

# Autonomous Learning for Industrial Manipulation: Enhancing Grasping and Insertion Tasks through Scalable Data Collection

*Letian Fu*



Electrical Engineering and Computer Sciences  
University of California, Berkeley

Technical Report No. UCB/EECS-2023-107

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2023/EECS-2023-107.html>

May 11, 2023

Copyright © 2023, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Autonomous Learning for Industrial Manipulation:  
Enhancing Grasping and Insertion Tasks through Scalable Data Collection

by

Letian Fu

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair  
Professor Jitendra Malik

Spring 2023

---

**Autonomous Learning for Industrial Manipulation:  
Enhancing Grasping and Insertion Tasks through Scalable Data Collection**

by Letian Fu

---

**Research Project**

Submitted to the Department of Electrical Engineering and Computer Sciences,  
University of California at Berkeley, in partial satisfaction of the requirements for the  
degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

**Committee:**



---

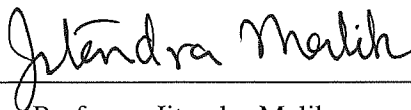
Professor Ken Goldberg  
Research Advisor

11 May 2023

---

(Date)

\*\*\*\*\*



---

Professor Jitendra Malik  
Second Reader

11 May 2023

---

(Date)

Autonomous Learning for Industrial Manipulation:  
Enhancing Grasping and Insertion Tasks through Scalable Data Collection

Copyright 2023  
by  
Letian Fu

Abstract

Autonomous Learning for Industrial Manipulation:  
Enhancing Grasping and Insertion Tasks through Scalable Data Collection

by

Letian Fu

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

Grasping and insertion represent two fundamental skills for robots, garnering significant interest within the robotics community due to their widespread applications in fields such as manufacturing, logistics, maintenance, and repair. Although numerous studies have demonstrated success in both tasks, several challenges persist. For instance, general-purpose, learning-based grasping systems often struggle to identify optimal grasps for novel, out-of-distribution industrial components, necessitating manual predefinition by humans. Likewise, many learning-based insertion algorithms require extensive demonstrations from human teleoperators and assume fixed grasp poses with minimal rotation, limiting their adaptability. Addressing these limitations, we study the problem of grasp identification and industrial insertion through two different learning-based approaches that can directly operate the physical robot with minimal human interventions, both of which achieved better performance than baselines in their respective tasks.

To parents and friends,

# Contents

<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Related Works</b>	<b>3</b>
2.1 Universal Grasping Algorithms . . . . .	3
2.2 Multi-Armed Bandits . . . . .	3
2.3 Exploratory Grasping . . . . .	4
2.4 Industrial Insertion . . . . .	4
2.5 Tactile Sensing for Industrial Insertion . . . . .	5
2.6 Multi-Modal Learning for Robotics Manipulation . . . . .	5
<b>3 LEGS: Learning Efficient Grasp Sets for Exploratory Grasping</b>	<b>6</b>
3.1 Introduction . . . . .	6
3.2 Problem Statement . . . . .	7
3.3 Learned Efficient Grasp Sets . . . . .	9
3.4 Simulation Experiments . . . . .	12
3.5 Physical Experiments . . . . .	14
3.6 Discussion . . . . .	17
<b>4 Safe Self-Supervised Learning in Real of Visuo-Tactile Feedback Policies for Industrial Insertion</b>	<b>18</b>
4.1 Introduction . . . . .	18
4.2 Problem Statement . . . . .	20
4.3 Method . . . . .	22
4.4 Experiments . . . . .	26
4.5 Discussions . . . . .	30
<b>5 Reflections</b>	<b>32</b>



**Bibliography**

# List of Figures

- 3.1 **Top: LEGS in Physical Experiments:** LEGS repeatedly attempts grasps on an object, and if the grasps are successful, it re-drops the object into a new stable pose. **Bottom: LEGS Active Set Evolution:** LEGS works by adaptively curating a small active set of promising grasps out of a large reservoir of grasp candidates (left). As exploration progresses, LEGS refines its active set (shown in bolded red/green) to contain higher quality grasps (right). . . . . 7
- 3.2 **Simulated Grasping Experiments Case Study:** We report the performance of LEGS and baselines on four specific objects to investigate how object properties affect performance. For each object, we include a 3D rendering of the object and the number  $N$  of stable poses (left), a histogram of the ground truth grasp success probabilities over 2000 sampled grasps (middle), and learning curves (right). . . 13
- 3.3 **Early Stopping Threshold Sensitivity:** We evaluate early stopping over the Dex-Net Adversarial object set in simulation with a range of stopping thresholds,  $\rho_{min}$ . We use a 95%-confidence lower bound on expected grasp robustness. **Left:** We plot the accuracy averaged over all objects and find that our empirical lower bound (Section 3.4) is highly accurate across all stopping thresholds,  $\rho_{min}$ . **Right:** We plot the number of steps before stopping, averaged across all objects. Intuitively, the required exploration time increases with higher performance thresholds. Importantly, the average number of steps before stopping is much lower than the 3000-step horizon. . . . . 15
- 3.4 **Physical Experiments Results:** We compare LEGS with BORGES ( $K_s = 2000$ ) on three objects (Bar Clamp, Pawn, and Pipe Connector) from the Dex-Net Adversarial Dataset [56] in physical experiments. All physical experiments are completed within 3 hours. LEGS significantly outperforms BORGES ( $K_s = 2000$ ) on Bar Clamp and Pawn, with minor improvement on Pipe Connector. . . . . 16
- 4.1 Overview of the learned two-phase insertion policy: the red arrows indicating the robot actions given by the policies. (A) The robot grasps the part at an initial pose. (B) The tactile guided policy  $\pi_{tac}$  estimates the grasp pose using the tactile image and aligns the z-axis of the part with the insertion axis. (C) A vision-guided policy  $\pi_{vis}$  is used to insert the part. (D) The part is inserted successfully into the receptacle. . . . . 19

4.2	Experiment setup and coordinate system. The $x$ , $y$ , $z$ axes are labeled by red, green, blue respectively. We label the gripper frame, part frame, human-provided target pose frame, and robot frame as $F_G$ , $F_p$ , $F_h$ , $F_R$ respectively. The insertion direction is defined as the $z$ -axis of $F_h$ . When the part is inserted, $F_h = F_p$ . . . . .	21
4.3	CAD models for the parallel jaw grippers and camera mount. . . . .	23
4.4	Data collection for alignment (Top) and insertion policies (Bottom). Data collection for the <b>Alignment policy</b> starts with the part inserted into the receptacle (Top (A)). The robot then samples and records different grasp poses and the corresponding tactile images (Top (B)). Data collection for the <b>Insertion policy</b> starts with a sampled refined grasp (Bottom (A)) and unplugs the part to apply sampled transformations (Bottom (B)). Then the robot inserts the part (Bottom (C)) and starts the next round of data collection with a different grasp pose. . . . .	25

# List of Tables

3.1	<b>Grasping in Simulation Aggregated Results:</b> We show the optimality gap (mean $\pm$ standard error) achieved by LEGS and baselines after $H = 3000$ steps of exploration averaged over the objects in the Dex-Net Adversarial and EGAD! evaluation datasets. LEGS achieves a lower optimality gap than all baselines, indicating that LEGS is able to discover new high-performing grasps. . . . .	13
4.1	Results suggest that (1) the IL trained on 50 human demonstrations is insufficient for training an accurate part pose estimation model, and (2) frequent slippage and rotations of the USB caused by collisions with the receptacle lead to failure in training TD3. Our approach outperforms both baseline policies. . . . .	28
4.2	Mean and standard deviation of the error in predicting part pose $(x, z, \beta)$ by the tactile-based alignment policy on the test set of 45 grasps. . . . .	28
4.3	Comparing data collection success rate. We measure the number of successful insertions until failure for 125 different grasps configurations. We compare Human Demonstration with axis alignment ( <b>ZA</b> ), Single Minimum Force-Torque Refinement ( <b>ZAWF</b> ), and Minimum Force-Torque Refinement for all grasps ( <b>ZAWFG</b> ). We report the mean success rate and the standard error for three distinct human-provided target poses. . . . .	29
4.4	Ablation study with noisy target poses comparing single-phase Tactile Only, modified Vision Only, and a Combined two-phase approach leveraging tactile and visual information. . . . .	30

## Acknowledgments

There are many people without whom this project would not be possible. First, I want to thank Professor Ken Goldberg for taking me into his lab. It has been a tremendous honor to be a part of AUTOLAB and it has propelled my research career forward.

My warmest thanks go to Michael Danielczuk, Raven Huang, Justin Kerr, Ashwin Balakrishna, Daniel Brown, and Jeffrey Ichnowski who generously offered their support and guidance when I embarked on my first research project in the lab. I would also like to thank Ilija Radosavovic, Chung Min Kim, Ryan Hoque, Yahav Avigal, Max Cao, Tete Xiao, and many others who have not only inspired me to become an exceptional researcher, but have also enriched my life as cherished friends.

I would also want to thank various friends I made along the way. Yanlai Yang, Han Guo, Boyuan Chen: your inspiration from my undergraduate years onwards has been truly profound, and I eagerly anticipate the incredible advancements you will bring to the field of AI. I am especially grateful to Fanghui Wan and Mochi for their unwavering support during life's uncertain moments, and I will be forever indebted for their kindness.

Lastly, I express my deepest gratitude to my family for their relentless support and encouragement in my academic pursuits.

# Chapter 1

## Introduction

Grasping and insertion are two fundamental skills that robots should acquire and are of great interest to the robotics community. These skills have applications in various settings, such as manufacturing and assembly, logistics and warehousing, and maintenance and repair. In the past, people leveraged geometry-based reasoning and modified the end-effector or the controller of the robot to make them more adaptable to these tasks [17, 65, 51, 68, 28]. In recent literature, robotics researchers strive to create general-purpose algorithms through various learning-based methods. Although many works have demonstrated successful results in both robotics tasks, numerous challenges remain. For example, while general-purpose, learning-based grasping systems [63, 57, 79, 67, 56, 55] can successfully grasp everyday objects, they struggle to find a good grasp on newly designed, out-of-distribution industrial parts [14]. In most cases, people must manually predefine a grasp for the robot to identify, which may not be the optimal grasp. Various learning-based insertion algorithms can generalize to different parts and plugs [72, 91, 71, 54, 74, 75, 86]; however, they often require numerous demonstrations provided by human teleoperators and assume a fixed grasp pose of the part with minimal rotation. In both settings, a more scalable method is needed to enhance productivity.

A common challenge in both settings is that learning-based systems require data. While many attempts have been made to gather ImageNet-scale robotics data [20, 89, 16, 58, 32], the diversity of robot morphology, robot applications, and the need for humans to provide task-specific demonstrations or motion primitives hinder the scalability of these systems. Although leveraging simulation to generate training data and transferring the policy trained in simulation directly onto the physical robot is a viable approach, the simulation-to-real (Sim2Real) gap remains prevalent in many settings where physical robot data is still needed to bridge this gap. Consequently, the pertinent question is how to create robotics data in a scalable manner, enabling robotics algorithms to efficiently learn or adapt to specific downstream tasks.

In chapter 3, we tackle the problem of how to efficiently adapt a general-purpose grasping system autonomously to grasp objects that it is not proficient in grasping directly on a physical robot with little to no human supervision. We present LEGS: Learned Efficient

Grasp Set [24]. In this work that appeared in the IEEE International Conference on Robotics and Automation (ICRA) 2022, we extended recent work on Exploratory Grasping, which has formalized the problem of systematically exploring grasps on these adversarial objects and explored a multi-armed bandit model for identifying high-quality grasps on each object’s stable pose. However, prior formulations are still limited to exploring a small number of grasps on each object. We present Learned Efficient Grasp Sets (LEGS), an algorithm that efficiently explores thousands of possible grasps by maintaining small active sets of promising grasps and determining when it can stop exploring the object with high confidence. Experiments suggest that LEGS can identify a high-quality grasp more efficiently than prior algorithms which do not use active sets. In simulation experiments, we measure the gap between the success probability of the best grasp identified by LEGS, baselines, and the most-robust grasp (verified ground truth). After 3000 exploration steps, LEGS outperforms baseline algorithms on 10/14 and 25/39 objects on the Dex-Net Adversarial and EGAD! datasets respectively. To demonstrate the effectiveness of our algorithm, we create a self-supervised physical grasping system where the robot explore candidate grasps with minimal human intervention (roughly 1 per every 100 grasps). We then evaluate LEGS on the physical setup; trials on 3 challenging objects suggest that LEGS converges to high-performing grasps significantly faster than baselines.

In chapter 4 of the thesis, we address the issue of how to safely learn industrial insertion tasks directly on a physical robot with minimal human supervision. Learning an industrial insertion policy in real is challenging as the collision between the parts and the environment can cause slippage or breakage of the part. In this work [25] that will appear in ICRA 2023, we present a safe self-supervised method to learn a visuo-tactile insertion policy that is robust to grasp pose variations. The method reduces human input and collisions between the part and the receptacle. The method divides the insertion task into two phases. In the first *align* phase, a tactile-based grasp pose estimation model is learned to align the insertion part with the receptacle. In the second *insert* phase, a vision-based policy is learned to guide the part into the receptacle. The robot uses force-torque sensing to achieve a safe self-supervised data collection pipeline. Physical experiments on the USB insertion task from the NIST Assembly Taskboard suggest that the resulting policies can achieve 45/45 insertion successes on 45 different initial grasp poses, improving on two baselines: (1) a behavior cloning agent trained on 50 human insertion demonstrations (1/45) and (2) an online RL policy (TD3) trained in real (0/45).

# Chapter 2

## Related Works

### 2.1 Universal Grasping Algorithms

Recent robotic grasping algorithms generalize to a wide range of objects [42]. Open-loop algorithms synthesize grasps and predict their quality based on the geometry of the object, and then plan and execute a motion to attempt a high-quality grasp without feedback [46, 56, 67, 55, 57]. Closed-loop grasp planners that use vision-based gripper servoing [63, 79] and RL [36, 37] have also been popular in prior work. LEGS is designed to leverage priors from these universal grasping algorithms to efficiently learn a robust grasp policy for a specific, difficult-to-grasp object [62, 80]. We use priors from Dex-Net 4.0 [57], a general grasp planner that learns a grasp-quality estimator from a large dataset of 3D object models in simulation and then uses this estimator to sample and evaluate the quality of grasps in physical trials.

