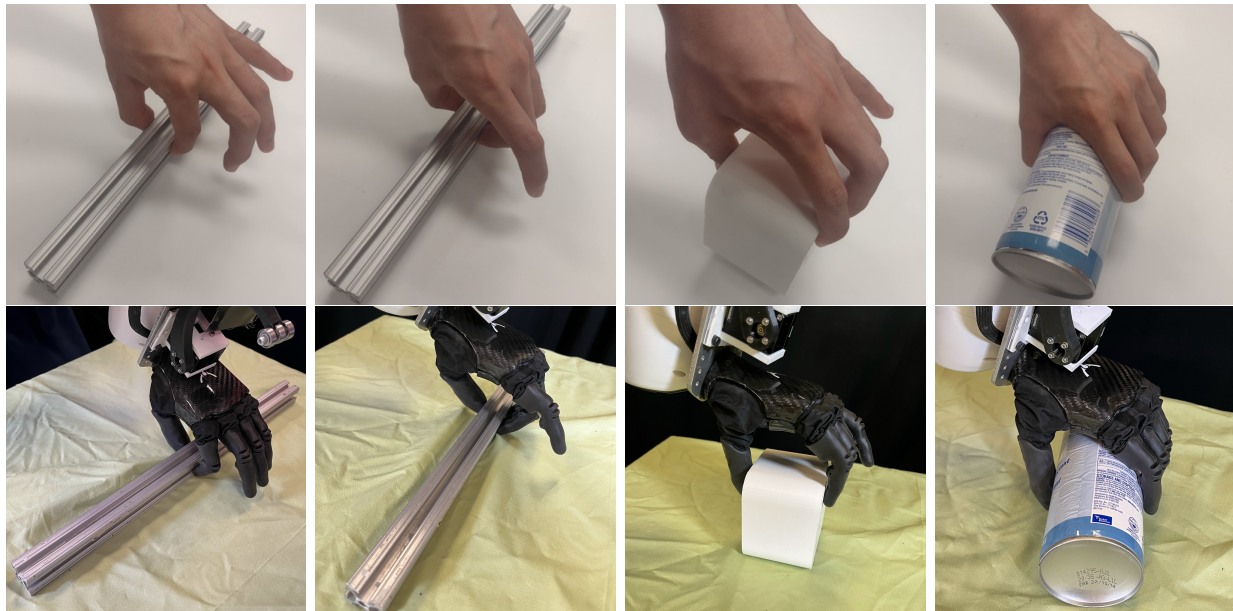


Figure 8.10: Pairs of poses used to test grasping. From left to right: Pinch 1, Pinch 2, Cube, Power.

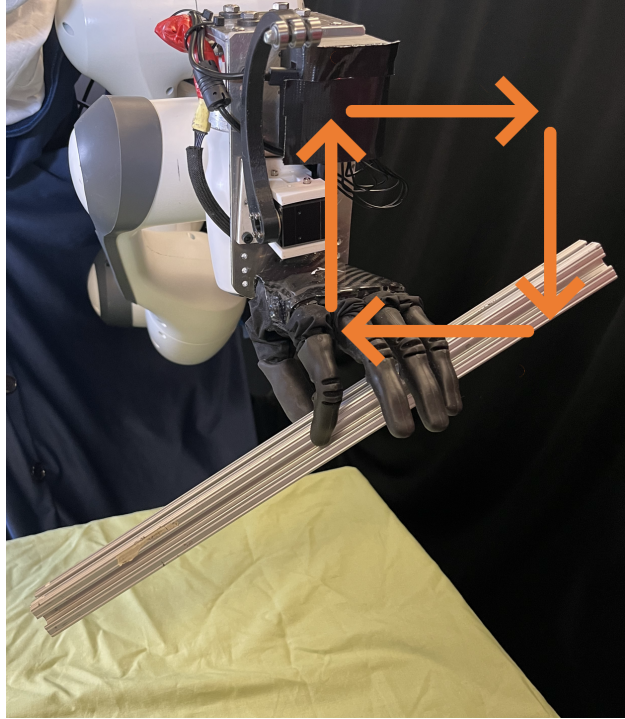


Figure 8.11: Illustration of shaking procedure. Robot picks object up from a predefined pose (in Fig. 8.10), then shakes in a square motion in the air with increasing intensity.

process regression. Using biased calibration increases grasp success, at the cost of increasing MAE. We decide to use biased calibration since the slight mismatch in angles could be compensated for by the operator.

8.6 END-TO-END EVALUATION: ANA AVATAR XPRIZE

Our team competed in the ANA Avatar XPRIZE finals competition using the AVATRINA robot in November of 2022. This competition required teams to train a new operator (a judge) on how to use their robot in 45 minutes to complete 10 different tasks in 25 minutes that ranged from navigation, to manipulation, to human interaction. The operator and recipient judges could also award up to 3 and 2 points respectively based on subjective impressions. More details on the tasks and scoring criteria can be found in the summary paper in this special issue [246].

8.6.1 Team AVATRINA Operator Training Procedure

During the competition, teams had 30 minutes to set up their robot and operator station, and 45 minutes to train a judge recruited by XPRIZE to use the robot to complete the 10

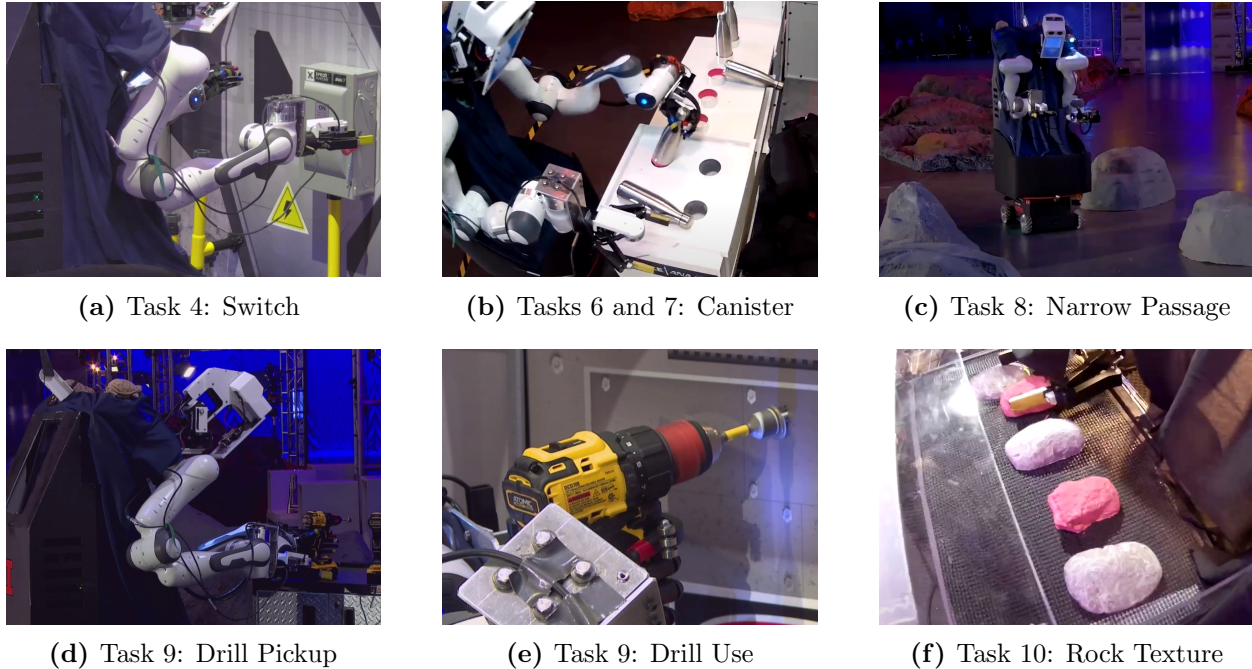


Figure 8.12: Images of AVATRINA executing some of the key tasks during the XPRIZE competition. [Best viewed in color.]

tasks as quickly as possible. To optimize our use of this time, we developed a training protocol to get the operator comfortable with controlling AVATRINA and allow them to practice the competition tasks. We rehearsed this procedure many times before the competition so that each team member knew their assigned role. One person was responsible for walking the operator through each of the basic functionalities of AVATRINA, then a different team member walked them through strategies used to complete each task. Tab. 8.6 outlines the training procedure.

8.6.2 Team Results

With a total score of 14.5/15 points, team AVATRINA came in 4th place overall and was one of four teams to complete all 10 tasks. Fig. 8.12 shows AVATRINA completing some of the tasks on the course. Our system proved easy to learn in the competition, giving judges ample time during the training period (20+ minutes) to practice on the actual competition tasks.

On our first official run, the operator successfully completed all 10 tasks. The operator made use of both the Psyonic gripper and the parallel jaw gripper for completing different tasks depending on which gripper was best suited for each task, highlighting the benefit of our asymmetric design. During the final task of texture sensing, they also used the assistive

Step	Time (min)
Measure operator IPD.	1
Show operator video of AVATRINA being teleoperated.	2
Don and calibrate the SenseGlove.	7
Don and adjust the HMD.	2
Learn and practice controlling AVATRINA’s head.	2
Learn and practice clutching each arm.	4
Learn and practice “homing” the arms.	1
Show warnings when AVATRINA enters a fault state.	1
Learn and practice moving AVATRINA’s base.	2
Practice manipulating canisters and feeling weight.	10
Practice picking up and using drill.	8
Practice scanning for and feeling rock texture.	5

Table 8.6: Outline of the training procedure used at the XPRIZE competition.

texture mode to scan and feel the textures of the various rocks. Interestingly, after scanning, the operator opted to disable the assistance to perform the final grasp, suggesting that more work needs to be done to automatically detect when the operator intends to use different assistive features. The competition conditions provided an unreliable network connection during this run, so we disabled our facial rendering pipeline to minimize the system’s required bandwidth. Despite this measure, the operator station still lost connection to AVATRINA partway through the run, which was resolved after a few minutes via an operator-requested manual restart of the operator station.

During our second run, our network connection was more stable, and our new operator completed tasks more aggressively, allowing them to reach the start of Task 10 within approximately 12 minutes. While attempting this task, AVATRINA’s arm collided with the environment with high force, causing it to engage mechanical brakes. Fig. 8.13 shows the operator’s view after this happened. Since the arm could not be remotely recovered, this prevented the operator from further using AVATRINA’s right arm. The operator tried to use the left arm to finish the task, but since there was no texture sensor on this arm, they were ultimately unsuccessful.

Further, we demonstrated that all 10 tasks could be completed without using arm force feedback mechanisms, indicating that lightweight interfaces can still be used to great effect for teleoperation. However, the emergency stop failure demonstrates the importance of designing remote resets into complex telerobotic system such as AVATRINA. Sensor-enabled collision avoidance strategies during driving and manipulation could have also avoided the failure.



Figure 8.13: The operator’s view after AVATRINA’s right arm engaged mechanical brakes in response to excessive force. [Best viewed in color.]

8.7 DISCUSSION AND LESSONS LEARNED

In this section, we offer our ideas of what the important considerations for building a successful Avatar robot are, lessons learned from the competition, and future directions.

8.7.1 One-to-one Correspondence

Multiple other teams have argued that making the mapping from operator to robot as simple [237] and as close to an identity transform as possible [236] maximizes operability and makes the system easy to learn. AVATRINA departs from this philosophy slightly by providing a user-configurable offset between the operator’s head and AVATRINA’s head, and by using a clutch-based, relative pose tracking system to control the arms. Both of these decisions were mainly motivated by the embodiment gap between AVATRINA and the operator: because the capabilities of AVATRINA do not exactly match those of the operator, AVATRINA often must complete tasks somewhat differently than a person would with their own body. For example, since the cameras and HMD used have a smaller field of view than a person does, the operator has to make more extreme movements with their head than they would in real life to gather the same amount of information or to achieve a satisfactory viewpoint to complete a task, which can be uncomfortable. The user-configurable offset between the operator’s head and AVATRINA’s head can alleviate some of this discomfort. A similar pattern is observed for AVATRINA’s arms, especially when using the parallel jaw gripper which functions much differently from a hand.

Despite these differences, we observed that many novice operators were able to quickly learn how to use the system, which suggests that one-to-one correspondence is not strictly necessary for task proficiency. This suggests an opportunity to improve operator ergonomics without sacrificing avatar capability, which must be considered more seriously if extended operation is required, or if the robot must complete a task in a manner that would be uncomfortable for a person.

8.7.2 Force Feedback

Teams at the competition adopted two different philosophies for force feedback to the operator. While many teams provided force feedback through exoskeletons, our team was the most successful system that adopted commodity VR hardware that does not offer force feedback. This suggests that while the lack of force feedback reduces the telepresence of the operator, novice operators are still able to complete complex manipulation tasks quickly. Since the comparatively low cost of these user interfaces could provide broader access to avatar technology, this presents motivation to further improve operator capabilities using accessible interfaces.

8.7.3 Autonomous Functions

The system we deployed at the XPRIZE finals incorporated some semi-autonomous functions, such as automatically returning the arms to the home configuration when the operator clicks a button. However, this was a small subset of functions we created during development, and we limited these functions to the most helpful subset to avoid overwhelming the operator and to improve training speeds. The tasks chosen by XPRIZE were sufficiently straightforward that autonomous assistance was not particularly helpful in many cases. For tasks that require more precision, such as handling hazardous or fragile materials, shared control may be more useful. Also, coordinated tasks like bimanual manipulation may also see more of a benefit from shared control. Moreover, the XPRIZE provided access to a relatively high bandwidth, low latency network, whereas less stable networks may render direct teleoperation difficult or even impossible. Autonomous and semi-autonomous functions may also be preferable for situations with unstable networks.

8.7.4 Robustness

System robustness in teleoperated systems is crucial, and although we performed significant amounts of testing and refinement of our system in the lab, human operators can trigger many unexpected conditions. Our system failure during the 10th task on Day 2 was due to an unrecoverable mechanical lock of the arm. Had the Franka Emika Panda arms provided an API to recover from such a state or had we developed more autonomous recovery protocols [236], the operator could have likely resumed the task.

8.7.5 Testing

In preparation for the XPRIZE competition, our team regularly tested our system under simulated competition settings. This allowed us to find and address false assumptions we had made about our system and the tasks, and gave us many opportunities to show our system to new operators, who often gave useful suggestions for new features or improvements. Frequent testing also helped to make set up of the system and training of the operator routine so that both could be done quickly and efficiently at the competition.

8.8 CONCLUSION

In this chapter, we presented an immersive, novice-friendly avatar system with human-like manipulation, communication, and sensing capabilities. Our system adopts exoskeleton-free operator hardware with commodity VR equipment and was the most successful exoskeleton-free avatar at the ANA Avatar XPRIZE competition. We believe this is essential for deploying Avatar robots in the world in the future where people can have easy access to the robot anywhere there is Internet connection. Finally, we offer our insights on how to build a successful immersive avatar system and lessons learned from the ANA Avatar XPRIZE competition.

CHAPTER 9: CONCLUSION

In this thesis, I demonstrate the use of gray-box learning and meta-learning to improve data efficiency on multiple challenging contact-rich tasks, including locomotion on and manipulation of granular media, learning visuo-tactile heterogeneous deformable object models, and auscultation. I also demonstrate my efforts in designing a human-in-the-loop system for contact-rich tasks and present the AVATRINA robot that achieved 4-th place in the ANA XPRIZE Avatar Challenge.

Next, I first suggest some future directions closely related to the research directions presented in this thesis and discuss more generally some of my thoughts towards future robotics research.

9.1 FUTURE DIRECTIONS

More Accurate Granular Media Contact Models and Online Contact Model Adaptation In Chapter 2, the proposed granular media contact model assumes that the stick-slip behavior is the dominant phenomenon. However, as we have discussed in Chapter 2, other effects such as memory effects can also be significant, and a contact model that considers these effects would further improve the accuracy of the simulator. This could be potentially tackled by employing a learning model for the feasible wrench space that is history-dependent. However, the challenge here is that the data required for learning such dependence could be large. In addition, while these wrench spaces can be carefully identified in the lab setting, a robot that operates in the real world would almost never encounter the exact same granular media whose wrench space would deviate from those learned in the lab. This challenge could be potentially tackled by applying a meta-learned wrench space model and performing online adaptation in a data-efficient manner.