### 2.2 Multi-Armed Bandits

Prior work on multi-armed bandits [73] has studied settings where the number of actions is large compared to the number of timesteps allocated for exploration [77, 81, 34, 3, 84, 31, 10]. One popular algorithmic framework for this setting is called *best arm identification*, where the goal is to adaptively reject a set of arms from consideration when there is high confidence that they are suboptimal [2, 38, 9]. LEGS builds on these ideas, by adaptively filtering actions from an active set by maintaining confidence bounds on the reward corresponding to each action. This mechanism makes it possible to efficiently perform best arm identification across multiple bandits problems, where each bandit problem represents a distinct stable pose of an object. LEGS can quickly converge to high-quality grasps on problems with thousands of grasps per stable pose.



## 2.3 Exploratory Grasping

Universal grasping algorithms often struggle with certain objects [80, 62]. Danielczuk et al. [14] show that grasping algorithms such as Dex-Net [57] are difficult to fine-tune online on such objects, and propose *Exploratory Grasping*, a problem formulation where the objective is to perform rapid online adaptation to grasp specific, unknown objects. To achieve this, prior works sample a fixed set of grasps on specific object stable poses and apply multi-armed bandit algorithms to rapidly identify high-performing candidates [45, 53, 21, 47]. Danielczuk et al. [14] extend these ideas with BORGES, which explores grasps across all object stable poses by using Thompson sampling and a learned Dex-Net prior [47]. However, BORGES can often overlook high-quality grasps since it restricts exploration to a small initial set of grasps. To address this issue, LEGS begins with a large set of grasp candidates and adaptively curates sets of promising grasps by adding and removing grasp candidates during exploration. By doing this, LEGS is able to converge to better long-term performance than BORGES (which uses a small fixed set of grasps), while also learning to robustly grasp an object faster than baselines that seek to directly explore large sets of grasp candidates.

## 2.4 Industrial Insertion

Industrial insertion has been central in robotics for 50 years. It is challenging due to occlusions brought by the robot gripper, grasp uncertainty from the process of acquiring the part and its collision with the environment, the fragility of the parts, and the precision required in controlling the robot for insertion. Early work approached this problem using CAD information to infer desired assembly sequences [17] and generating designs of part feeders based on object geometry [65]. Other work approached the problem from an algorithmic design perspective, with a focus on developing motion planning strategies for peg insertion [52, 68].

Recently, learning-based methods have shown success on this task. This includes learning insertion policies with a physical robot via Sim2Real transfer [35], online adaptation with meta-learning [72, 91], reinforcement learning [71, 54], self-supervised data collection with impedance control [74], accurate state estimation [86], or decomposing the insertion algorithm into a residual policy that relies on conventional feedback control [35]. These approaches assume that the parts are grasped with a fixed pose. To overcome this assumption, Wen et al. [86] perform accurate pose estimation and motion tracking with a high-precision depth camera and use a behavioral cloning algorithm to insert the part. Spector and Castro [74] and Spector, Tchuiev, and Castro [75] require contact between the part and the environment to occur during data collection, a process that is expensive and often impractical for fragile parts. In comparison, we use inexpensive tactile sensors and a safe self-supervised data collection procedure that does not require such contact.

## 2.5 Tactile Sensing for Industrial Insertion

Grasped parts are often visually occluded by the gripper. Tactile feedback can be an alternative sensing modality for grasp pose estimation. Recent work uses tactile images from vision-based tactile sensors such as GelSight [90] and DIGIT [44] to estimate the pose of grasped objects. Li et al. [48] use Gelsight sensors, BRISK features and RANSAC to estimate grasp pose. Gelsight produces high-quality 3D tactile images and can determine depth imprint, which improves feature detection by isolating the object from the background. DIGIT, a more affordable tactile sensor, provides a 2D RGB image but not the light incident direction (to generate the depth image). Kelestemur, Platt, and Padir [39] generates tactile image data in simulation for pose estimation of bottle caps but simulating contact and physical interaction between tactile sensors and objects with more intricate geometry is still challenging [83]. In this work, we collect a dataset of tactile images in real for the USB connector with different grasp poses to train a tactile-based policy for grasp pose estimation.

## 2.6 Multi-Modal Learning for Robotics Manipulation

Most prior work on tight tolerance insertion tasks [86, 23, 48, 22] leverages a single modality, such as vision, tactile, or force-torque, limiting the accuracy of the system due to occlusion, perspective effect, and sensory inaccuracy. Multi-modal systems have been explored to improve the robustness of automated insertion. Spector and Castro [74] and Spector, Tchuiev, and Castro [75] use RGB cameras and a force-torque sensor for learning contact and impedance control. Chaudhury et al. [11] couple vision and tactile data to perform localization and pose estimation, and demonstrate that vision helps with disambiguating tactile signals for objects without distinctive features. Ichiwara et al. [33] leverage tactile and vision for deformable bag manipulation by performing auto-regressive prediction. Hansen et al. [29] use a contact-gated tactile, vision and proprioceptive observation to train reinforcement learning policies. Okumura, Nishio, and Taniguchi [66] also tackle the problem of grasp pose uncertainty for insertion by using Newtonian Variational Autoencoders to combine camera observations and tactile images. They demonstrate results for USB insertion accounting for grasp pose uncertainty in one translation direction. In this work, we separate the insertion problem into an alignment phase and an insertion phase, decoupling vision and tactile inputs and also present a novel safe self-supervised approach to data collection. We are able to handle both grasps pose rotation and translation uncertainty for the USB insertion task.

## Chapter 3

# LEGS: Learning Efficient Grasp Sets for Exploratory Grasping

### 3.1 Introduction

Recent advances in deep learning have enabled the development of universal grasping systems that can robustly grasp a wide variety of objects [63, 57, 79, 67, 56, 55]. However, these systems can still struggle to grasp objects with adversarial [62, 80] geometries or which are significantly out of distribution from the objects seen during training. This problem is common in many industrial settings, in which newly manufactured machine parts for custom applications may look very different from the objects in the datasets typically used for training universal grasping systems.

Recently, bandit-style algorithms have been used to augment general-purpose grasping policies by rapidly adapting them to specific objects [45, 53, 21, 47]. Recently, Danielczuk et al. [14] introduced Exploratory Grasping, where a robot learns to grasp novel objects through online exploration of grasps and stable poses. Their algorithm, Bandits for Online Rapid Grasp Exploration Strategy (BORGES), learns robust pose-specific grasping policies. However, BORGES limits exploration to a fixed set of 100 grasps per stable pose, possibly overlooking other high-quality grasps.

In this work, we extend Danielczuk et al. [14] to explore thousands of grasps per stable pose. Considering grasp sets of this scale increases the likelihood of converging to a robust grasp, but also makes efficient exploration challenging. To address this challenge, we propose *Learned Efficient Grasp Sets* (LEGS), which adaptively curates an active set of promising grasps rather than restricting exploration to a small fixed subset. The key insight is to use a combination of priors from a universal grasping system and online trials to maintain confidence bounds on grasp-success probabilities. LEGS uses these bounds to (1) update the grasps in its active set and (2) decide when to stop exploring.

This paper makes the following contributions: (1) a novel adaptive multi-armed bandits algorithm that curates a small set of high-performing grasps by actively removing and

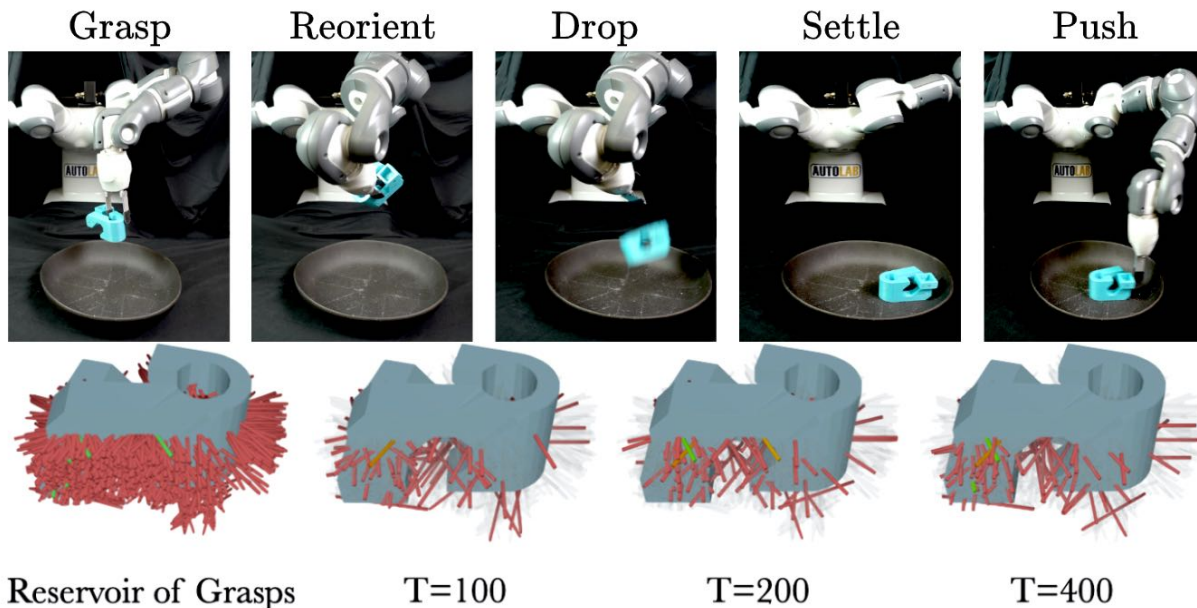


Figure 3.1: **Top: LEGS in Physical Experiments:** LEGS repeatedly attempts grasps on an object, and if the grasps are successful, it re-drops the object into a new stable pose. **Bottom: LEGS Active Set Evolution:** LEGS works by adaptively curating a small active set of promising grasps out of a large reservoir of grasp candidates (left). As exploration progresses, LEGS refines its active set (shown in bolded red/green) to contain higher quality grasps (right).

resampling grasps based on performance bounds and a novel termination condition that enables a robot to predict (with high confidence) when it reaches a desired level of performance; (2) a *self-supervised* physical grasping system where a robot explores candidate grasps with minimal human intervention (roughly 1 in every 100 grasp attempts); (3) simulation and physical experiments suggesting that LEGS can identify higher quality grasps within a fixed time horizon than prior algorithms which do not learn an active set.

## 3.2 Problem Statement

**Overview:** Given a difficult-to-grasp polyhedral object of unknown geometry that rests on a planar surface and is viewed by an overhead depth camera, we seek to learn to successfully grasp the object in all of its stable poses.

**Problem Setup:** Given a polyhedral object  $o$ , let  $N$  be its number of stable poses. Each stable pose  $s \in \{1, 2, \dots, N\}$  is associated with a landing probability  $\lambda_s$ , which indicates the probability of the object landing in pose  $s$  when released from sufficient height in a randomized orientation [27, 60]. Following Danielczuk et al. [14], we model our problem as a finite-horizon Markov Decision Process  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, R, H)$ . We let  $\mathcal{S}$  be the set of equivalence classes of distinguishable stable poses of the object and  $\mathcal{A}$  be the set of all possible grasps on the object.

Thus,  $\mathcal{A} = \bigcup_{s \in \mathcal{S}} \mathcal{A}_s$ , where  $\mathcal{A}_s$  are the grasps available at a stable pose  $s$ . Given a grasp action  $a$  in stable pose  $s$ , the transition function  $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$  determines the probability distribution over next stable poses. The reward function  $R : \mathcal{S} \times \mathcal{A} \rightarrow \{0, 1\}$  is binary: a grasp is successful and  $R(s, a) = 1$  if the grasped object does not fall from the gripper after it is lifted, and  $R(s, a) = 0$  otherwise. Let  $p_{s_a} = \mathbb{E}[R(s, a)]$  be the expected success probability of grasp  $a$  on stable pose  $s$ . We define a grasping policy as:  $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ , where  $\pi(a|s)$  denotes the probability of selecting grasp  $a$  in pose  $s$ . We denote the finite horizon of the MDP as  $H$ . The robot initially does not know any of the stable poses or the number of stable poses  $N$ . If a grasp is successful, the robot randomizes the orientation of the object in the gripper, drops the object so that the next stable pose  $s'$  is determined by the landing probabilities  $\{\lambda_s\}_{s=1}^N$ , and records the observed stable pose  $s'$ .

We represent the actions,  $\mathcal{A}_s$ , at each stable pose  $s$  as candidate grasps sampled on the object. We use the same method as Mahler et al. [57] to sample antipodal grasps on each stable pose. We do not make any assumptions on the grasping modality, so in practice these grasps can be sampled from various different grasp planners, including parallel-jaw or suction grasp planners. We denote the number of possible grasps for pose  $s$  as  $K_s = |\mathcal{A}_s|$  and the total number of grasps over all states as  $K = \sum_{s \in \mathcal{S}} K_s$ .

An important difference between our problem setting and prior work [14] is that we consider settings in which  $K$  is large ( $> 1000$ ) and thus is of the same order of magnitude as the exploration horizon,  $H$ . This significantly exacerbates exploration challenges, since there is not enough time to fully explore each grasp, motivating the key innovations in LEGS.

**Assumptions:** In this work, we assume access to the following: (1) a grasp sampler which accepts as input a depth map and outputs a set of candidate grasp configurations on the surface of the depth map with associated robustness values; (2) a robot/gripper that can either execute these grasps or detect that they are in collision; (3) sufficient information in the camera image to detect whether the object stable pose changes; (4) an evaluation function to detect whether a grasp is successful. We note that these assumptions are satisfied by the system we build to instantiate LEGS in practice. In addition, we make the following assumptions about object’s interaction with the environment: (5) if a grasp is unsuccessful, the object either remains in the same stable pose or topples into another stable pose; and (6) there exists a grasp with non-zero success probability on each stable pose. These last two assumptions are consistent with [14].

**Metrics:** We define the *optimality gap*,  $\Delta_\pi$  as

$$\Delta_\pi = \mathbb{E}_{s \in \mathcal{S}} \left[ p_s^* - p_{s_{\pi(s)}} \right] = \sum_{s \in \mathcal{S}} \lambda_s \cdot \left( p_s^* - p_{s_{\pi(s)}} \right), \quad (3.2.1)$$

where  $p_s^* = \max_{a \in \mathcal{A}_s} \mathbb{E}[R(s, a)]$  and  $p_{s_{\pi(s)}} = \mathbb{E}[R(s, \pi(s))]$ . In simulation, we can evaluate the ground-truth grasp-success probability for a given grasp with robust quasi-static grasp wrench space analysis [85]. We thus approximate  $p_s^*$  by sampling a large number of grasps on each stable pose. Intuitively, the optimality gap  $\Delta_\pi$  measures the expected difference, across all stable poses, between the optimal policy, which selects the best available grasp, and the

policy  $\pi$ . In physical experiments, the optimality gap cannot be computed so we report the grasp-success rate of the learned policy  $\pi$ .