Build Deformable Object Simulator from Manipulation Data While we set up a data collection apparatus for obtaining point poking experience with deformable objects in Chapter 4, the more naturalistic data collection strategy would be to obtain data from general manipulations, e.g. while a robot is grabbing an object with a gripper. This involves two main challenges. The first challenge is with sensing and perception, where the object is in gross motion, with visual occlusions and multiple contact locations. To deal with this challenge, one needs to leverage recent advancements in visual segmentation, tracking, and tactile sensing. Another challenge is solving the inverse problem - the collected data will no longer necessarily have point poking experience, and the point poking model needs to be

estimated from contact data with robot manipulators. In addition, the current simulator only supports rigid-deformable object interaction, and an extension is to also support rigid-rigid and deformable-deformable object interactions. This would require the simulator to not just solve force from surface displacements, which is what is currently done, but also solve force and surface displacements at the same time.

Vision-and-Tactile-based Adaptive Scooping System In the scooping system presented in Chapter 6, the signal for the surrogate model to adapt is the scooped volume, which contains a small amount of information. However, the close interaction of the scoop with the terrain during scooping contains very rich information about the latent variables that characterize the terrain, and including this information in the surrogate model is likely going to improve adaptation. While deep kernels are not very flexible for including such additional information, alternative classes of methods based on neural networks might be preferred. In addition, instead of planning open-loop reference trajectories, a close-loop scooping strategy that uses vision and tactile information to adapt while scooping would further accelerate adaptation and give more optimal scooping policies.

9.2 DISCUSSION

On the spectrum of learning vs modeling, I believe the future of robotics would be somewhere in the middle where physics and data would complement each other in the best way possible in the entire stack of the robot pipeline, as advocated by this thesis. To achieve this, new methods and frameworks will be needed to combine data and physics knowledge in a principled way, without hand-engineering models for each different task. Based on the topics and applications that I discuss in this thesis, here I also list a few exciting general future research topics.

Online Gray-box Learning of Novel Categories In this thesis, gray-box learning is done via hand-designed gray-box models. However, these models would only work for the narrow set of objects and environments that the model is designed for. On the other hand, pure black-box learning has the potential to learn arbitrary models given enough data. How to find a middle ground between these? Can we build gray-box models that are general to different types of physical phenomena but still enjoy the benefits of the structure given by the physics principles? In addition, can the robot also adjust the complexities of the gray-box model based on the amount of online data to embed Occam’s Razor?

Gray-box Policy Learning Gray-box learning is mainly explored in the context of dynamics learning in this thesis. However, it is not clear how to integrate the knowledge of physics into reinforcement learning policies that are notoriously data-hungry.

Continual learning. The framework of meta-learning aims to accelerate online learning of single tasks using priors learned from offline data. For example, in Chapter 6, once the deep kernels are trained, the hyperparameters remain fixed. However, humans and animals perform learning constantly using new data and experience. Continual and lifelong learning have the potential to empower robots to do the same but this field is still at its early stage.

Active Sensing in the open world. In most of the chapters in this thesis, the training data is a carefully curated dataset collected in a very controlled environment. How would a robot be able autonomously to explore the unstructured open world and build models of the world? This would require that the robot has a rich set of prior skills and knowledge. Different from obtaining prior knowledge from physics or offline data as done in this thesis on specific tasks, this would require a suite of general knowledge.

REFERENCES

- [1] M. W. Spong, S. Hutchinson, M. Vidyasagar et al., *Robot modeling and control*. Wiley New York, 2006, vol. 3.
- [2] J. Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason, “A Convex Polynomial Force-Motion Model for Planar Sliding : Identification and Application,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [3] A. Kloss, S. Schaal, and J. Bohg, “Combining learned and analytical models for predicting action effects,” in *arXiv:1710.04102*, 2017.
- [4] J. Wu, J.-x. Wang, and H. Xiao, “Physics-Informed Machine Learning for Predictive Turbulence Modeling: A Priori Assessment of Prediction Confidence,” *arXiv:1607.04563*, 2016.
- [5] A. Tasora, R. Serban, H. Mazhar, A. Pazouki, D. Melanz, J. Fleischmann, M. Taylor, H. Sugiyama, and D. N. B, “Chrono: An open source multi-physics dynamics engine,” in *International Conference on High Performance Computing in Science and Engineering*, 2015, pp. 19–49.
- [6] M. Raissi and G. E. Karniadakis, “Hidden physics models : Machine learning of non-linear partial differential equations,” *Journal of Computational Physics*, vol. 357, pp. 125–141, 2018.
- [7] G. Carleo, I. Cirac, L. Cranmer, KyleDaudet, D. Bourse, M. Schuld, L. Vogt-maranto, and L. Zdeborová, “Machine learning and the physical sciences,” 2019.
- [8] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.
- [9] M. Parmar, M. Halm, and M. Posa, “Fundamental challenges in deep learning for stiff contact dynamics,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [10] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende et al., “Interaction networks for learning about objects, relations and physics,” *Advances in neural information processing systems*, vol. 29, 2016.
- [11] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, “Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids,” *arXiv preprint arXiv:1810.01566*, 2018.
- [12] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, “Propagation networks for model-based control under partial observation,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

- [13] Y. Li, Z. O’Neill, L. Zhang, J. Chen, P. Im, and J. DeGraw, “Grey-box modeling and application for building energy simulations-a critical review,” *Renewable and Sustainable Energy Reviews*, vol. 146, p. 111174, 2021.
- [14] S. Thrun and L. Pratt, “Learning to Learn: Introduction and Overview,” *Learning to Learn*, pp. 3–17, 1998.
- [15] T. Wu and J. Movellan, “Semi-parametric Gaussian process for robot system identification,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.
- [16] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, “Gaussian processes and reinforcement learning for identification and control of an autonomous blimp,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007.
- [17] A. Ajay, J. Wu, N. Fazeli, M. Bauza, L. P. Kaelbling, J. B. Tenenbaum, and A. Rodriguez, “Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [18] Y. Yin, V. L. Guen, J. Dona, E. de Bézenac, I. Ayed, N. Thome, and P. Gallinari, “Augmenting physical models with deep networks for complex dynamics forecasting,” *Journal of Statistical Mechanics: Theory and Experiment*, 10 2020.
- [19] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, “Tossingbot: Learning to throw arbitrary objects with residual physics,” in *Robotics: Science and Systems (RSS)*, 2019.
- [20] Y. Zhu, A. Smith, and K. Hauser, “Automated Heart and Lung Auscultation in Robotic Physical Examinations,” *IEEE Robotics and Automation Letters*, 2022.
- [21] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics* 2021 3:6, vol. 3, pp. 422–440, 5 2021.
- [22] Z. Chen, Y. Liu, and H. Sun, “Physics-informed learning of governing equations from scarce data,” *Nature Communications* 2021 12:1, vol. 12, pp. 1–13, 10 2021.
- [23] S. Seo and Y. Liu, “Graph networks with physics-aware knowledge informed in latent space,” in *AAAI Spring Symposium: MLPS*, 2021.
- [24] M. Lahariya, C. Innes, C. Develder, and S. Ramamoorthy, “Learning physics-informed simulation models for soft robotic manipulation: A case study with dielectric elastomer actuators,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [25] S. Lee, H. Yang, and W. Seong, “Identifying physical law of hamiltonian systems via meta-learning,” in *International Conference on Learning Representations (ICLR)*, 2021.

- [26] Y. Jiang, J. Sun, and C. K. Liu, “Data-augmented contact model for rigid body simulation,” in *The 4th Annual Learning for Dynamics and Control Conference (L4DC)*, 2022.
- [27] M. Kim, J. Yoon, D. Son, and D. Lee, “Data-driven contact clustering for robot simulation,” in *International Conference on Robotics and Automation (ICRA)*, 2019.
- [28] S. Pfrommer, M. Halm, and M. Posa, “Contactnets: Learning of discontinuous contact dynamics with smooth, implicit representations,” in *Conference on Robot Learning (CoRL)*, 2020.
- [29] J. Schmidhuber, “Evolutionary principles in self-referential learning,” Diploma Thesis, Technische Universitat Munchen, Germany, 14 May 1987.
- [30] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, “Meta-Learning in Neural Networks: A Survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, apr 2020.
- [31] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical Networks for Few-shot Learning,” in *Advances in Neural Information Processing Systems*, mar 2017.
- [32] O. Vinyals, G. Deepmind, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, “Matching Networks for One Shot Learning,” in *Advances in Neural Information Processing Systems*, 2016.
- [33] F. Sung, Y. Yongxin, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to Compare: Relation Network for Few-Shot Learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
- [34] C. Finn, P. Abbeel, and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks,” in *34th International Conference on Machine Learning (ICML)*, 2017.
- [35] A. Nichol, J. Achiam, and J. Schulman, “On first-order meta-learning algorithms,” *arXiv:1803.02999*, 2018.
- [36] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *5th International Conference on Learning Representations (ICLR)*, 2017.
- [37] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, “Learning to learn by gradient descent by gradient descent,” in *Advances in neural information processing systems*, 2016.
- [38] D. Ha, A. Dai, and Q. V. Le Google Brain, “HyperNetworks,” *Hypernetworks in the Science of Complex Systems*, pp. 151–176, sep 2016.
- [39] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, “Meta-Learning with Latent Embedding Optimization,” in *International Conference on Learning Representations (ICLR)*, 2018.

- [40] D. Maclaurin, D. Duvenaud, and R. P. Adams, “Gradient-based Hyperparameter Optimization through Reversible Learning,” *PMLR*, pp. 2113–2122, jun 2015.
- [41] J. Rajendran, A. Irpan, and E. Jang, “Meta-learning requires meta-augmentation,” in *Advances in Neural Information Processing Systems*, 2020.
- [42] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, “Deep kernel learning,” in *International Conference on Artificial Intelligence and Statistics*, 2016.
- [43] M. Patacchiola, J. Turner, E. J. Crowley, M. O. Boyle, and A. Storkey, “Bayesian meta-learning for the few-shot setting via deep kernels,” in *Conference on Neural Information Processing Systems*, 2020.
- [44] P. Tossou, B. Dura, F. Laviolette, M. Marchand, and A. Lacoste, “Adaptive Deep Kernel Learning,” *arXiv:1905.12131*, may 2019.
- [45] T. Iwata and Y. Tanaka, “Few-shot Learning for Spatial Regression,” *arXiv:2010.04360*, 2020.
- [46] F. Alet, T. Lozano-Perez, and L. P. Kaelbling, “Modular meta-learning,” in *Conference on Robot Learning (CoRL)*, 2018.
- [47] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, “Learning multiple visual domains with residual adapters,” in *Advances in Neural Information Processing Systems*, 2017.
- [48] Y. Zhu, L. Abdulmajeid, and K. Hauser, “A Data-driven Approach for Fast Simulation of Robot Locomotion on Granular Media,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [49] Y. Zhu, Z. Pan, and K. Hauser, “Contact-Implicit Trajectory Optimization With Learned Deformable Contacts Using Bilevel Optimization,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [50] Y. Zhu, K. Lu, and K. Hauser, “Semi-Empirical Simulation of Learned Force Response Models for Heterogeneous Elastic Objects,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [51] Y. Zhu, P. Thangeda, M. Ornik, and K. Hauser, “Few-shot adaptation for manipulating granular materials under domain shift,” 2023.
- [52] Y. Zhu, L. Wang, and L. Zhang, “Excavation of Fragmented Rocks with Multi-modal Model-based Reinforcement Learning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [53] T. Erez, Y. Tassa, and E. Todorov, “Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX,” 2015.
- [54] Y. Bai and C. K. Liu, “Coupling cloth and rigid bodies for dexterous manipulation,” in *ACM/SIGGRAPH conference on Motion, Interaction and Games (MIG)*, 2014.

- [55] D. M. Conachie and D. Berenson, “Bandit-Based Model Selection for Deformable Object Manipulation,” *arXiv:1703.10254*, 2017.
- [56] H.-j. Kang, K. Hashimoto, K. Nishikawa, E. Falotico, H.-o. Lim, A. Takanishi, C. Laschi, P. Dario, and A. Berthoz, “Biped walking stabilization on soft ground based on gait analysis,” in *IEEE International Conference Biomedical Robotics and Biomechanics*, 2012.
- [57] C. M. Hubicki and J. W. Hurst, “Running on Soft Ground: Simple, Energy-Optimal Disturbance Rejection,” *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2012.
- [58] V. Vasilopoulos, I. S. Paraskevas, and E. G. Papadopoulos, “Compliant terrain legged locomotion using a viscoplastic approach,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.
- [59] X. Xiong, A. D. Ames, and D. I. Goldman, “A stability region criterion for flat-footed bipedal walking on deformable granular terrain,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [60] C. Li, T. Zhang, and D. I. Goldman, “A Terradynamics of Legged Locomotion on Granular Media,” *Science*, vol. 339, no. 6126, pp. 1408–1412, 2013.
- [61] S. Taheri, C. Sandu, S. Taheri, E. Pinto, and D. Gorsich, “A technical survey on Terramechanics models for tire-terrain interaction used in modeling and simulation of wheeled vehicles,” *Journal of Terramechanics*, vol. 57, pp. 1–22, 2015.
- [62] M. Bauza and A. Rodriguez, “A probabilistic data-driven model for planar pushing,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [63] N. Fazeli, E. Drumwright, S. Zapolsky, and A. Rodriguez, “Learning Data-Efficient Rigid-Body Contact Models : Case Study of Planar Impact,” *Conference on Robot Learning (CoRL)*, 2017.
- [64] H. Wang, J. F. O’Brien, and R. Ramamoorthi, “Data-Driven Elastic Models for Cloth: Modeling and Measurement,” *ACM Transactions on Graphics*, vol. 30, no. 4, p. 1, 2011.
- [65] B. Bickel, M. Bächer, M. A. Otaduy, W. Matusik, H. Pfister, and M. Gross, “Capture and modeling of non-linear heterogeneous soft tissue,” in *SIGGRAPH*, 2009.
- [66] T. A. Brzinski, P. Mayor, and D. J. Durian, “Depth-Dependent Resistance of Granular Media to Vertical Penetration,” *Physical Review Letters*, vol. 111, no. 16, p. 168002, oct 2013.
- [67] I. Albert, P. Tegzes, B. Kahng, R. Albert, J. G. Sample, M. Pfeifer, A. L. Barabási, T. Vicsek, and P. Schiffer, “Jamming and fluctuations in granular drag,” *Physical Review Letters*, vol. 84, no. 22, pp. 5122–5125, 2000.

- [68] R. Albert, M. A. Pfeifer, A. L. Barabási, and P. Schiffer, “Slow drag in a granular medium,” *Physical Review Letters*, vol. 82, no. 1, pp. 205–208, jan 1999.
- [69] K. Hauser, S. Wang, and M. R. Cutkosky, “Efficient Equilibrium Testing Under Adhesion and Anisotropy Using Empirical Contact Force Models,” *IEEE Transactions on Robotics*, vol. PP, pp. 1–13, 2018.
- [70] J. Kozicki and F. Donze, “YADE-OPEN DEM: an open-source software using a discrete element method to simulate granular material,” *Engineering Computations*, vol. 26, no. 7, pp. 786–805, 2009.
- [71] E. Drumwright and D. A. Shell, “Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation,” *Springer Tracts in Advanced Robotics*, vol. 68, no. STAR, pp. 249–266, 2010.
- [72] D. Baraff, “Fast Contact Force Computation for Nonpenetrating Rigid Bodies,” in *SIGGRAPH*, 1994.
- [73] M. Anitescu and F. Potra, “Formulating Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems,” *ASME Journal of Nonlinear Dynamics*, vol. 14, no. 93, pp. 231–247, 1997.
- [74] B. Henze, M. A. Roa, and C. Ott, “Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios,” *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.
- [75] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, “STANCE: Locomotion Adaptation over Soft Terrain,” *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 443–457, 2020.
- [76] J. T. Betts, “Survey of Numerical Methods for Trajectory Optimization,” *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [77] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [78] G. Schultz and K. Mombaur, “Modeling and optimal control of human-like running,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 783–792, 2010.
- [79] M. Posa, S. Kuindersma, and R. Tedrake, “Optimization and stabilization of trajectories for constrained dynamical systems,” in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2016-June. IEEE, 2016, pp. 1366–1373.
- [80] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [81] A. Patel, S. L. Shield, S. Kazi, A. M. Johnson, and L. T. Biegler, “Contact-implicit trajectory optimization using orthogonal collocation,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2242–2249, 2019.