The objective is to find a policy that minimizes the optimality gap for a given object within  $H$  grasp attempts. Denoting a policy learned after  $H$  grasp attempts by  $\pi_H$ , the objective is to identify  $\pi_H^*$  such that:

$$\pi_H^* = \arg \min_{\pi_H} \Delta_{\pi_H}. \quad (3.2.2)$$

### 3.3 Learned Efficient Grasp Sets

We propose Learned Efficient Grasp Sets (LEGS), a multi-arm bandits algorithm that uses confidence bounds on grasp-success probability to maintain a small active set of candidate grasps. LEGS starts with an estimate of the prior success probabilities for all grasps in a large reservoir of possible grasps, and updates their grasp-success probabilities based on online grasp trials using Thompson sampling as in Danielczuk et al. [14]. However, unlike BORGES, LEGS uses the priors and online grasp trials to construct confidence bounds on the grasp-success probabilities for each grasp (Section 20).

LEGS is summarized in Algorithm 2. Once LEGS visits a stable pose  $s$ , it checks whether it has visited  $s$  (line 4). In Sec. 3.5, we describe how to recognize stable poses in the physical setup. If the stable pose  $s$  has never been visited (line 5), LEGS adds the stable pose to the set of visited stable poses  $\hat{S}$  (line 6) and initializes an active set of candidate grasps,  $\tilde{A}_s$ , along with the parameters of a Beta distribution associated with each grasp in the active set (lines 7-8). We rank the grasps in the reservoir by their estimated grasp success probabilities under the Grasp Quality Convolutional Neural Network (GQ-CNN) from Dex-Net 4.0 [57] and select the  $k = 100$  grasps with the highest values. In each iteration, LEGS executes the grasp with the highest sampled value from the posterior (lines 9-11), observes the outcome (line 12), and updates the posterior distribution [70] (lines 13-16). In conjunction, LEGS also constructs confidence bounds on each of the success probabilities of each grasp (Section 20). Every  $n$  iterations, it uses these confidence bounds to identify and remove the grasps with low robustness (Section 20) (line 18), and replaces them with newly sampled grasps where grasps are ranked by their estimated grasp success probabilities under GQ-CNN (lines 19-20).

#### Constructing Confidence Bounds on Robustness

To determine which grasps to remove from the active set, LEGS constructs upper and lower confidence bounds on grasp robustness. We model the success probability of grasp  $i$  via  $X_i \sim \text{Beta}(\alpha_i, \beta_i)$ , and empirically select a confidence threshold  $\delta$ . Then the percent-point function  $\text{PPF}(X_i, \delta)$ , the inverse of the cumulative distribution function  $F_{X_i}(x)$ , returns the value  $x$  such that  $F_{X_i}(x) = \delta$ . The  $(1 - \delta)$ -lower and -upper confidence bounds for  $X_i$  are  $X_{i,\ell} = \text{PPF}(X_i, \delta)$  and  $X_{i,u} = \text{PPF}(X_i, 1 - \delta)$ , respectively. As a grasp is sampled more often, the interval  $[X_{i,\ell}, X_{i,u}]$  tightens to reflect increased certainty in the robustness of the grasp.

---

**Algorithm 1:** Learned Efficient Grasp Sets (LEGS)
 

---

```

1 Input: object  $o$ , grasp sampler  $f_\theta$ , resample interval  $n$ , number of active grasps  $k$ 
2 Initialize the set of visited stable poses  $\hat{S} = \emptyset$ 
3 for  $t = 1, 2, \dots$  do
4     Recognize the current stable pose  $s$ 
5     if  $s \notin \hat{S}$  then
6          $\hat{S} \leftarrow \hat{S} \cup \{s\}$ 
7         Use  $f_\theta$  to sample  $k$  grasps as the active set  $\tilde{A}_s$ 
8         Set  $\alpha_i$  and  $\beta_i$  based on prior for all  $i \in \tilde{A}_s$ 
9     foreach grasp  $i \in \tilde{A}_s$  do
10        sample  $\phi_i \sim \text{Beta}(\alpha_i + 1, \beta_i + 1)$ 
11    Execute grasp  $i = \operatorname{argmax}_{j \in \tilde{A}_s} \phi_j$ 
12    Observe reward  $r = R(s, i)$ 
13    if  $r = 1$  then
14         $\alpha_i \leftarrow \alpha_i + 1$ 
15    else
16         $\beta_i \leftarrow \beta_i + 1$ 
17    if  $t \equiv 0 \pmod{n}$  then
18        Remove the grasps in  $\mathcal{B} = \mathcal{B}_\ell \cup \mathcal{B}_\gamma$  from  $\tilde{A}_s$  (see equations (3.3.1) and (3.3.2))
19        Sample  $|\mathcal{B}|$  new grasps using  $f_\theta$ 
20        For each new grasp  $j = 1, \dots, |\mathcal{B}|$ , set  $\alpha_j, \beta_j$  using prior from  $f_\theta$  and add new
        grasp to  $\tilde{A}_s$ 
    
```

---

## Posterior Dependent Grasp Removal

LEGS avoids over-exploring less robust grasps by identifying and removing grasps from the active set that are highly likely to be either (1) inferior to another grasp in the active set (*locally suboptimal*) or (2) below a desired global grasp success probability threshold (*globally suboptimal*). Let the highest lower confidence bound across all active grasps be:  $X_\ell^* = \max_{i \in \tilde{A}_s} X_{i,\ell}$ . We define the set of *locally suboptimal grasps* as the set of grasps for which their  $(1 - \delta)$ -confidence upper bound is worse than the  $(1 - \delta)$ -confidence lower bound for the best grasp in the active set:

$$\mathcal{B}_\ell = \{i : X_{i,u} < X_\ell^*\}. \quad (3.3.1)$$

Thus,  $\mathcal{B}_\ell$  represents the set of grasps that are likely to be inferior to the best known grasp in the active set. However, in the early stages of exploration, we may not yet have sampled a high-performing grasp and  $\mathcal{B}_\ell$  may be empty. In these cases, we still desire to remove and resample grasps that, with high-confidence, are clearly low performing. Thus, given a minimum performance threshold  $\gamma \in [0, 1]$ , we define the set of *globally suboptimal* grasps in the active set (denoted  $\mathcal{B}_\gamma$ ): grasps which have been sampled, but are likely to have success

probability less than  $\gamma$ . We define  $\mathcal{B}_\gamma$  as

$$\mathcal{B}_\gamma = \{i : X_{i,u} < \gamma\}. \quad (3.3.2)$$

We denote the set of attempted grasps in the active set as  $\mathcal{P}$ , and let the index of the currently known best grasp be  $i^*$ . The full set of grasps removed by LEGS is constructed by taking the union of the above sets:  $\mathcal{B} = (\mathcal{B}_\ell \cup \mathcal{B}_\gamma) \cap \mathcal{P} \setminus \{i^*\}$ . This allows LEGS to remove grasps which are unlikely to outperform the best known grasp in the current active set.

## Early Stopping

Rather than setting the exploration horizon  $H$  to a fixed value, we can set a performance threshold and let LEGS stop exploring once it has high confidence that it has achieved the desired threshold. This early stopping condition allows LEGS to efficiently allocate exploration time by only continuing to explore objects that it cannot yet robustly grasp.

Given a user-specified, minimum performance threshold  $\rho_{\min} \in [0, 1]$ , we want to detect when, with high likelihood, the true performance of LEGS is above this threshold. More formally, given a confidence parameter  $\delta_{stop} \in [0, 1]$ , we want to calculate a  $(1 - \delta_{stop})$ -confidence lower bound, denoted by  $p_\ell$ , on the true expected performance of the grasping policy  $\pi$ , i.e., we want to find  $p_\ell$  such that  $Pr\left(p_\ell \leq \mathbb{E}_{s \in \mathcal{S}}[p_{s,\pi(s)}]\right) \geq 1 - \delta_{stop}$ . Then, the robot can stop exploring when  $p_\ell \geq \rho_{\min}$ .

We cannot directly compute  $\mathbb{E}_{s \in \mathcal{S}}[p_{s,\pi(s)}]$  since we do not know the true stable pose distribution  $\mathcal{S}$ . Thus, we take a Bayesian approach where we approximate  $p_\ell$  by sampling likely values of  $\mathbb{E}_{s \in \mathcal{S}}[p_{s,\pi(s)}]$  given the observed data and then by taking the  $\delta_{stop}$ -percentile of these samples [6, 7]. First, for each observed stable pose,  $s$ , we estimate the expected performance of the best grasp as  $\hat{p}_s^* = \max_{i \in \mathcal{A}_s} \frac{\alpha_i}{\alpha_i + \beta_i}$ , where  $\alpha_i$  and  $\beta_i$  are the parameters of the Beta posterior distribution over the success probability of grasp  $i$ . To reason about the performance of LEGS, we must account for uncertainty over the stable pose distribution, parametrized by the drop probabilities  $\lambda_1, \dots, \lambda_N$ . However,  $N$  is unknown. Thus, we model our belief over drop probabilities using a Dirichlet posterior distribution over  $\hat{N} + 1$  drop probabilities, where  $\hat{N}$  is the number of observed stable poses and the +1 allocates probability mass to unobserved stable poses.

Assuming a uniform Dirichlet prior, we take the empirical drop counts  $c_1, \dots, c_{\hat{N}}$  for  $\hat{N}$  observed stable poses, and sample from the posterior distribution over stable pose drop probabilities,  $Pr(\{\lambda_s\}_{s=1}^{\hat{N}+1} \mid c_1, \dots, c_{\hat{N}}, 0)$ . Due to conjugacy [18], the desired posterior distribution is also a Dirichlet distribution with parameters  $(\alpha_1 = c_1 + 1, \dots, \alpha_{\hat{N}} = c_{\hat{N}} + 1, \alpha_{\hat{N}+1} = 1)$ . Given a sample,  $\{\lambda'_s\}_{s=1}^{\hat{N}+1}$ , from the above Dirichlet posterior, we transform it into a sample from the posterior over expected grasp robustness:  $p'_\pi = \sum_{s=1}^{\hat{N}} \hat{p}_s^* \cdot \lambda'_s$ . where we conservatively assume that the robot will fail to grasp the object in any unseen poses. We calculate a  $(1 - \delta_{stop})$ -confidence lower bound on the overall grasp robustness by finding the  $\delta_{stop}$  percentile,  $\hat{p}_\ell = \text{PPF}(p'_\pi, \delta_{stop})$ , using  $M$  samples of  $p'_\pi$ .



## 3.4 Simulation Experiments

### Experimental Setup

We first evaluate LEGS in Exploratory Grasping with a variety of adversarial objects in simulation. Same as in Danielczuk et al. [14], we consider 14 Dex-Net 2.0 Adversarial objects [56] and all 39 EGAD! Adversarial evaluation objects [62]. We use Dex-Net 4.0 [57] to sample a large reservoir of  $K = 2000$  grasps for each stable pose. We also use GQ-CNN to set the Beta prior for LEGS following the method from [47, 14]. Using the method outlined in Section 4.3, we update the active grasp set after every  $n = 100$  timesteps and use  $\delta = 0.05$  for constructing grasp confidence intervals with upper confidence threshold  $\gamma = 0.2$ . All experiments use a time horizon of  $H = 3000$ . We run 10 trials of each algorithm with 10 rollouts per trial, where each trial involves sampling a different reservoir of grasps, and each rollout for a trial involves running a grasp exploration algorithm.

### Baselines

We compare LEGS against five baseline algorithms: Dex-Net, Tabular Q-Learning (TQL), BORGES ( $K_s = 100$ ), BORGES ( $K_s = 2000$ ), and LEGS (-AS). Dex-Net greedily chooses the best grasp evaluated by Dex-Net 4.0 [57] for each stable pose and does not do any online exploration. BORGES ( $K_s = 100$ ) leverages a prior calculated by GQ-CNN to seed grasp success probability estimates, and then performs Thompson Sampling for each encountered stable pose to explore an initial active set of 100 grasps sampled on each of the poses. While BORGES ( $K_s = 100$ ) is provided with the same initial active set as LEGS, unlike LEGS, BORGES ( $K_s = 100$ ) does not update its set over time. However, different from [14], it is not guaranteed that there will exist successful grasps on all stable poses when  $K_s = 100$ . This implies that BORGES ( $K_s = 100$ ) may not be able to transit between stable poses. The  $K_s = 100$  Upper Bound refers to the optimality gap if on each stable pose, the best grasp in the active set is selected. BORGES ( $K_s = 2000$ ) is identical to BORGES ( $K_s = 100$ ), but instead directly explores the full reservoir of  $K_s = 2000$  sampled grasps. TQL implements tabular Q-learning on the full reservoir of  $K_s = 2000$  sampled grasps where each pose is a separate state  $s$  and each action  $a$  is a grasp on that pose and a Q-table  $Q[s, a]$  is constructed to keep track of the corresponding 1-step Q-values. The values in the Q-table are initialized using the GQ-CNN prior and actions are chosen based on an  $\epsilon$ -greedy policy [76] with  $\epsilon = 0.1$ . Finally, LEGS (-AS) is not provided with an initial active set, but instead operates on the full reservoir of  $K_s = 2000$  grasps and uses the posterior dependent removal procedure in Section 20 to remove grasps from the reservoir.

### Experimental Results

We first study aggregated results of LEGS and baselines over objects in the Dex-Net Adversarial and EGAD! evaluation datasets in Table 3.1. We find that LEGS performs better

Dataset	Dex-Net	TQL	BORGES ( $K_s = 100$ )	$K_s = 100$ Upper Bound	BORGES ( $K_s = 2000$ )	LEGS (-AS)	LEGS
Dex-Net	$0.56 \pm 0.07$	$0.23 \pm 0.08$	$0.13 \pm 0.07$	$0.08 \pm 0.04$	<b><math>0.04 \pm 0.03</math></b>	$0.22 \pm 0.06$	<b><math>0.04 \pm 0.03</math></b>
EGAD!	$0.59 \pm 0.03$	$0.32 \pm 0.04$	$0.25 \pm 0.04$	$0.13 \pm 0.03$	$0.17 \pm 0.03$	$0.28 \pm 0.04$	<b><math>0.14 \pm 0.03</math></b>

Table 3.1: **Grasping in Simulation Aggregated Results:** We show the optimality gap (mean  $\pm$  standard error) achieved by LEGS and baselines after  $H = 3000$  steps of exploration averaged over the objects in the Dex-Net Adversarial and EGAD! evaluation datasets. LEGS achieves a lower optimality gap than all baselines, indicating that LEGS is able to discover new high-performing grasps.

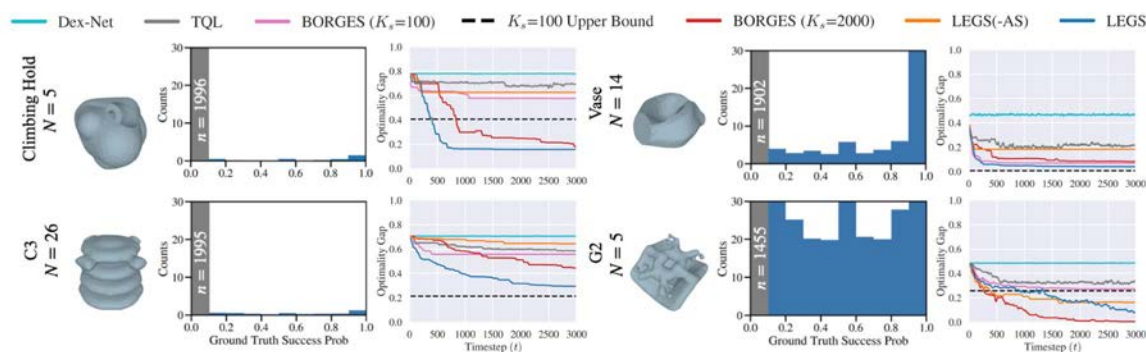


Figure 3.2: **Simulated Grasping Experiments Case Study:** We report the performance of LEGS and baselines on four specific objects to investigate how object properties affect performance. For each object, we include a 3D rendering of the object and the number  $N$  of stable poses (left), a histogram of the ground truth grasp success probabilities over 2000 sampled grasps (middle), and learning curves (right).

than or equal to the baseline algorithms on 10 out of 14 objects in the Dex-Net Adversarial dataset, and on 25 out of 39 objects in the EGAD! evaluation dataset. In comparison, the best performing baseline algorithm, BORGES ( $K_s = 2000$ ), only performs at least as well as rest of the algorithms on 5 out of 14 Dex-Net Adversarial objects and 14 out of 39 EGAD! evaluation dataset. On all of these objects we find that Dex-Net, which is not updated online, has high optimality gap, motivating online grasp exploration. The improvement for LEGS over LEGS (-AS) and BORGES ( $K_s = 2000$ ) indicates the increased efficiency of restricting exploration to a small active set, while the gap between LEGS and BORGES ( $K_s = 100$ ) indicates the importance of updating this active set over time to prune poor performing grasps while discovering new, high-quality grasps outside of the initial active set. BORGES ( $K_s = 100$ ) cannot outperform the success rate of the best grasp in its initial set ( $K_s = 100$  upper bound). By contrast, LEGS, retains the efficiency of only exploring a small set of grasps while also being able to adapt this set over time to obtain successful grasps on difficult-to-grasp stable poses and reach a lower optimality gap. TQL learns much

more slowly than BORGES because it fails to leverage the structure in the grasp exploration problem and does not learn separate policies for each stable pose.

In Figure 3.2, we study LEGS and baselines on specific objects. We show two objects (Climbing Hold and C3) where LEGS converges faster to high performing grasps than prior algorithms and two objects (F6 and Turbine Housing) where LEGS does not outperform all baselines. We find that when high performing grasps are abundant, LEGS may converge to suboptimal grasps. However, when there are only few successful grasps, LEGS can converge to good grasps much faster than baselines. If high quality grasps are already in the active set, LEGS can rapidly distinguish them from other grasps. If the active set does not contain successful grasps, LEGS can quickly replace bad grasps in the active set.

## Early Stopping Results

Next, we study the accuracy and effectiveness of the early stopping criterion (Section 20). We test the proposed high-confidence performance bound across all objects in the Dex-Net Adversarial object set (individual results per object are reported in the supplement). We check whether LEGS has reached the stopping condition every 100 grasps for a horizon of  $H = 3000$  total grasp attempts and use  $\delta_{stop} = 0.05$ , resulting in a 95%-confidence lower bound  $\hat{p}_\ell$ . We sample  $M = 3000$  samples to estimate  $\hat{p}_\ell$ .

We first test how often the predicted bound is a true lower bound on performance. We find that, on average, across all Dex-Net Adversarial objects, our empirical lower bound is a 95.8%-accurate lower bound on the true performance over the true stable pose distribution. Thus,  $\hat{p}_\ell$  forms an empirically valid  $(1 - \delta_{stop})$ -confidence lower bound. We next test the tightness of our lower bound. On average, the difference between the true performance of LEGS and our empirical lower bound is only 2.97%. These results suggest that our lower bound is highly accurate and tight enough to provide a practical signal for when the robot can safely stop exploring.

We next study, in simulation, the use of our high-confidence bounds on performance for early stopping. As described in Section 20, given a user-specified, minimum performance threshold,  $\rho_{min}$ , the robot stops exploring when the lower confidence bound  $\hat{p}_\ell$  is greater than  $\rho_{min}$ . When the robot chooses to stop exploring the object, we evaluate the ground truth performance of the learned policy and evaluate whether the true performance is also above the threshold  $\rho_{min}$ . We evaluate a wide range of thresholds and plot the results in Figure 3.3. Results suggest that we can achieve highly accurate early stopping, allowing the robot to accurately terminate exploration well before the full horizon of 3000 steps.

## 3.5 Physical Experiments

In this section, we discuss our experimental setup for physical experiments, the methods we used to enable intervention-free grasp exploration on a physical robot and results evaluating the performance of LEGS and BORGES ( $K_s = 2000$ ) across 3 physical objects.

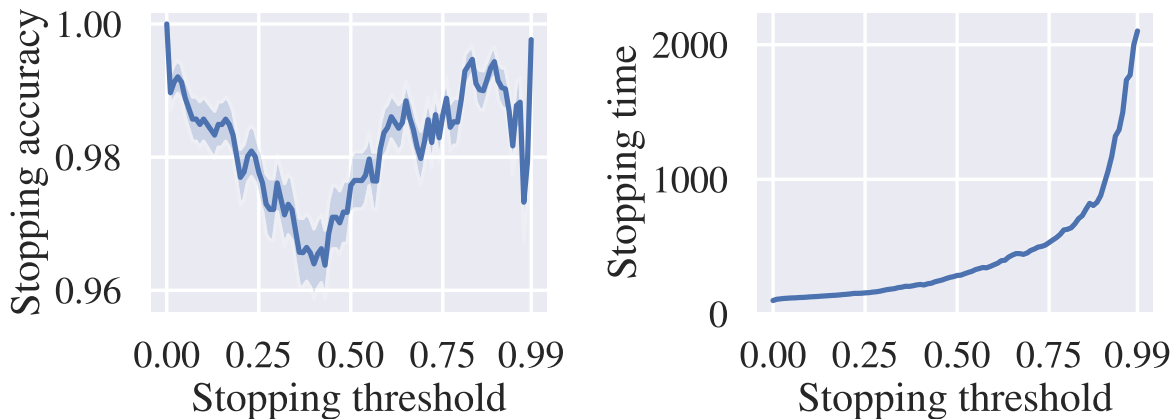


Figure 3.3: **Early Stopping Threshold Sensitivity:** We evaluate early stopping over the Dex-Net Adversarial object set in simulation with a range of stopping thresholds,  $\rho_{min}$ . We use a 95%-confidence lower bound on expected grasp robustness. **Left:** We plot the accuracy averaged over all objects and find that our empirical lower bound (Section 3.4) is highly accurate across all stopping thresholds,  $\rho_{min}$ . **Right:** We plot the number of steps before stopping, averaged across all objects. Intuitively, the required exploration time increases with higher performance thresholds. Importantly, the average number of steps before stopping is much lower than the 3000-step horizon.

## Experimental Setup

To deploy exploratory grasping algorithms on a physical robot, we modify the perception system introduced in Danielczuk et al. [14] to sample grasps and identify changes in the object stable pose. We capture a depth image of the object from an overhead camera, deproject it into a point cloud using the known camera intrinsics, demean the point cloud, and apply 3600 evenly spaced rotations to the point cloud around the camera’s optical axis. We measure the chamfer distance between the rotated point clouds with previously cached point clouds and find the pair of point clouds that serves as the closest match. As in Danielczuk et al. [14], if at least 80% of the points are less than 0.02 mm away from the closest points in the cached point cloud, we classify the two point clouds as belonging to the same stable pose. If none of the cached point clouds satisfies this condition, the point cloud is cached and treated as a new stable pose. If there exists a matching point cloud, we further align the translation and rotation of the point cloud via iterative closest point [12].

Upon discovery of a new stable pose, we use Dex-Net 4.0 [57] to sample, evaluate, and cache grasps in the grasp reservoir. Thus, LEGS can explore grasps on objects with unknown geometries and unknown numbers of stable poses.

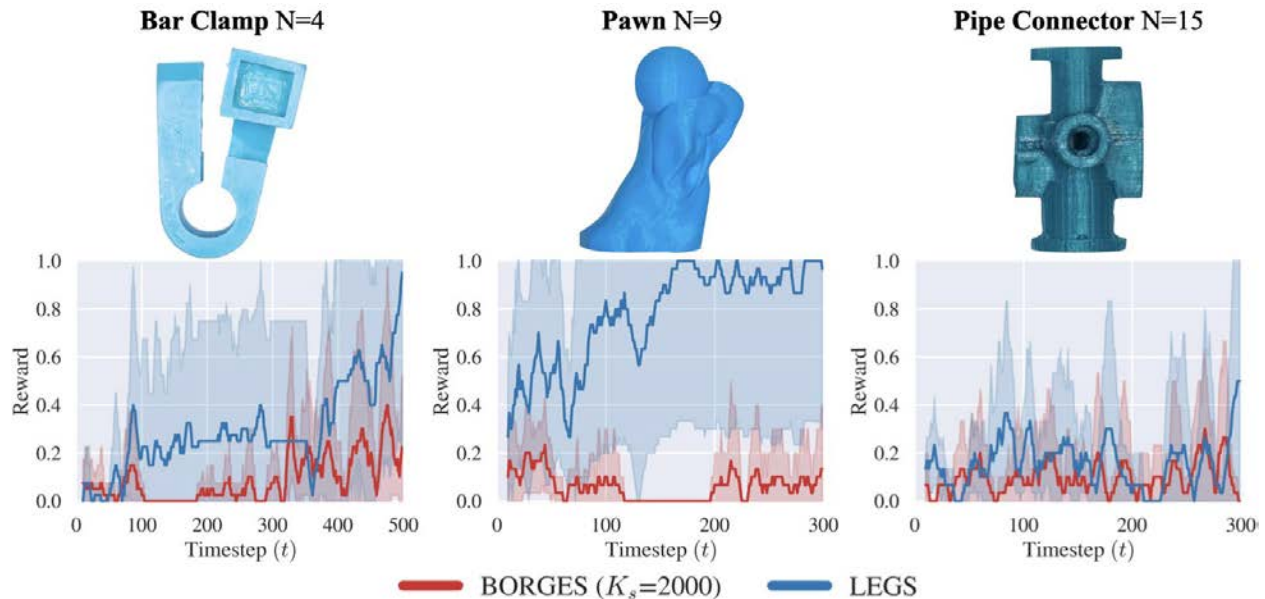


Figure 3.4: **Physical Experiments Results:** We compare LEGS with BORGES ( $K_s = 2000$ ) on three objects (Bar Clamp, Pawn, and Pipe Connector) from the Dex-Net Adversarial Dataset [56] in physical experiments. All physical experiments are completed within 3 hours. LEGS significantly outperforms BORGES ( $K_s = 2000$ ) on Bar Clamp and Pawn, with minor improvement on Pipe Connector.

## Self-Supervised Exploratory Grasping with LEGS

Danielczuk et al. [14] find that re-dropping the object during experiments often cause it to fall out of the workspace, requiring extensive human effort to reset the object. To enable the robot to collect grasp data without human intervention, we introduce strategies to prevent the object from toppling out of the workspace while maintaining access to a wide variety of grasps. We drop the object within a bowl (Fig.4.1), where the object’s rebound height is lower than the rim of the bowl. The bowl allows the object to stay in the visible range of the overhead camera. However, the bowl’s rim can be an obstacle to grasps. We introduce two autonomous *reset* behaviors to address this: (1) we center the object above the bowl before dropping the object, and (2) when the object topples near the boundary, the robot pushes the object towards the center of the bowl to improve grasp access [15].

## Experimental Results

Figure 3.4 shows learning curves from physical experiments comparing LEGS with BORGES ( $K_s = 2000$ ) on three challenging objects from the Dex-Net Adversarial Dataset [56]. We run 3 trials with 1 rollout per trial for each object. We find that on 2 out of the 3 objects, LEGS is able to outperform BORGES ( $K_s = 2000$ ) and identify high-performing grasps within a few hundred timesteps of online exploration.

## 3.6 Discussion

We present Learned Efficient Grasp Sets, an algorithm which efficiently explores large sets of grasps by adaptively constructing a small active set of promising grasps. Experiments suggest that LEGS identifies high-performing grasps more efficiently than baseline algorithms across 53 objects in simulation experiments and on three challenging objects in physical trials. We also propose a novel early stopping condition by computing a high-confidence lower bound on the expected grasp performance. Simulation results suggest that this high-confidence lower bound is highly accurate and tight. In future work, we will analyze LEGS to determine how the quality of the Dex-Net prior and the distribution over grasp success probabilities affect its convergence rate. Moreover, we will search for possible ways for LEGS to generalize across different stable poses and objects.