- [82] B. Landry, J. Lorenzetti, Z. Manchester, and M. Pavone, “Bilevel Optimization for Planning through Contact: A Semidirect Method,” *arXiv:1906.04292*, 2019.
- [83] J. Carius, R. Ranftl, V. Koltun, and M. Hutter, “Trajectory optimization with implicit hard contacts,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3316–3323, 2018.
- [84] I. Chatzinikolaïdis, Y. You, and Z. Li, “Contact-Implicit Trajectory Optimization Using an Analytically Solvable Contact Model for Locomotion on Variable Ground,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6357–6364, 2020.
- [85] D. E. Stewart and J. C. Trmkle, “An Implicit Time-stepping Scheme for Rigid Body Dynamics with Inelastic Collisions and Coulomb Friction,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. September 1995, 1996.
- [86] Y. Tassa, T. Erez, and E. Todorov, “Synthesis and stabilization of complex behaviors through online trajectory optimization,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, no. October 2012, 2012.
- [87] A. Sinha, P. Malo, and K. Deb, “A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2018.
- [88] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, “An efficient optimal planning and control framework for quadrupedal locomotion,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [89] G. Tang, W. Sun, and K. Hauser, “Time-Optimal Trajectory Generation for Dynamic Vehicles: A Bilevel Optimization Approach,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.
- [90] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press, Mar 1983.
- [91] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Zico Kolter, “Differentiable convex optimization layers,” in *Advances in Neural Information Processing Systems*, no. NeurIPS, 2019.
- [92] R. H. Byrd, J. Nocedal, and R. A. Waltz, “Knitro: An Integrated Package for Nonlinear Optimization,” *Large-scale nonlinear optimization*, pp. 35–59, 2006.
- [93] T. M. Lejeune, P. A. Willems, and N. C. Heglund, “Mechanics and energetics of human locomotion on sand.” *Journal of Experimental Biology*, vol. 201, no. 13, pp. 2071–2080, 1998.
- [94] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *arXiv:1707.06347*, 2017.

- [95] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, “Trust region policy optimization,” in *International Conference on Machine Learning (ICML)*, 2015.
- [96] A. Oishi and G. Yagawa, “Computational Mechanics Enhanced by Deep Learning,” *Computational Methods in Applied Mechanics and Engineering*, vol. 327, pp. 327–351, 2017.
- [97] A. Nealen, M. Matthias, R. Keiser, E. Boxerman, and M. Carlson, “Physically Based Deformable Models in Computer Graphics,” *Computer Graphics Forum*, vol. 25, no. 4, pp. 809–836, 2006.
- [98] B. Frank, R. Schemedding, C. Stachniss, M. Teschner, and W. Burgard, “Learning the elasticity parameters of deformable objects with a manipulation robot,” in *IEEE/RSJ International Conference Intelligent Robots Systems (IROS)*, 2010.
- [99] P. Boonvisut and M. C. Cavusoglu, “Estimation of Soft Tissue Mechanical Parameters From Robotic Manipulation Data,” *Transactions on Mechatronics*, vol. 18, no. 5, pp. 1602–1611, 2013.
- [100] O. Deussen, L. Kobbelt, and T. Peter, “Using Simulated Annealing to Obtain Good Nodal Approximations of Deformable Bodies,” in *Eurographics Workshop*, 1995.
- [101] S. Burion, F. Conti, A. Petrovskaya, C. Baur, and O. Khatib, “Identifying Physical Properties of Deformable Objects by Using Particle Filters,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
- [102] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura, “Tissue Property Estimation and Graphical Display for Teleoperated Robot-Assisted Surgery,” in *International Conference on Robotics and Automation (ICRA)*, 2009.
- [103] H. Salman, E. Ayvali, R. A. Srivatsan, Y. Ma, N. Zevallos, R. Yasin, L. Wang, N. Siman, and H. Choset, “Trajectory-Optimized Sensing for Active Search of Tissue Abnormalities in Robotic Surgery,” in *IEEE International Conference Robots Aut.*, 2018.
- [104] R. E. Goldman, A. Bajo, and N. Simaan, “Algorithms for autonomous exploration and estimation in compliant environments,” *Robotica*, vol. 31, no. 1, pp. 71–87, jan 2013.
- [105] H. Liu, D. P. Noonan, B. J. Challacombe, P. Dasgupta, L. D. Seneviratne, and K. Althoefer, “Rolling Mechanical Imaging for Tissue Abnormality Localization During Minimally Invasive Surgery,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 404–414, 2010.
- [106] Q.-y. Zhou, J. Park, and V. Koltun, “Open3D : A Modern Library for 3D Data Processing,” *arXiv:1801.0984*, 2018.
- [107] M. Feurer, A. Klein, K. Eggenberger, J. Springenberg, M. Blum, and F. Hutter, “Efficient and Robust Automated Machine Learning,” in *Advances in Neural Information Processing Systems*, 2015.

- [108] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn : Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [109] R. Weller, “A Brief Overview of Collision Detection,” in *New Geometric Data Structures for Collision Detection and Haptics*, 2013.
- [110] A. Khamis, J. Meng, J. Wang, A. T. Azar, E. Prestes, H. Li, I. A. Hameed, Á. Takács, I. J. Rudas, and T. Haidegger, “Robotics and intelligent systems against a pandemic,” *Acta Polytechnica Hungarica*, vol. 18, no. 5, pp. 13–35, 2021.
- [111] M. Giuliani, D. Szczęśniak-Stańczyk, N. Mirnig, G. Stollnberger, M. Szyszko, B. Stańczyk, and M. Tscheligi, “User-centred design and evaluation of a tele-operated echocardiography robot,” *Health and Technology*, vol. 10, no. 3, pp. 649–665, 2020.
- [112] G. Yang, H. Lv, Z. Zhang, L. Yang, J. Deng, S. You, J. Du, and H. Yang, “Keep Healthcare Workers Safe: Application of Teleoperated Robot in Isolation Ward for COVID-19 Prevention and Control,” *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 33, no. 1, 2020.
- [113] Z. Li, P. Moran, Q. Dong, R. J. Shaw, and K. Hauser, “Development of a tele-nursing mobile manipulator for remote care-giving in quarantine areas,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [114] K. Arent, M. Cholewiński, W. Domski, M. Drwiega, J. Jakubiak, M. Janiak, B. Kreczmer, A. Kurnicki, B. Stańczyk, D. Szczęśniak-Stańczyk et al., “Selected topics in design and application of a robot for remote medical examination with the use of ultrasonography and auscultation from the perspective of the remedi project,” *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 11, no. 2, pp. 82–94, 2017.
- [115] R. Kim, J. Schloen, N. Campbell, S. Horton, V. Zderic, I. Efimov, D. Lee, and C. H. Park, “Robot-Assisted Semi-Autonomous Ultrasound Imaging with Tactile Sensing and Convolutional Neural-Networks,” *IEEE Transactions Medical Robotics and Bionics*, vol. 3, no. 1, pp. 96–105, 2021.
- [116] G. Yang, H. Lv, Z. Zhang, L. Yang, J. Deng, S. You, J. Du, and H. Yang, “Keep Healthcare Workers Safe: Application of Teleoperated Robot in Isolation Ward for COVID-19 Prevention and Control,” *Chinese Journal of Mechanical Engineering*, vol. 33, no. 1, 2020.
- [117] X. V. Wang and L. Wang, “A literature survey of the robotic technologies during the COVID-19 pandemic,” *Journal of Manufacturing Systems*, Feb 2021.
- [118] A. Di Lallo, R. Murphy, A. Krieger, J. Zhu, R. H. Taylor, and H. Su, “Medical Robots for Infectious Diseases: Lessons and Challenges from the COVID-19 Pandemic,” *IEEE Robotics and Automation Magazine*, 2021.