## Chapter 4

# Safe Self-Supervised Learning in Real of Visuo-Tactile Feedback Policies for Industrial Insertion

### 4.1 Introduction

Industrial assembly [59] is a precise manipulation task requiring contact between parts. Part feeding, peg insertion and object reorientation (three sub-tasks of industrial assembly) have been extensively studied [51, 52, 28, 65, 59]. Early work considers the mechanical design aspect [65, 52] and motion planning aspect [51, 28, 68]. Through Computer Aided Design (CAD), the order of part assembly can be predetermined in simulation with precise pose information [17], allowing robots to plan the necessary actions to assemble the design [43]. Learning-based approaches recently have shown promise on industrial insertion tasks [86] on the NIST taskboard [40], a standard benchmark that represents common industrial insertion tasks with parts that have complex geometries [49]. However, applying learning-based methods for industrial insertion remains challenging due to the requirement for frequent human inputs during learning [54] or high-precision sensors for collecting training data [86]. There is also a need for a safe training and data collection method for learning insertion tasks since parts are prone to breakage.

Another challenge in an industrial insertion task is that the precise grasp pose is often unknown due to variations in kitting and feeding of parts as they arrive for assembly. As the grasped part is often occluded by the gripper visually, grasp pose estimation is better achieved when using tactile sensing [66]. While recent work has shown improvement in simulation accuracy for industrial insertion [64] and successes in Sim2Real transfer for tactile-based insertion tasks [39, 82], the simulation of soft contacts between tactile sensors and objects with complex geometries remains an open problem [83], and often can not transfer to real because the object models are not publicly available. In this work, we present a novel method to safely learn visuo-tactile feedback policies in real for industrial insertion tasks under grasp

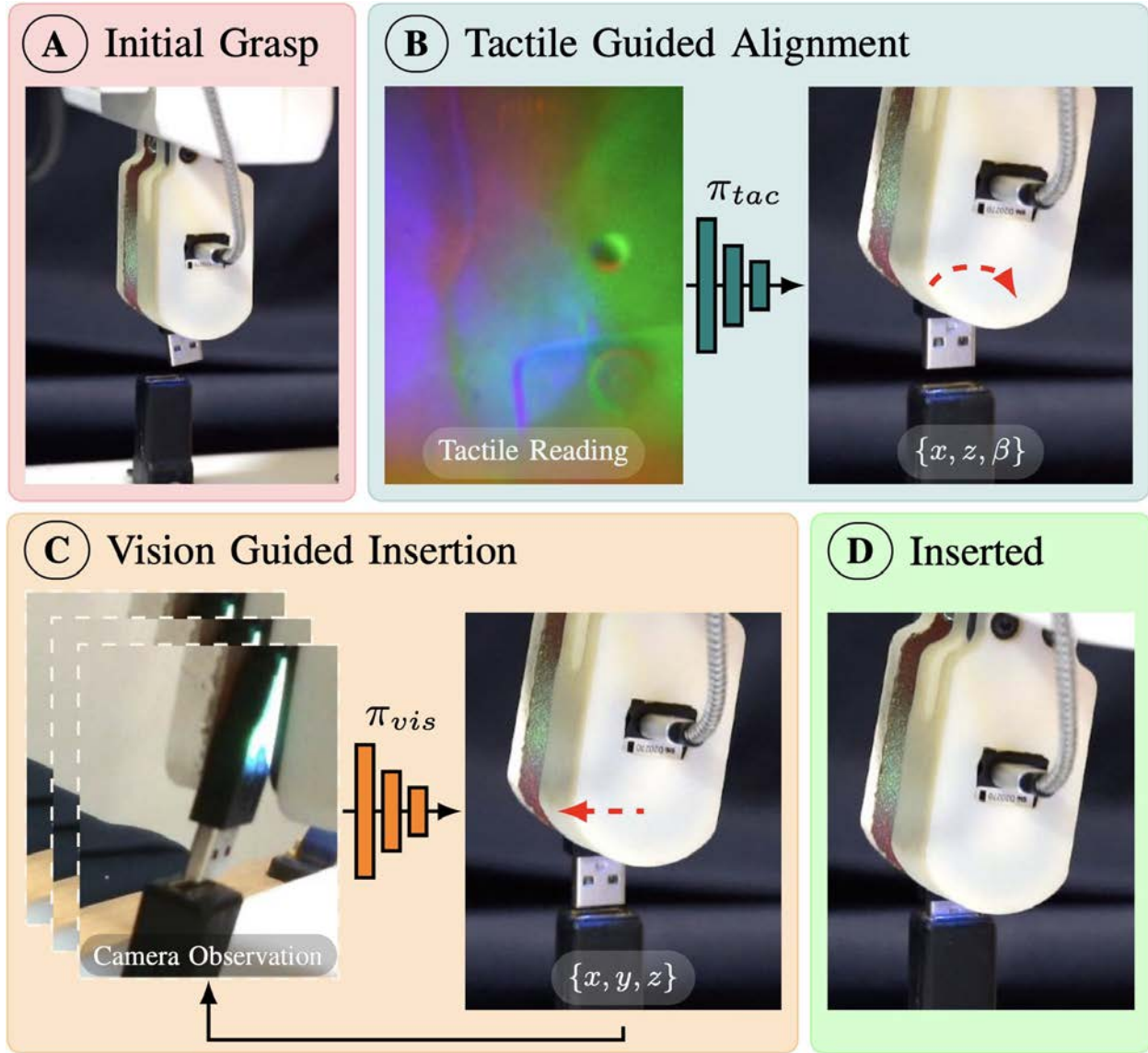


Figure 4.1: Overview of the learned two-phase insertion policy: the red arrows indicating the robot actions given by the policies. (A) The robot grasps the part at an initial pose. (B) The tactile guided policy  $\pi_{tac}$  estimates the grasp pose using the tactile image and aligns the z-axis of the part with the insertion axis. (C) A vision-guided policy  $\pi_{vis}$  is used to insert the part. (D) The part is inserted successfully into the receptacle.

pose uncertainties, with inexpensive off-the-shelf sensors. Our approach draws on tactile and visual feedback to deal with the grasp pose uncertainty and force-torque sensing for a self-supervised training procedure that is *safe*, minimizing damage during the training phase. We divide the insertion task into two phases (as shown in Fig. 4.1):

- An initial *Align* phase where a tactile-based policy  $\pi_{tac}$  estimates the grasp pose. The



robot reorients and aligns the part with the insertion axis of the receptacle.

- A second *Insert* phase where an RGB image-based policy  $\pi_{vis}$  guides the robot to insert the part.

A significant challenge in learning tactile feedback policies in real for industrial insertion is the frequent slippage of the part that occurs due to collision with the environment and the smooth surface of the tactile sensor gel pad. This makes RL methods difficult to succeed without human intervention or an automatic reset mechanism to detect and correct slippage. In this work, we develop a self-supervised data collection pipeline that avoids collision between the part and its environment, by recognizing that the insertion operation is reversible only from certain target insertion poses – i.e. starting from such poses, the part can be repeatedly unplugged from and inserted into the receptacle. Prior to data collection, a human free-drives the robot to provide one approximate target pose where the part is inserted. The robot refines this target pose to find such a reversible pose by minimizing the grasping force-torque, which helps minimize collisions during data collection, resulting in a safer training process that is unlikely to damage the insertion part and receptacle.

This paper contributes:

1. A safe self-supervised data collection pipeline with force-torque sensing in real for insertion, designed to minimize contact force for data collection.
2. A two-phase policy learned from the collected data including a tactile-based alignment policy for orienting the part and an RGB image-based insertion policy;
3. Experimental results suggest that the policy achieves 45/45 successes on USB connector insertion, outperforming two baseline methods (1/45 and 0/45).

## 4.2 Problem Statement

### Overview

We consider a part insertion task using a 7-DoF robot, equipped with a parallel-jaw gripper with a tactile sensor mounted on one jaw. The end-effector has a wrist-mounted RGB camera, and the robot provides reliable force-torque readings at the end-effector. The objective is to learn a policy that can robustly insert the part into the receptacle with an unknown part’s pose within the gripper, while minimizing human inputs and part-receptacle collisions during training. Fig. 4.2 shows the experiment setup and the coordinate frames.

### Assumptions

We make the following assumptions:

1. A human provides one top-down grasp pose of the part inserted in the receptacle.

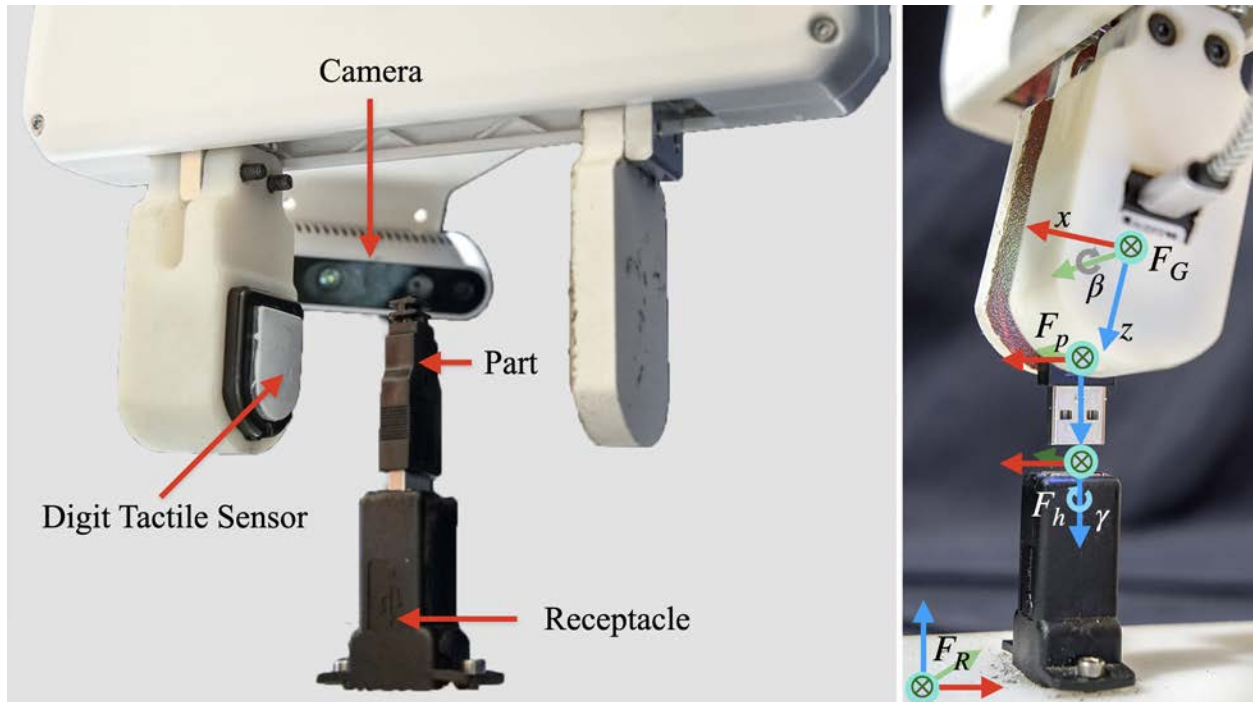


Figure 4.2: Experiment setup and coordinate system. The  $x$ ,  $y$ ,  $z$  axes are labeled by red, green, blue respectively. We label the gripper frame, part frame, human-provided target pose frame, and robot frame as  $F_G$ ,  $F_p$ ,  $F_h$ ,  $F_R$  respectively. The insertion direction is defined as the  $z$ -axis of  $F_h$ . When the part is inserted,  $F_h = F_p$ .

2. The robot can accurately measure force and torque either with an external sensor or internal current sensing;
3. An experiment begins with the part pre-grasped by the robot gripper with the part grasp pose within a range relative to the human-provided pose.
4. During data collection, the robot operates in a rectangular collision-free configuration space above the receptacle.

## Problem Setup

Given a tight-fitting receptacle for the part to be inserted into (Fig. 4.2), we find the target insertion pose  $T_{R,h}$  of the part (grasped at an unknown pose) from one human-provided imperfect demonstration. At any time step  $t$ , we have access to the RGB observation  $o_\rho(t)$  from the wrist-mounted camera, the RGB tactile image  $o_\psi(t)$  from the DIGIT tactile sensor, and reliable force  $\vec{f}(t)$  and torque  $\tau(t)$  readings from the robot. Since we know the insertion axis, we parameterize the action space with 4 degrees of freedom: gripper translation in the robot frame and gripper  $y$ -axis rotation.

## 4.3 Method

### Hardware

We design a novel parallel jaw gripper mount to accommodate the DIGIT tactile sensor [44] and camera mount (Fig. 4.2, 4.3). The elastomer gel on the DIGIT sensor deforms, causing torque applied to the part. This torque and the force applied from the receptacle to the part during insertion often produce undesired slippage and rotation. Therefore, we develop an asymmetric mounting setup where we mount the DIGIT sensor on one jaw with reinforcement to prevent outward bending, while keeping the other jaw flat. We apply sandpaper on the surface of the non-tactile gripper, increasing the friction to reduce slippage. We find that we can predict the part’s grasp pose of a USB connector using a single DIGIT sensor.

### Self-Supervised Data Collection

#### One Human Provided Imperfect Target Pose

The target pose  $T_{R,h}$  is provided by a human free-driving the robot with a pre-grasped part to insert it in the receptacle from top down. Since this target pose may not have a perfect axis alignment with the receptacle, the system performs a  $z$ -axis alignment of the target pose. To account for the change in grasp center after axis alignment, we refine  $T_{R,h}$  by finding a target pose that minimizes gripper force-torque using a grid search through a set of translations and rotations  $\{T_\Delta\}$ . Formally, we find

$$\tilde{T}_{R,h} = \arg \min_{\tilde{T}_{R,h} \in \{T_\Delta \cdot T_{R,h}\}} \left\| \vec{f}(\tilde{T}_{R,h}) \right\| + \left\| \tau(\tilde{T}_{R,h}) \right\|. \quad (4.3.1)$$

Here  $\vec{f}(\tilde{T}_{R,h})$  and  $\tau(\tilde{T}_{R,h})$  denote the 3-DoF force and torque vectors respectively when the gripper is at  $\tilde{T}_{R,h}$ . Intuitively, this objective minimizes the external force applied on the part when being unplugged, increasing the likelihood of the insertion process being reversible. Practically, we perform grid sampling over 5 values of  $x \in [-1, 1]$ mm, 5 values of  $y \in [-1, 1]$ mm and 4 values of  $\gamma \in [-\frac{\pi}{180}, \frac{\pi}{180}]$ rad ( $x, y, \gamma$  are in  $F_h$ , refer to Fig. 4.2). The pose with minimum gripper external force-torque is recorded as the refined demonstration pose  $\tilde{T}_{R,h}$ , and we have  $F_h = F_G$  at  $\tilde{T}_{R,h}$ .

A cascaded impedance controller, implemented within the robot’s real-time control loop, allows fine-grained force control. In case of a force violation, our system calculates a trajectory to a safe state within a single control cycle. After refinement of the target pose, we search for the minimum offset  $z_{\min}$  for the part to be unplugged from the receptacle. Finding the minimum height helps to determine the boundary for data collection and allows the pipeline to collect more data closer to the receptacle while reducing collisions. Iteratively, the robot moves the gripper by  $-\Delta_z$  in  $F_G$ . We then move the gripper by  $\Delta_x$  (in the  $F_G$  frame) and measure  $\vec{f}_x$  (x-component of the gripper force in the  $F_G$  frame). If  $\vec{f}_x \leq \eta$ , we register the total

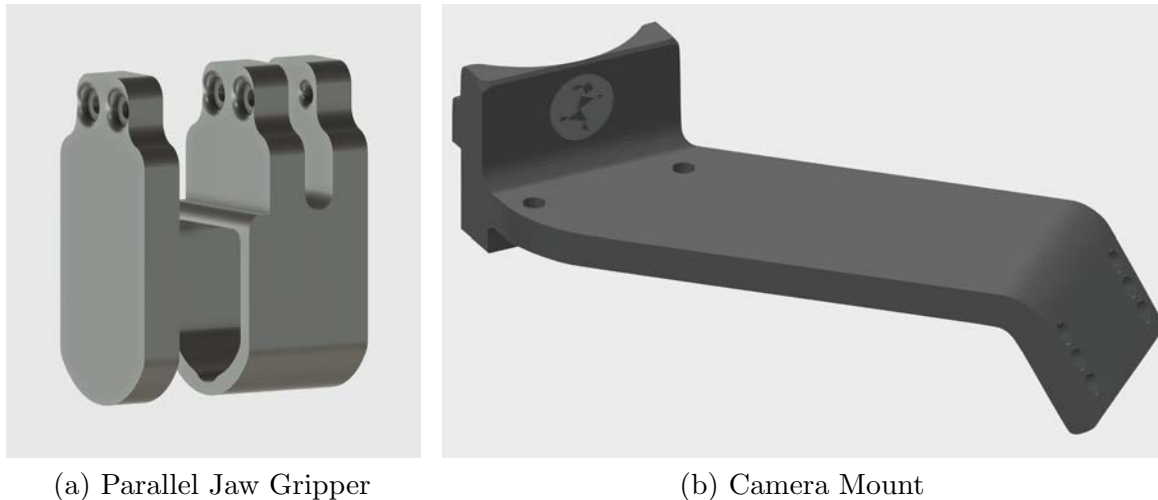


Figure 4.3: CAD models for the parallel jaw grippers and camera mount.

upward distance traveled as the minimum height  $z_{\min}$  for removal of the part. Empirically, we find setting  $\Delta_x = \Delta_z = 1$  mm, and  $\eta = 3.5$  N works well.

### Data Collection for Alignment

The part remains inserted throughout data collection. We explore grasp pose variations in 3-DoF ( $x$ ,  $z$  translation and  $y$ -axis rotation  $\beta$ ) in  $F_G$  (Fig. 4.2). We perform uniform random sampling over the range  $[-3, 3]$ mm,  $[-8, -2]$ mm,  $[-\frac{\pi}{15}, \frac{\pi}{15}]$ rad for  $x$ ,  $z$ ,  $\beta$ , with 5, 10 and 20 samples respectively. The robot closes the gripper with a force of 70N at each of the sampled poses and records the pair of tactile image readings and  $x, z, \beta$ . To account for the noise in the DIGIT tactile sensor, we take a median filter over 5 consecutively captured tactile images. We collect 2000 data points in 120 minutes.

### Data Collection for Insertion

Upon completing the tactile image collection phase, we collect robot poses and RGB images for training the insertion policy for different grasps. We perform grid sampling with 5 samples each of  $x, z, \beta$  in  $F_G$  within the same range as in the previous stage, resulting in a total of 125 grasps. To account for the difference between the sampled grasp  $g$  and  $\tilde{T}_{R,h}$ , we perform minimum force-torque refinement on the sampled grasp to calculate the grasp-specific target pose  $\tilde{T}_{R,h}(g)$ .  $\tilde{T}_{R,h}(g)$  translated by an offset of  $z_{\min}$  gives us the unplugged part pose  $T_{\text{unplug}}(g)$ .

For each grasp  $g$ , we collect image data for the visuoservo policy by moving the gripper to sampled points on a grid above the target pose. In particular, we uniformly sample 5 values each of  $x \in [-5, 5]$ mm,  $y \in [-5, 5]$ mm and  $z \in [-5, 0]$ mm in  $F_R$  with respect to  $T_{\text{unplug}}(g)$ , resulting in 125 different translations for the gripper. For each translation, we collect *one*

data point that does not contain additional rotation and sample *two* gripper  $y$ -axis rotation conditioned on  $z$  to avoid collision. Specifically, given a height  $z$ , the two rotations are sampled from the uniform distribution  $\frac{z}{5} \cdot \mathcal{U}[-\frac{\pi}{15}, 0]$ rad and  $\frac{z}{5} \cdot \mathcal{U}[0, \frac{\pi}{15}]$ rad. These rotated data points provide the system with additional data for camera pose variation with respect to the target pose, which leads to a balanced dataset with 375 distinct gripper poses. Each data point is composed of the gripper pose (translated and rotated away from the target pose) and the corresponding RGB image observation at that pose. Upon visiting all 375 gripper poses for a given grasp, the robot moves to  $T_{\text{unplug}}(g)$ , performs a vertical movement to  $\tilde{T}_{R,h}(g)$ , and opens the gripper jaws, thereby resetting the part in the receptacle. The system repeats this data collection process for all 125 grasps without human supervision.

## Learning to Insert

While human demonstrations usually serve as “expert policies” for industrial insertion tasks, the self-supervised data collection pipeline allows us to collect ground truth actions at scale. This allows us to formulate two supervised-learning problems based on Sec. 4.3 and Sec. 4.3.

### Alignment Policy

We use the data collected from Sec. 4.3 to train an alignment policy  $\pi_{tac}$  that, given the tactile image, outputs the desired displacement of the gripper  $T_{p,G}$  to align the part with the receptacle (Fig. 4.1.B). We augment the tactile images by randomly jittering the brightness and contrast over the range  $\mathcal{U}[0.7, 1.3]$ .

### Insertion Policy

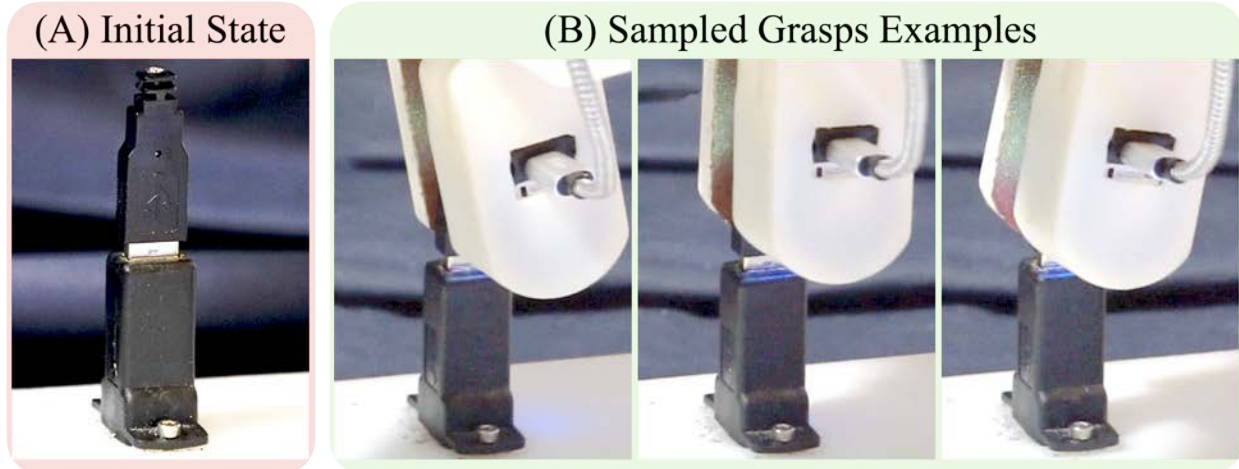
We use data collected from Sec. 4.3 to train a visuoservo insertion policy  $\pi_{vis}$  taking normalized camera observation, gripper  $y$ -axis rotation  $\beta$  and  $x, y$  translations in  $F_h$  as inputs, and predict the action: desired translation  $\Delta_x, \Delta_y$  and rotation  $\Delta_\beta$  in  $F_R$  (Fig. 4.1.C). We augment camera observations by randomly jittering the brightness and contrast over the range  $\mathcal{U}[0.7, 1.3]$ .

We use RegNet 3.2GF [69] as the backbone for both policies. For the alignment policy, we replace the last layer of RegNet with a linear layer with 3 outputs. For the insertion policy, we concatenate the robot’s pose with the latent vector of the image and replace the last layer with a linear layer with 3 outputs. For both networks, we use a batch size of 64, a learning rate of 1e-3, and a learning rate decay of 0.99 for every 100 gradient steps. We pick the mean squared error as the loss function and use the Adam optimizer [41].

## Execution of Insertion

To avoid catastrophic failure (i.e. collision between the part and the surrounding environment or moving out of the training set distribution), we deliberately formulate the visuoservo policy to only control  $x, y$ -axis translation but not  $z$ -axis translation since the insertion direction is

## Tactile Guided Alignment Policy Data Collection



## Vision Guided Insertion Policy Data Collection

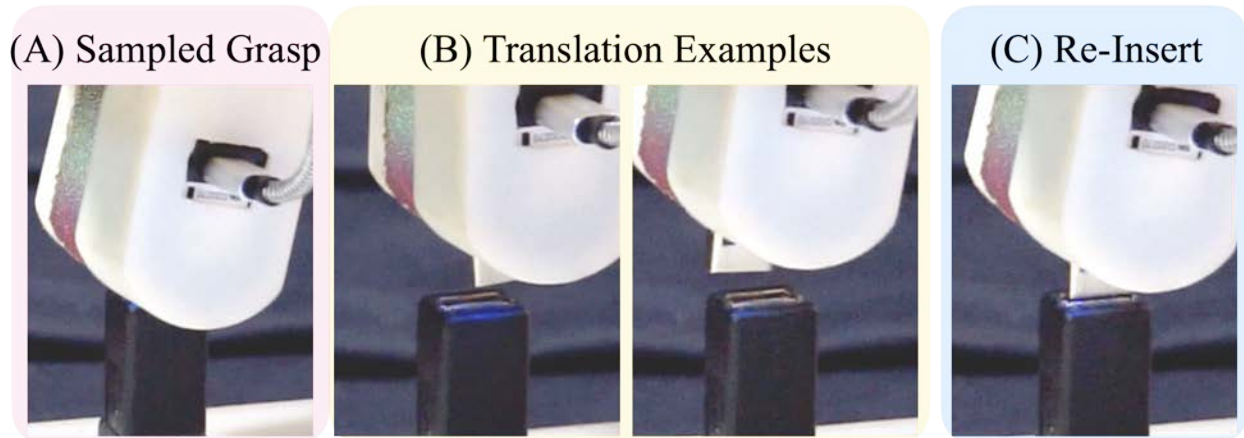


Figure 4.4: Data collection for alignment (Top) and insertion policies (Bottom). Data collection for the **Alignment policy** starts with the part inserted into the receptacle (Top (A)). The robot then samples and records different grasp poses and the corresponding tactile images (Top (B)). Data collection for the **Insertion policy** starts with a sampled refined grasp (Bottom (A)) and unplugs the part to apply sampled transformations (Bottom (B)). Then the robot inserts the part (Bottom (C)) and starts the next round of data collection with a different grasp pose.

already aligned with the  $z$ -axis by the *Align* policy. The formulation is detailed in Algorithm 2. The execution procedure starts by inferring the grasp pose from the tactile image via  $\pi_{tac}$  (line 2). The system then performs the insertion axis alignment of the part with the receptacle (line 3). We measure the  $z$ -direction force based on the gripper force-torque sensor (line 12). We then calculate rotation and translation based on the camera observation via  $\pi_{vis}$  (line 5, 10) and execute the corresponding actions (line 8) until the action has a norm

---

**Algorithm 2:** Policy Execution Procedure

---

```

1 Input: Tactile Image  $o_\rho(t)$ , Camera Image  $o_\psi(t)$ , tactile based grasp pose estimation
   network  $\pi_{tac}$ , and visuoservo insertion policy  $\pi_{vis}$ , Target Pose  $\tilde{T}_{R,h}$ , Minimum
   Wrench Height  $z_{\min}$ , action norm threshold  $\epsilon$ , z direction step size  $d_z$ 
2  $T_{p,G} = \pi_{tac}(o_\rho(t))$ 
3 Move gripper to  $\tilde{T}_{R,h}T_{p,G} + 2z_{\min}$ 
4 attempts = 0
5  $(\Delta_x, \Delta_y, \Delta_\beta) = \pi_{vis}(o_\psi(t), T_{R,G}(t))$ 
6 while True do
7     while  $\|[\Delta_x, \Delta_y]\| > \epsilon$  and  $attempts < H$  do
8         Move gripper by  $(\Delta_x, \Delta_y)$ 
9         attempts = attempts + 1
10         $(\Delta_x, \Delta_y, \Delta_\beta) = \pi_{vis}(o_\psi(t), T_{R,G}(t))$ 
11    Translate gripper in insertion direction by  $d_z$ 
12    Measure gripper force-torque in z-axis as  $F_z$ 
13    if  $F_z > 15N$  or  $attempts \geq H$  then
14        Terminate

```

---

smaller than  $\epsilon$  (line 7). Note that the rotation prediction from the *Insertion Policy* is not used, because the part is aligned with the insertion axis by the *Alignment Policy*. When the action converges, we lower the gripper in the  $z$  direction by a step size of  $d_z$  (line 11) and continue to query  $\pi_{vis}$  until the force constraint is satisfied or the number of attempts exceeds the horizon  $H$  (line 13-14). We empirically set  $d_z = 1.5\text{mm}$ . Empirically, we find setting the action norm  $\epsilon = 0.0005$ ,  $d_z = 1.5\text{mm}$  and  $H = 200$  works well, as the force-based termination condition is usually triggered first for a successful or unsuccessful insertion.