- [119] K. D. Evans, Q. Yang, Y. Liu, R. Ye, and C. Peng, “Sonography of the Lungs: Diagnosis and Surveillance of Patients With COVID-19,” *Journal of Diagnostic Medical Sonography*, vol. 36, no. 4, pp. 370–376, 2020.
- [120] S. J. Adams, B. Burbridge, L. Chatterson, V. McKinney, P. Babyn, and I. Mendez, “Telerobotic ultrasound to provide obstetrical ultrasound services remotely during the COVID-19 pandemic,” *Journal of Telemedicine and Telecare*, 2020.
- [121] R. Z. Yu, Y. Q. Li, C. Z. Peng, R. Z. Ye, and Q. He, “Role of 5G-powered remote robotic ultrasound during the COVID-19 outbreak: Insights from two cases,” *European Review for Medical and Pharmacological Sciences*, vol. 24, no. 14, pp. 7796–7800, 2020.
- [122] J. Wang, C. Peng, Y. Zhao, R. Ye, J. Hong, H. Huang, and L. Chen, “Application of a Robotic Tele-Echography System for COVID-19 Pneumonia,” *Journal on Ultrasound in Medicine*, vol. 40, pp. 385–390, 2021.
- [123] B. Mathur, A. Topiwala, S. Schaffer, M. Kam, H. Saeidi, T. Fleiter, and A. Krieger, “A semi-autonomous robotic system for remote trauma assessment,” in *IEEE 19th International Conference on Bioinformatics and Bioengineering*, 2019.
- [124] T. Haidegger, “Autonomy for Surgical Robots: Concepts and Paradigms,” *IEEE Transactions on Medical Robotics and Bionics*, vol. 1, no. 2, pp. 65–76, 2019.
- [125] D. Driess, D. Hennes, and M. Toussaint, “Active multi-contact continuous tactile exploration with gaussian process differential entropy,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [126] E. Ayvali, A. Ansari, L. Wang, N. Simaan, and H. Choset, “Utility-Guided Palpation for Locating Tissue Abnormalities,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 864–871, 2017.
- [127] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, “Active planning for underwater inspection and the benefit of adaptivity,” *International Journal of Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.
- [128] G. Hollinger and G. Sukhatme, “Sampling-based Motion Planning for Robotic Information Gathering,” in *Robotics: Science and Systems (RSS)*, vol. 3, no. 5, 2013.
- [129] J. Binney and G. S. Sukhatme, “Branch and bound for informative path planning,” in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2012, pp. 2147–2154.
- [130] R. Camoriano, S. Traversaro, L. Rosasco, G. Metta, and F. Nori, “Incremental semi-parametric inverse dynamics learning,” in *IEEE International Conference on Robotics and Automation Conference on Robotics and Automation (ICRA)*, 2016.
- [131] E. Brochu, V. M. Cora, and N. de Freitas, “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning,” *arXiv:1012.2599*, 2010.

- [132] J. M. Vukanovic-Criley, S. Criley, C. M. Warde, J. R. Boker, L. Guevara-Matheus, W. H. Churchill, W. P. Nelson, and J. M. Criley, “Competency in cardiac examination skills in medical students, trainees, physicians, and faculty: a multicenter study,” *Archives of internal medicine*, vol. 166, no. 6, pp. 610–616, 2006.
- [133] E. Grooby, J. He, J. Kiewsky, D. Fattahi, L. Zhou, A. King, A. Ramanathan, A. Malhotra, G. A. Dumont, and F. Marzbanrad, “Neonatal Heart and Lung Sound Quality Assessment for Robust Heart and Breathing Rate Estimation for telehealth Applications,” *IEEE Journal of Biomedical and Health Informatics*, vol. 2194, no. c, pp. 1–12, 2020.
- [134] T. K. Koo and M. Y. Li, “A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research,” *Journal of Chiropractic Medicine*, vol. 15, no. 2, pp. 155–163, 2016.
- [135] T. Sainburg, M. Thielk, and T. Q. Gentner, “Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires,” *PLoS computational biology*, vol. 16, no. 10, 2020.
- [136] T. T. Le, W. Fu, and J. H. Moore, “Scaling tree-based automated machine learning to biomedical big data with a feature set selector,” *Bioinformatics*, vol. 36, no. 1, pp. 250–256, 2020.
- [137] M. A. Fischler and R. C. Bolles, “Random Sample Paradigm for Model Consensus: A Apphcatlons to Image Fitting with Analysis and Automated Cartography,” *Graphics and Image Processing*, vol. 24, no. 6, pp. 381–395, 1981.
- [138] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, apr 1992, pp. 586–606.
- [139] B. Amberg, S. Romdhani, and T. Vetter, “Optimal Step Nonrigid ICP Algorithms for Surface Registration,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
- [140] K. M. Nowak and L. Nowak, “On the relation between pressure applied to the chest piece of a stethoscope and parameters of the transmitted bioacoustic signals,” in *The 22nd Internation Congress on Acoustics*, 2016.
- [141] Y. Yang, Y. Yu, Y. Zhou, S. Du, J. Davis, and R. Yang, “Semantic Parametric Reshaping of Human Body Models,” in *2014 International Conference on 3D Vision Workshops*, 2015, pp. 41–48.
- [142] W. Haynes, “Tukey’s test,” *Encyclopedia of Systems Biology*, pp. 2303–2304, 2013.
- [143] A. Damianou and N. D. Lawrence, “Deep Gaussian Processes,” in *International Conference on Artificial Intelligence and Statistics*, 2013.
- [144] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

- [145] E. V. Bonilla, K. Chai, and C. Williams, “Multi-task Gaussian process prediction,” in *Conference on Neural Information Processing Systems*, 2007.
- [146] G. Skolidis, “Transfer learning with Gaussian processes,” Ph.D. dissertation, University of Edinburgh, 2012.
- [147] V. Fortuin, H. Strathmann, and G. Rätsch, “Meta-learning mean functions for Gaussian processes,” in *NeurIPS workshop on Bayesian Deep Learning*, 2019.
- [148] J. Rothfuss, V. Fortuin, M. Josifoski, and A. Krause, “PACOH: Bayes-optimal meta-learning with PAC-guarantees,” in *International Conference on Machine Learning (ICML)*, 2021.
- [149] J. Huang, S. Zhang, C. Tao, F. Yang, C. Yan, D. Zhou, and X. Zeng, “Bayesian optimization approach for analog circuit design using multi-task Gaussian process,” in *IEEE International Symposium on Circuits and Systems*, 2021, pp. 1–5.
- [150] Z. Wang, B. Kim, and L. P. Kaelbling, “Regret bounds for meta Bayesian optimization with an unknown Gaussian process prior,” in *Conference on Neural Information Processing Systems*, 2018.
- [151] S. Hochreiter, A. S. Younger, and P. R. Conwell, “Learning to learn using gradient descent,” in *International Conference on Artificial Neural Networks*, 2001.
- [152] D. Ha, A. M. Dai, and Q. V. Le, “HyperNetworks,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [153] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [154] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” in *International Conference on Learning Representations (ICLR)*, 2019.
- [155] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Conference on Neural Information Processing Systems*, 2017.
- [156] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra, “Matching networks for one shot learning,” in *Conference on Neural Information Processing Systems*, 2016.
- [157] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International Conference on Machine Learning*, vol. 70, 2017, pp. 1126–1135.
- [158] Z. Mandi, P. Abbeel, and S. James, “On the effectiveness of fine-tuning versus meta-reinforcement learning,” *arXiv:2206.03271*, 2022.
- [159] A. Ballou, X. Alameda-Pineda, and C. Reinke, “Variational meta reinforcement learning for social robotics,” *arXiv:2206.03211*, 2022.

- [160] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *ICML Deep Learning Workshop*, 2015.
- [161] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap, “One-shot learning with memory-augmented neural networks,” in *NIPS Deep Learning Symposium*, 2016.
- [162] S. Bartunov and D. P. Vetrov, “Fast adaptation in generative models with generative matching networks,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [163] A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu, O. Vinyals, and A. Graves, “Conditional image generation with PixelCNN decoders,” in *Conference on Neural Information Processing Systems*, 2016.
- [164] S. E. Reed, Y. Chen, T. Paine, A. van den Oord, S. M. A. Eslami, D. J. Rezende, O. Vinyals, and N. de Freitas, “Few-shot autoregressive density estimation: towards learning to learn distributions,” in *International Conference on Learning Representations (ICLR)*, 2018.
- [165] J. Bornschein, A. Mnih, D. Zoran, and D. Jimenez Rezende, “Variational memory addressing in generative models,” in *Conference on Neural Information Processing Systems*, 2017.
- [166] D. J. Rezende, S. Mohamed, I. Danihelka, K. Gregor, and D. Wierstra, “One-shot generalization in deep generative models,” in *International Conference on Machine Learning*, 2016.
- [167] M. Gauch, M. Beck, T. Adler, D. Kotsur, S. Fiel, H. Eghbal-zadeh, J. Brandstetter, J. Kofler, M. Holzleitner, W. Zellinger, D. Klotz, S. Hochreiter, and S. Lehner, “Few-Shot learning by dimensionality reduction in gradient space,” in *Conference on Lifelong Learning Agents*, 2022, pp. 1043–1064.
- [168] Z. Li, F. Zhou, F. Chen, and H. Li, “Meta-SGD: Learning to learn quickly for few-shot learning,” *arXiv:1707.09835*, 2017.
- [169] K. Hand, A. Murray, J. Garvin, W. Brinckerhoff, B. Christner, K. Edgett, and T. Hoehler, “Report of the Europa lander science definition team,” NASA, Tech. Rep., 2017.
- [170] Q. Lu and L. Zhang, “Excavation learning for rigid objects in clutter,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7373–7380, 2021.
- [171] C. Schenck, J. Tompson, S. Levine, and D. Fox, “Learning robotic manipulation of granular media,” in *Conference on Robot Learning (CoRL)*, 2017.
- [172] S. Dadhich, U. Bodin, and U. Andersson, “Key challenges in automation of earth-moving machines,” *Automation in Construction*, vol. 68, pp. 212–222, 2016.