## 4.4 Experiments

### Experiment Setup

We focus on the USB insertion task on the NIST taskboard. We use a 7-DoF Franka Emika robot with a parallel gripper, where one DIGIT tactile sensor is mounted on the inside of one of the fingers. An Intel RealSense camera is mounted offset from the gripper (Fig. 4.3). To control the Cartesian position of the Franka robot, a time-optimal trajectory respecting velocity, acceleration, and jerk constraints is applied to the policy’s positional output [4]. We use grid sampling to obtain 5 values of  $\beta$  ranging from  $[-\frac{\pi}{20}, \frac{\pi}{20}]$ rad, 3 translations in x and z from the range  $[-3, 3]\text{mm}$  and  $[-6, -2]\text{mm}$  in  $F_h$ , resulting in a test set of 45 different grasp configurations that lie in the training distribution of the algorithms.

## Experimental Procedure

At the beginning of each test experiment, the USB connector (part) is pre-grasped by the robot with a grasp pose selected from the test set and located at a position with a  $z$ -axis translation of  $2z_{\min}$  relative to  $F_h$ . At this starting pose, the gripper is aligned vertically down while the USB connector is misaligned with the receptacle in both translation and rotation as in Fig. 4.1(A). The robot first executes the *Align* policy to estimate the part grasp pose and aligns it with the insertion direction of the receptacle as in Fig. 4.1(B). It then uses the *Insert* policy to visuoservo and inserts the part into the receptacle as in Fig. 4.1(C-D). The robot then resets to the next grasp by releasing the part, re-grasping it, raising it to a start pose as outlined above, and executing the *Align* and *Insert* policies for this new grasp. It steps through all the test grasp poses using the same procedure.

An experiment terminates if the gripper frame ( $F_G$ ) force in the  $z$  direction  $\vec{f}_z(t)$  exceeds 15 N. An experiment trial is considered successful if the gripper is within 5 mm of the target pose in  $H = 200$  iterations, upon which we also visually inspect whether the insertion is successful. In this set of experiments, the refined human-provided target pose  $\tilde{T}_{R,h}$  is provided as an input to the policies. In Sec. 4.4, we perform ablation studies on noisy target poses.

## Comparison

The method is designed with two objectives: (1) minimizing human intervention so that ideally no human needs to be involved in data collection or training of the policy, and (2) minimizing collision among the robot, the part, and the environment. We compare our approach with two baseline learning methods described below with the same environment setup (using the same grippers and camera mount).

### Twin Delayed Deep Deterministic Policy Gradient(TD3)

An off-policy, online reinforcement learning policy [26] that learns the end-to-end part insertion. This baseline satisfies objective (1) but violates objective (2) — i.e. it is incapable of avoiding collisions among the robot, the part, and the environment. We simplify the problem to a fixed, axis-aligned grasp pose and restrict the action space to translations only, so that the policy only have to learn insertion instead of both alignment and insertion. The learned policy runs with a frequency of 10 Hz. Since the policy outputs Cartesian position changes, we controls the Cartesian velocity of the robot, which is equivalent to Cartesian position changes per time step. Due to the part’s axis-alignment with the receptacle, the policy’s input can be restricted to the (low-dimensional) robot pose, velocity, and the force-torque at frame  $F_G$ . We use the default TD3 implementation of Ray RLLib [61]. If the force applied on the gripper exceeds 15 N, the episode terminates and the robot resets to a safe starting pose.



### Imitation learning (IL)

Imitation learning from 50 human demonstrations of insertion trained with behavior cloning. Each human demonstration starts with a randomly sampled grasp pose as in Sec. 4.3. This baseline algorithm violates objective (1) but satisfies objective (2), where the human demonstrator selects actions that minimize collision between the part and the environment. It takes about 30 minutes to provide all these human demonstrations.

Algorithms	IL	TD3	Proposed Approach
Success/Total	1/45	0/45	45/45

Table 4.1: Results suggest that (1) the IL trained on 50 human demonstrations is insufficient for training an accurate part pose estimation model, and (2) frequent slippage and rotations of the USB caused by collisions with the receptacle lead to failure in training TD3. Our approach outperforms both baseline policies.

## Results

The results are summarized in Table. 4.1. The imitation learning agent (IL) is only able to perform a single successful insertion out of the 45 grasp poses. Intuitively, 50 different grasp configurations from human demonstrations are not sufficient for training an accurate part pose estimation model; additionally, for different grasp poses, at the same gripper pose, two different human demonstrations may exist. The multi-modality in the distribution of target insertion poses contributes to the failure of the IL policy. Training the TD3 policy in the physical environment led to divergent results in all 5 training trials we attempted. In all cases, the part collides with the receptacle, leading to a drastic change in the grasp pose. This cannot be corrected directly since there is no reset procedure that can systematically recover the gripper to its original state without human supervision. Our approach succeeds for every single grasp pose tested. Empirically, we find that the part rotation and translation predicted from tactile images are fairly accurate (refer to Table. 4.2).

	$x$ (mm)	$z$ (mm)	$\beta$ (rad)
Mean Error	8.97e-2	1.46e-1	5.59e-3
Standard Deviation	4.89e-3	6.62e-2	4.89e-3

Table 4.2: Mean and standard deviation of the error in predicting part pose ( $x, z, \beta$ ) by the tactile-based alignment policy on the test set of 45 grasps.

Setup	Trial 1	Trial 2	Trial 3	Mean±Standard Error
ZA	0/125	0/125	0/125	0.0±0.0%
ZAWF	125/125	125/125	34/125	75.7±19.8%
ZAWFG	125/125	125/125	125/125	100.0±0.0%

Table 4.3: Comparing data collection success rate. We measure the number of successful insertions until failure for 125 different grasps configurations. We compare Human Demonstration with axis alignment (**ZA**), Single Minimum Force-Torque Refinement (**ZAWF**), and Minimum Force-Torque Refinement for all grasps (**ZAWFG**). We report the mean success rate and the standard error for three distinct human-provided target poses.

## Ablation Studies

### Effects of Leveraging Force-Torque Sensing in Data Collection

We compare the completion rate of the data collection process for insertion with or without grasp pose refinement. We consider the following three different methods for refining the target pose gained from the human demonstration  $T_{R,h}$ : 1) **ZA**: apply  $z$ -axis alignment on  $T_{R,h}$ , 2) **ZAWF**: perform minimum force-torque refinement only once for  $T_{R,h}$  after  $z$ -axis alignment (this step is only performed for the first grasp and the results reused for all grasps) and 3) **ZAWFG**: perform  $z$ -axis alignment for  $T_{R,h}$  and apply minimum force-torque refinement separately for each grasp.

At the beginning of each experiment, a human provides  $T_{R,h}$  by free-driving the robot with one pre-grasped part to insert the part. A total of 125 different grasp poses are sampled. For each grasp pose  $T_{h,G}$ , we calculate the grasp pose in robot frame by  $T_{R,G} = \tilde{T}_{R,h}T_{h,G}$  with  $\tilde{T}_{R,h}$  determined by one of the three methods **ZA**, **ZAWF** or **ZAWFG**. The robot grasps the part with the pose  $T_{R,G}$ , lifts the part, and tries to re-insert the part. If insertion is successful, the robot executes the next grasp otherwise the experiment terminates. We report total number of successful insertions before termination (Table. 4.3). We repeat the experiment three times for each method with different human demonstrations.

After applying  $z$ -axis alignment for the human-provided target pose (**ZA**), the insertion fails as the center of the grasp is not aligned with the center of the receptacle. **ZAWF** addresses this issue by using minimum pose refinement, and can perform successful insertions. However, since the pose refinement is specific to the human demonstration, the refinement is not sufficiently granular, leading to failures when the new grasp configuration has a large  $y$ -axis rotation. **ZAWFG** performs pose refinement for each of the grasps, resulting in consistent insertion performance. **ZAWFG** needs to wait for the force measurements to settle and thus takes longer to execute.

### Exploring Utility of Tactile and Vision Information

We perform study the relative benefits of using tactile and vision for insertion tasks. We test 3 different approaches: (1) A Tactile Only approach (2) A Vision Only approach trained

Algorithm	Tactile Only	Vision Only (No Rot)	Tactile + Vision (No Rot)
Success/Total	21/45	8/45	40/45

Table 4.4: Ablation study with noisy target poses comparing single-phase Tactile Only, modified Vision Only, and a Combined two-phase approach leveraging tactile and visual information.

using a limited amount of the camera observation data and (3) a Combined Approach. This ablation study differs from our earlier experiments by injecting a uniformly sampled noise in the range  $\pm 1\text{mm}$  into the target pose’s  $x, y$  translation to imitate imprecise knowledge of the target pose.

The Tactile Only approach attempts the entire insertion task in a single phase using the tactile information to align the USB connector with the receptacle and then move straight down to insert it. For the purpose of this study, we train a new Vision Only model using a third of the collected camera observation data set that have no additional gripper rotation. This mimics a mono-view visuoservo model for receptacle localization and top-down insertion. We modify the insertion motion from Sec. 4.3 so the model only uses camera observation and the  $y$ -axis rotation of the gripper, and it only outputs translation in  $x$  and  $y$  based on the same regression objective as in Alg. 4.3. The combined approach sequences a Tactile Only approach in an *Align* phase with the modified Vision Only approach in an *Insert* phase. We perform experiments with the three different approaches with the same test set as in Sec. 4.4 and report results in Table 4.4.

With noisy target pose, the Tactile Only model succeeds only 21/45 times since the model does not estimate the target receptacle state. Since the Vision Only model is constrained to translation actions and is trained on a limited set of data that does not include additional gripper rotation, it inserts the part when the grasps do not have any rotation (8/9 successes) but fails otherwise (for a total of 8/45 successes). A separate Vision Only model trained with all the data is similarly unsuccessful (11/45 successes), indicating the importance of the ability to correct for grasp pose variation using tactile data. The combination of the two models outperforms either model by leveraging the part rotation prediction (using tactile) and implicitly estimating the environment state (using visual information), suggesting that tactile and vision observation jointly reduce the uncertainties in the insertion problem.

## 4.5 Discussions

### Limitations

Despite promising results, this method has not been tested for generalization to other types of assembly tasks, objects with more complex geometries, or objects that are larger than the tactile sensor. Parts made of different materials may require distinct maximal forces; the grid search for finetuning the insertion pose lengthens the data collection process. The robot must also unplug the part, which can pose a challenge as some parts are designed to be difficult to

remove (i.e. an Ethernet connector). This work did not measure the time required for data collection.

## **Future Work**

Future research can improve the time required for collecting data. In summary, we present a safe, self-supervised method for learning a visuotactile insertion policy in real industrial settings with unknown grasp poses. We achieve this by using force-torque sensing to refine human-demonstrated target poses and constructing a two-phase approach to insertion that separates the task into alignment and insertion based on tactile and visual feedback.

# Chapter 5

## Reflections

In the aforementioned works, we explored two specific robotic manipulation tasks: grasping and insertion. In particular, for robots to perform well in these settings using a learning-based framework, a significant obstacle is the source of data for the learning algorithms. Both works approached this problem from an automation standpoint by examining the repeatable properties of the tasks. In the grasping setting, to achieve continuous grasp data collection, we modified the robot learning environment by introducing boundaries and utilized heuristics such as linear pushing policies and random dropping to allow the robot to explore different stable poses of the object. In the insertion setting, to autonomously collect plug insertion data, we observed that the process of part insertion must be invertible – a good trajectory for unplugging the part can be inverted to create a good trajectory for inserting the plug. This insight led us to use force torque as a measure for ranking different end-effector poses when the part is inserted, which forms the backbone of our scalable autonomous data collection pipeline for insertion.

However, automating these two seemingly simple tasks remains challenging, and a significant amount of manual tuning is required before the process runs smoothly. For example, in [24], one might question the optimal height for dropping the object, the appropriate speed for the robot to push the object so that it is centered within the workspace, and the ideal location for placing the bowl to ensure it falls entirely within the reachable workspace. In [25], hours of tuning have gone into determining the force threshold for the minimal force-torque search to identify the optimal insertion pose, finding the angle at which the camera should be pointed at the gripper, and designing the end-effector such that it minimizes slip during insertion while also reducing the likelihood of breaking the part in case of insertion failure.

While the two methods above offer valuable insights into scalable approaches for gathering physical data in grasping and part insertion tasks, most robotics applications continue to operate in low-data environments, relying heavily on human demonstrations. Long-horizon tasks demand that robots master not just a single skill, but a multitude of primitive abilities, necessitating a larger dataset than what is currently accessible. Furthermore, we consistently face the challenge of "Moravec's paradox," where tasks considered simple by humans, such as cutting vegetables, wiping surfaces, and screwing bolts, are not only challenging for robots to

do but also lack scalable methods for robots to learn them effectively.

Over the past year, we have observed initial attempts at scaling robotic data and reaping the benefits of such scaling. Recent methods based on supervised or reinforcement learning, such as RT1 [5], Palm-SayCan [1], and PaLM-E [19], utilize large-scale human teleoperation demonstrations to produce language-conditioned behaviors. This enables the generation of downstream robot policies that either select from an array of RL expert policies or directly generate actions based on language conditioning. Inspired by the recent success of self-supervised representation learning in the vision and language communities [78, 30, 8], an alternative approach involves pretraining a neural network on robotics data for improved downstream adaptability. Recent studies [88, 50, 87] have shown that this pretraining scheme enhances robot trajectory modeling capabilities and adaptability for downstream control policies. However, structured exploration for collecting pretraining data remains challenging. I surmise that a favorable approach would be to leverage numerous expert policies (e.g., DexNet [56] for grasping) currently available in various domains to generate training data in regions where task resetting is straightforward. Moreover, recent work has demonstrated success in utilizing energy-based models to train robot policies with few human demonstrations [13]. It may be possible to train a handful of policies from human demonstrations, which can then collect more data within their environments to learn specific skills, and a set of dual policies for learning and executing actions that can reset the state, so that the data collection can progress without human interventions. With increased data scale, we can distill these policies into a single policy capable of leveraging multiple skills to complete downstream tasks.

However, before addressing these questions or exploring these methods, creating a generalizable robot dataset is difficult, and sharing it across labs may prove challenging. For instance, operating a Franka Emika robot for a single afternoon with a three-camera setup can generate up to 4TB of data. Thus, it is crucial to find a generalizable input and control output representation that can be applied to various robot modalities and shared across research labs. Additionally, identifying the appropriate pretraining scheme using these representations is essential.