- [173] H. J. T. Suh and R. Tedrake, “The surprising effectiveness of linear models for visual foresight in object pile,” in *Workshop on the Algorithmic Foundations of Robotics*, 2021, pp. 347–363.
- [174] K. Takahashi, W. Ko, A. Ummadisingu, and S.-i. Maeda, “Uncertainty-aware self-supervised target-mass grasping of granular foods,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2620–2626.
- [175] P. Ray and M. J. Howard, “Robotic untangling of herbs and salads with parallel grippers,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [176] S. Shrivastava, A. Karsai, Y. O. Aydin, R. Pettinger, W. Bluethmann, R. O. Ambrose, and D. I. Goldman, “Material remodeling and unconventional gaits facilitate locomotion of a robophysical rover over granular terrain,” *Science Robotics*, vol. 5, 2020.
- [177] A. Karsai, D. Kerimoglu, D. Soto, S. Ha, T. Zhang, and D. I. Goldman, “Real-Time Remodeling of Granular Terrain for Robot Locomotion,” *Advanced Intelligent Systems*, vol. 4, 2022.
- [178] Y. Yang, P. Long, X. Song, J. Pan, and L. Zhang, “Optimization-based framework for excavation trajectory generation,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1479–1486, 2021.
- [179] S. Dadhich, U. Bodin, and U. Andersson, “Automation in Construction Key challenges in automation of earth-moving machines,” *Automation in Construction*, vol. 68, pp. 212–222, 2016.
- [180] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, “Conditional neural processes,” in *International Conference on Machine Learning (ICML)*, 2018.
- [181] S. Sing, “Synthesis of tactical plans for robotic excavation,” Ph.D. dissertation, Carnegie Mellon University, 1995.
- [182] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An imperative style, high-performance deep learning library,” in *Conference on Neural Information Processing Systems*, 2019.
- [183] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, “GPY-Torch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration,” in *Conference on Neural Information Processing Systems*, 2018.

- [184] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [185] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems*, 2017.
- [186] S. Marsh and D. Fosbroke, “Trends of occupational fatalities involving machines, United States, 1992-2010,” *American Journal of Industrial Medicine*, 2015.
- [187] A. A. Dobson, J. A. Marshall, and J. Larsson, “Admittance control for robotic loading: Design and experiments with a 1-tonne loader and a 14-tonne load-haul-dump machine,” *Journal of Field Robotics*, vol. 34, no. 1, pp. 123–150, 2017.
- [188] H. Fernando, J. A. Marshall, and J. Larsson, “Iterative learning-based admittance control for autonomous excavation,” *Journal of Intelligent & Robotic Systems*, vol. 96, no. 3, pp. 493–500, 2019.
- [189] S. Dadhich, U. Bodin, F. Sandin, and U. Andersson, “Machine learning approach to automatic bucket loading,” in *24th Mediterranean Conference on Control and Automation, MED 2016*, 2016.
- [190] O. Azulay and A. Shapiro, “Wheel Loader Scooping Controller Using Deep Reinforcement Learning,” *IEEE Access*, vol. 9, pp. 24 145–24 154, 2021.
- [191] Q. Lu, Y. Zhu, and L. Zhang, “Excavation Reinforcement Learning Using Geometric Representation,” *IEEE Robotics and Automation Letters*, 2022.
- [192] P. Egli and M. Hutter, “Towards RL-Based Hydraulic Excavator Automation,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [193] G. J. Maeda, I. R. Manchester, and D. C. Rye, “Combined ILC and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1754–1769, 2015.
- [194] F. E. Sotiropoulos and H. H. Asada, “A model-free extremum-seeking approach to autonomous excavator control based on output power maximization,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1005–1012, 2019.
- [195] R. J. Sandzimier and H. H. Asada, “A Data-Driven Approach to Prediction and Optimal Bucket-Filling Control for Autonomous Excavators,” *IEEE Robots and Automation Letters*, vol. 5, no. 2, pp. 2682–2689, 2020.
- [196] D. Jud, G. Hottiger, P. Leemann, and M. Hutter, “Planning and Control for Autonomous Excavation,” *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2151–2158, 2017.

- [197] J. Seo, S. Lee, J. Kim, and S. K. Kim, “Task planner design for an automated excavation system,” *Automation in Construction*, vol. 20, no. 7, pp. 954–966, 2011.
- [198] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, “An autonomous excavator system for material loading tasks,” *Science Robotics*, vol. 6, no. 55, jun 2021.
- [199] D. Kappler, P. Pastor, M. Kalakrishnan, M. Wüthrich, and S. Schaal, “Data-Driven Online Decision Making for Autonomous Manipulation,” in *Robotics: Science and Systems (RSS)*, 2015.
- [200] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, and S. Levine, “More than a feeling: Learning to grasp and regrasp using vision and touch,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3300–3307, oct 2018.
- [201] M. A. Lee, Y. Zhu, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, A. Garg, and J. Bohg, “Making Sense of Vision and Touch: Learning Multimodal Representations for Contact-Rich Tasks,” *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.
- [202] Y. Liu, D. Romeres, D. K. Jha, and D. Nikovski, “Understanding Multi-Modal Perception Using Behavioral Cloning for Peg-In-a-Hole Insertion Tasks,” *arXiv:2007.11646*, jul 2020.
- [203] M. V. Balakuntala, U. Kaur, X. Ma, J. Wachs, and R. M. Voyles, “Learning Multimodal Contact-Rich Skills from Demonstrations Without Reward Engineering,” in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2021.
- [204] Z. Wu, W. Lian, V. Unhelkar, M. Tomizuka, and S. Schaal, “Learning Dense Rewards for Contact-Rich Manipulation Tasks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2021.
- [205] F. Yi, W. Fu, and H. Liang, “Model-based Reinforcement Learning: A Survey,” in *Int’l Conference on Electronic Business*, vol. 2018-Decem. International Consortium for Electronic Business, jun 2020, pp. 421–429.
- [206] T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba, “Benchmarking Model-Based Reinforcement Learning,” *arXiv:1907.02057*, jul 2019.
- [207] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [208] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

- [209] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Conference on Empirical Methods in Natural Language Processing*, 2014.
- [210] D. Bahdanau, K. H. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *International Conference on Learning Representations (ICLR)*, pp. 1–15, 2015.
- [211] I. Lenz, R. Knepper, and A. Saxena, “DeepMPC: Learning deep latent features for model predictive control,” *Robotics: Science and Systems (RSS)*, vol. 11, 2015.
- [212] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Conference on Computer Vision and Pattern Recognition*, 2010.
- [213] J. V. Draper, D. B. Kaber, and J. M. Usher, “Telepresence,” *Human Factors*, vol. 40, no. 3, pp. 354–375, 1998.
- [214] S. Tachi, K. Minamizawa, M. Furukawa, and C. L. Fernando, “Telexistence — from 1980 to 2012,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [215] S. Mehrdad, F. Liu, M. T. Pham, A. Lelevé, and S. F. Atashzar, “Review of Advanced Medical Telerobots,” *Applied Sciences*, vol. 11, no. 1, 2021.
- [216] L. Qian, A. Deguet, and P. Kazanzides, “ARssist: augmented reality on a head-mounted display for the first assistant in robotic surgery,” *Healthcare Technology Letters*, vol. 5, no. 5, pp. 194–200, 2018.
- [217] T. Klamt, M. Schwarz, C. Lenz, L. Baccelliere, D. Buongiorno, T. Cichon, A. Di-Guardo, D. Droschel, M. Gabardi, M. Kamedula, N. Kashiri, A. Laurenzi, D. Leonardis, L. Muratore, D. Pavlichenko, A. S. Periyasamy, D. Rodriguez, M. Sollazzi, A. Frisoli, M. Gustmann, J. Roßmann, U. Süß, N. G. Tsagarakis, and S. Behnke, “Remote mobile manipulation with the centauro robot: Full-body telepresence and autonomous operator assistance,” *Journal of Field Robotics*, vol. 37, no. 5, pp. 889–919, 2020.
- [218] T. Klamt, D. Rodriguez, M. Schwarz, C. Lenz, D. Pavlichenko, D. Droschel, and S. Behnke, “Supervised Autonomous Locomotion and Manipulation for Disaster Response with a Centaur-Like Robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [219] N. Y. Lii, D. Leidner, P. Birkenkamp, B. Pleintinger, R. Bayer, and T. Krueger, “Toward scalable intuitive telecommand of robots for space deployment with METERON SUPVIS Justin,” in *The 14th Symposium on Advanced Space Technologies for Robotics and Automation (ASTRA)*, 2017.

- [220] J. O. Burns, D. A. Kring, J. B. Hopkins, S. Norris, T. J. W. Lazio, and J. Kasper, “A lunar L2-Farside exploration and science mission concept with the Orion Multi-Purpose Crew Vehicle and a teleoperated lander/rover,” *Advances in Space Research*, vol. 52, no. 2, pp. 306–320, 2013.
- [221] G. G. Muscolo, S. Marcheschi, M. Fontana, and M. Bergamasco, “Dynamics Modeling of Human–Machine Control Interface for Underwater Teleoperation,” *Robotica*, vol. 39, no. 4, p. 618–632, 2021.
- [222] J. Avalos and O. E. Ramos, “Real-time teleoperation with the baxter robot and the kinect sensor,” in *IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, 2017.
- [223] J. Orlosky, K. Theofilis, K. Kiyokawa, and Y. Nagai, “Effects of Throughput Delay on Perception of Robot Teleoperation and Head Control Precision in Remote Monitoring Tasks,” *Presence: Teleoperators and Virtual Environments*, vol. 27, no. 2, pp. 226–241, 02 2018.
- [224] C. Yang, J. Zhang, Y. Chen, Y. Dong, and Y. Zhang, “A review of exoskeleton-type systems and their key technologies,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 222, no. 8, pp. 1599–1612, 2008.
- [225] A. Wang, J. Ramos, J. Mayo, W. Ubellacker, J. Cheung, and S. Kim, “The hermes humanoid system: A platform for full-body teleoperation with balance feedback,” in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015.
- [226] Toyota, “Toyota unveils third generation humanoid robot t-hr3,” 2017. [Online]. Available: <https://global.toyota/en/detail/19666346>
- [227] E. Krotkov, D. Hackett, L. Jackel, M. Perschbacher, J. Pippine, J. Strauss, G. Pratt, and C. Orłowski, “The DARPA Robotics Challenge Finals: Results and Perspectives,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 229–240, 2017.
- [228] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburg, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt, “Team IHMC’s Lessons Learned from the DARPA Robotics Challenge Trials,” *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.
- [229] J. Lim, I. Lee, I. Shim, H. Jung, H. M. Joe, H. Bae, O. Sim, J. Oh, T. Jung, S. Shin, K. Joo, M. Kim, K. Lee, Y. Bok, D.-G. Choi, B. Cho, S. Kim, J. Heo, I. Kim, J. Lee, I. S. Kwon, and J.-H. Oh, “Robot System of DRC-HUBO+ and Control Strategy of Team KAIST in DARPA Robotics Challenge Finals,” *Journal of Field Robotics*, vol. 34, no. 4, pp. 802–829, 2017.

- [230] S. Karumanchi, K. Edelberg, I. Baldwin, J. Nash, J. Reid, C. Bergh, J. Leichty, K. Carpenter, M. Shekels, M. Gildner, D. Newill-Smith, J. Carlton, J. Koehler, T. Dobrev, M. Frost, P. Hebert, J. Borders, J. Ma, B. Douillard, P. Backes, B. Kennedy, B. Satzinger, C. Lau, K. Byl, K. Shankar, and J. Burdick, “Team RoboSimian: Semi-autonomous Mobile Manipulation at the 2015 DARPA Robotics Challenge Finals,” *Journal of Field Robotics*, vol. 34, no. 2, pp. 305–332, 3 2017.
- [231] M. Schwarz, M. Beul, D. Droschel, T. Klamt, C. Lenz, D. Pavlichenko, T. Rodehutsors, M. Schreiber, N. Araslanov, I. Ivanov, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke, “DRC Team NimbRo Rescue: Perception and Control for Centaur-Like Mobile Manipulation Robot Momaro,” in *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*. Cham: Springer International Publishing, 2018, pp. 145–190.
- [232] V. Nguyen, “Increasing Independence with Stretch: A Mobile Robot Enabling Functional Performance in Daily Activities,” 2023.
- [233] K. Takeuchi, Y. Yamazaki, and K. Yoshifuji, “Avatar Work: Telework for Disabled People Unable to Go Outside by Using Avatar Robots,” in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. New York, NY, USA: Association for Computing Machinery, 2020, p. 53–60.
- [234] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, “The Design of Stretch: A Compact, Lightweight Mobile Manipulator for Indoor Human Environments,” in *International Conference on Robotics and Automation (ICRA)*, 2022.
- [235] S. Tachi, Y. Inoue, and F. Kato, “TELESAR VI: Telexistence Surrogate Anthropomorphic Robot VI,” *International Journal of Humanoid Robotics*, vol. 17, no. 05, p. 2050019, 2020.
- [236] M. Schwarz, C. Lenz, R. Memmesheimer, B. Pätzold, A. Rochow, M. Schreiber, and S. Behnke, “Robust Immersive Telepresence and Mobile Telemanipulation: NimbRo wins ANA Avatar XPRIZE Finals,” *arXiv:2303.03297v1*, 3 2023.
- [237] R. Luo, C. Wang, C. Keil, D. Nguyen, H. Mayne, S. Alt, E. Schwarm, E. Mendoza, s. P. Ta and J. P. Whitney, “Team northeastern’s approach to ana xprize avatar final testing: A holistic approach to telepresence and lessons learned,” *ArXiv*, 2023.
- [238] J. M. C. Marques, P. Naughton, Y. Zhu, N. Malhotra, and K. Hauser, “Commodity Telepresence with the AvaTRINA Nursebot in the ANA Avatar XPRIZE Semifinals,” in *RSS 2022 Workshop on “Towards Robot Avatars: Perspectives on the ANA Avatar XPRIZE Competition*, 2022.
- [239] B. Pätzold, A. Rochow, M. Schreiber, R. Memmesheimer, C. Lenz, M. Schwarz, and S. Behnke, “Audio-based Roughness Sensing and Tactile Feedback for Haptic Perception in Telepresence,” *arXiv:2303.07186v1*, 2023.

- [240] L. Carlone and C. Pinciroli, “Robot Co-design: Beyond the Monotone Case,” in *International Conference on Robotics and Automation (ICRA)*, 2019.
- [241] P. K. Jamwal, S. Xie, and K. C. Aw, “Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm,” *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1018–1027, 2009.
- [242] G. Marani, J. Kim, J. Yuh, and W. K. Chung, “A real-time approach for singularity avoidance in resolved motion rate control of robotic manipulators,” *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1973–1978, 2002.
- [243] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sutan, and K. Salisbury, “Methods for collision-free arm teleoperation in clutter using constraints from 3d sensor data,” in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013.
- [244] K. Hauser, “Robust Contact Generation for Robot Simulation with Unstructured Meshes,” in *The International Symposium on Robotics Research (ISRR)*, 2016.
- [245] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, “A rewriting system for convex optimization problems,” *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.
- [246] K. Hauser, E. Watson, J. Bae, J. Bankston, S. Behnke, B. Borgia, M. Catalano, S. Daffarra, J. van Erp, T. Ferris, J. Fishel, G. Hoffman, S. Ivaldi, F. Kanehiro, A. Kheddar, G. Lannuzel, J. Morie, P. Naughton, S. NGuyen, P. Oh, T. Padir, J. Pippine, J. Park, D. Pucci, J. Vaz, P. Whitney, P. Wu, and D. Locke, “Analysis and perspectives on the ana avatar xprize competition,” *International Journal of Social Robotics*, 2023 (submitted).