As preliminary objectives for my PhD journey, I aim to tackle the following problems: 1) identifying a suitable input-output representation for robot learning that can be shared among robots with different morphologies, 2) developing a structured and efficient method for generating robotic data at scale, and 3) investigating the possibility of creating emergent robot behavior by merging different robot policies. I hope to make progress in addressing some of these challenges during my studies.

I am grateful that, over the past two and a half years, AUTOLAB has offered me invaluable hands-on experience and lessons in integrating learning-based algorithms with physical robotic systems, creating 3D designs, and more importantly designing experiments and delivering clear and well-motivated presentations. I would like to express my appreciation once again to Professor Goldberg, collaborators, and friends for giving me this opportunity. I eagerly anticipate applying what I've learned in my future research, fostering collaborations, and forging new friendships during my PhD at Berkeley in the coming years.

# Bibliography

- [1] Michael Ahn et al. “Do As I Can and Not As I Say: Grounding Language in Robotic Affordances”. In: *arXiv preprint arXiv:2204.01691*. 2022.
- [2] Jean-Yves Audibert, Sébastien Bubeck, and Remi Munos. “Best Arm Identification in Multi-Armed Bandits”. In: *COLT 2010 - The 23rd Conference on Learning Theory* (Nov. 2010), pp. 41–53.
- [3] Donald Berry et al. “Bandit Problems With Infinitely Many Arms”. In: *The Annals of Statistics* 25 (Oct. 1997). DOI: 10.1214/aos/1069362389.
- [4] Lars Berscheid and Torsten Kröger. “Jerk-limited Real-time Trajectory Generation with Arbitrary Target States”. In: *Robotics: Science and Systems XVII* (2021).
- [5] Anthony Brohan et al. “RT-1: Robotics Transformer for Real-World Control at Scale”. In: *arXiv preprint arXiv:2212.06817*. 2022.
- [6] Daniel Brown and Scott Niekum. “Efficient probabilistic performance bounds for inverse reinforcement learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.
- [7] Daniel S Brown, Yuchen Cui, and Scott Niekum. “Risk-aware active inverse reinforcement learning”. In: *Conference on Robot Learning*. PMLR. 2018, pp. 362–372.
- [8] Tom Brown et al. “Language models are few-shot learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.
- [9] Sébastien Bubeck, Tengyao Wang, and Nitin Viswanathan. “Multiple Identifications in Multi-Armed Bandits”. In: *ICML’13* (2013), I–258–I–265.
- [10] A. Carpentier and Michal Valko. “Simple regret for infinitely many armed bandits”. In: (2015).
- [11] Arkadeep Narayan Chaudhury et al. “Using Collocated Vision and Tactile Sensors for Visual Servoing and Localization”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3427–3434.
- [12] Dmitry Chetverikov et al. “The trimmed iterative closest point algorithm”. In: 3 (2002), pp. 545–548.
- [13] Cheng Chi et al. “Diffusion Policy: Visuomotor Policy Learning via Action Diffusion”. In: *Proceedings of Robotics: Science and Systems (RSS)*. 2023.

- [14] Michael Danielczuk et al. “Exploratory Grasping: Asymptotically Optimal Algorithms for Grasping Challenging Polyhedral Objects”. In: *Conf. on Robot Learning (CoRL)* (2020).
- [15] Michael Danielczuk et al. “Linear push policies to increase grasp access for robot bin picking”. In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2018, pp. 1249–1256.
- [16] Sudeep Dasari et al. “Robonet: Large-scale multi-robot learning”. In: *arXiv preprint arXiv:1910.11215* (2019).
- [17] LS Homem De Mello and Arthur C Sanderson. “A correct and complete algorithm for the generation of mechanical assembly sequences”. In: *1989 IEEE International Conference on Robotics and Automation*. IEEE Computer Society. 1989, pp. 56–57.
- [18] Persi Diaconis and Donald Ylvisaker. “Conjugate priors for exponential families”. In: *The Annals of statistics* (1979), pp. 269–281.
- [19] Danny Driess et al. “PaLM-E: An Embodied Multimodal Language Model”. In: *arXiv preprint arXiv:2303.03378*. 2023.
- [20] Frederik Ebert et al. “Bridge data: Boosting generalization of robotic skills with cross-domain datasets”. In: *arXiv preprint arXiv:2109.13396* (2021).
- [21] Clemens Eppner and Oliver Brock. “Visual detection of opportunities to exploit contact in grasping using contextual multi-armed bandits”. In: (Sept. 2017), pp. 273–278. DOI: 10.1109/IRROS.2017.8202168.
- [22] Yongxiang Fan, Jieliang Luo, and Masayoshi Tomizuka. “A Learning Framework for High Precision Industrial Assembly”. In: *2019 International Conference on Robotics and Automation (ICRA)* (2019), pp. 811–817.
- [23] Pete Florence et al. “Implicit Behavioral Cloning”. In: *Conf. on Robot Learning (CoRL)* (2021).
- [24] Letian Fu et al. “Legs: Learning efficient grasp sets for exploratory grasping”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 8259–8265.
- [25] Letian Fu et al. “Safely Learning Visuo-Tactile Feedback Policies in Real For Industrial Insertion”. In: *arXiv preprint arXiv:2210.01340* (2022).
- [26] Scott Fujimoto, Herke Hoof, and David Meger. “Addressing function approximation error in actor-critic methods”. In: *International conference on machine learning*. PMLR. 2018, pp. 1587–1596.
- [27] K. Goldberg et al. “Part pose statistics: estimators and experiments”. In: *IEEE Trans. Robotics and Automation* 15.5 (1999), pp. 849–857.
- [28] Kenneth Y Goldberg. “Orienting polygonal parts without sensors”. In: *Algorithmica* 10.2 (1993), pp. 201–225.



- [29] Johanna Hansen et al. “Visuotactile-RL: Learning Multimodal Manipulation Policies with Deep Reinforcement Learning”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 8298–8304.
- [30] Kaiming He et al. “Masked Autoencoders Are Scalable Vision Learners”. In: *arXiv:2111.06377* (2021).
- [31] R. D. Heide et al. “Bandits with many optimal arms”. In: *ArXiv abs/2103.12452* (2021).
- [32] Daniel Ho et al. “Retinagan: An object-aware approach to sim-to-real transfer”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 10920–10926.
- [33] Hideyuki Ichiwara et al. “Contact-rich manipulation of a flexible object based on deep predictive learning using vision and tactility”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 5375–5381.
- [34] Matthieu Jedor, Jonathan Lou  dec, and Vianney Perchet. “Be Greedy in Multi-Armed Bandits”. In: (2021). arXiv: 2101.01086 [cs.LG].
- [35] Tobias Johannink et al. “Residual reinforcement learning for robot control”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6023–6029.
- [36] Dmitry Kalashnikov et al. “Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation”. In: *Proceedings of Machine Learning Research 87* (Oct. 2018). Ed. by Aude Billard et al., pp. 651–673. URL: <http://proceedings.mlr.press/v87/kalashnikov18a.html>.
- [37] Dmitry Kalashnikov et al. “MT-OPT: Continuous Multi-Task Robotic Reinforcement Learning at Scale”. In: *arXiv* (2021).
- [38] Zohar Karnin, Tomer Koren, and Oren Somekh. “Almost Optimal Exploration in Multi-Armed Bandits”. In: *ICML’13* (2013), III–1238–III–1246.
- [39] Tarik Kelestemur, Robert Platt, and Taskin Padir. “Tactile Pose Estimation and Policy Learning for Unknown Object Manipulation”. In: *Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)* (2022).
- [40] K. Kimble et al. “Benchmarking protocols for evaluating small parts robotic assembly systems”. In: 5(2) (2020), pp. 883–889.
- [41] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: 2015.
- [42] Kilian Kleeberger et al. “A Survey on Learning-Based Robotic Grasping”. In: *Current Robotics Reports* (2020).
- [43] Yotto Koga, Heather Kerrick, and Sachin Chitta. “On CAD Informed Adaptive Robotic Assembly”. In: *arXiv preprint arXiv:2208.01773* (2022).

- [44] Mike Lambeta et al. “Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation”. In: *IEEE Robotics and Automation Letters* (2020).
- [45] Michael Laskey et al. “Multi-armed bandit models for 2D grasp planning with uncertainty”. In: (Aug. 2015), pp. 572–579. DOI: 10.1109/CoASE.2015.7294140.
- [46] Ian Lenz, Honglak Lee, and Ashutosh Saxena. “Deep learning for detecting robotic grasps”. In: *The International Journal of Robotics Research* 34.4-5 (2015), pp. 705–724. DOI: 10.1177/0278364914549607. eprint: <https://doi.org/10.1177/0278364914549607>. URL: <https://doi.org/10.1177/0278364914549607>.
- [47] Han Li et al. “Accelerating Grasp Exploration by Leveraging Learned Priors”. In: (Nov. 2020).
- [48] Rui Li et al. “Localization and manipulation of small parts using GelSight tactile sensing”. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (2014).
- [49] Wenzhao Lian et al. “Benchmarking Off-The-Shelf Solutions to Robotic Assembly Tasks”. In: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)* (2021).
- [50] Fangchen Liu et al. “Masked Autoencoding for Scalable and Generalizable Decision Making”. In: *36th Conference on Neural Information Processing Systems* (2022).
- [51] Tomas Lozano-Perez, Matthew T Mason, and Russell H Taylor. “Automatic synthesis of fine-motion strategies for robots”. In: *The International Journal of Robotics Research* 3.1 (1984), pp. 3–24.
- [52] Tomas Lozano-Pérez. “Motion planning and the design of orienting devices for vibratory part feeders”. In: *in IEEE Journal Of Robotics And Automation. MIT AI Laboratory* (1986).
- [53] Qingkai Lu, Mark Merwe, and Tucker Hermans. “Multi-Fingered Active Grasp Learning”. In: (Oct. 2020), pp. 8415–8422. DOI: 10.1109/IROS45743.2020.9340783.
- [54] Jianlan Luo et al. “Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study”. In: *Robotics: Science and Systems (RSS)* (2021).
- [55] Jeffrey Mahler and Ken Goldberg. “Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences”. In: *Conf. on Robot Learning (CoRL)* (2017).
- [56] Jeffrey Mahler et al. “Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics”. In: *Robotics: Science and Systems (RSS)* (2017).
- [57] Jeffrey Mahler et al. “Learning ambidextrous robot grasping policies”. In: *Science Robotics* 4.26 (2019). DOI: 10.1126/scirobotics.aau4984.

- [58] Viktor Makoviychuk et al. *Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning*. 2021.
- [59] KE McKee. “Automatic Assembly by G. Boothroyd, C. Poli and LE Murch, Marcel Dekker, New York, 378 pp., 1982”. In: *Robotica* 3.3 (1985), pp. 195–196.
- [60] Mark Moll and Michael A. Erdmann. “Manipulation of Pose Distributions”. In: *Int. Journal of Robotics Research (IJRR)* 21.3 (2002), pp. 277–292.
- [61] Philipp Moritz et al. “Ray: A distributed framework for emerging AI applications”. In: *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. 2018, pp. 561–577.
- [62] D. Morrison, P. Corke, and J. Leitner. “EGAD! An Evolved Grasping Analysis Dataset for Diversity and Reproducibility in Robotic Manipulation”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4368–4375.
- [63] Douglas Morrison, Peter Corke, and JÃƒergen Leitner. “Learning robust, real-time, reactive robotic grasping”. In: *The International Journal of Robotics Research* 39.2-3 (2020), pp. 183–201.
- [64] Yashraj Narang et al. “Factory: Fast Contact for Robotic Assembly”. In: *Robotics: Science and Systems (RSS)* (2022).
- [65] Balas K Natarajan. “Some paradigms for the automated design of parts feeders”. In: *The International journal of robotics research* 8.6 (1989), pp. 98–109.
- [66] Ryo Okumura, Nobuki Nishio, and Tadahiro Taniguchi. “Tactile-Sensitive NewtonianVAE for High-Accuracy Industrial Connector-Socket Insertion”. In: *arXiv preprint arXiv:2203.05955* (2022).
- [67] Lerrel Pinto and Abhinav Gupta. “Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours”. In: (2016), pp. 3406–3413.
- [68] Hong Qiao, BS Dalay, and RM Parkin. “Fine motion strategies for robotic peg-hole insertion”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 209.6 (1995), pp. 429–448.
- [69] Ilija Radosavovic et al. “Designing network design spaces”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10428–10436.
- [70] Daniel Russo et al. “A tutorial on thompson sampling”. In: *arXiv preprint arXiv:1707.02038* (2017).
- [71] Gerrit Schoettler et al. “Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 5548–5555.
- [72] Gerrit Schoettler et al. “Meta-reinforcement learning for robotic industrial insertion tasks”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 9728–9735.

- [73] Aleksandrs Slivkins. “Introduction to multi-armed bandits”. In: *arXiv preprint arXiv:1904.07272* (2019).
- [74] Oren Spector and Dotan Di Castro. “InsertionNet - A Scalable Solution for Insertion”. In: *IEEE Robotics and Automation Letters* (2021).
- [75] Oren Spector, Vladimir Tchuiev, and Dotan Di Castro. “InsertionNet 2.0: Minimal Contact Multi-Step Insertion Using Multimodal Multiview Sensory Input”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)* (2022).
- [76] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [77] Olivier Teytaud, Sylvain Gelly, and Michèle Sebag. “Anytime many-armed bandits”. In: *CAP07*. Grenoble, France, 2007. URL: <https://hal.inria.fr/inria-00173263>.
- [78] Hugo Touvron et al. “Llama: Open and efficient foundation language models”. In: *arXiv preprint arXiv:2302.13971* (2023).
- [79] Ulrich Viereck et al. “Learning a visuomotor controller for real world robotic grasping using simulated depth images”. In: *Proceedings of the 1st Annual Conference on Robot Learning*. Ed. by Sergey Levine, Vincent Vanhoucke, and Ken Goldberg. Vol. 78. Proceedings of Machine Learning Research. PMLR, 2017, pp. 291–300.
- [80] D. Wang et al. “Adversarial Grasp Objects”. In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)* (2019), pp. 241–248.
- [81] Erli Wang, Hanna Kurniawati, and Dirk P. Kroese. “CEMAB: A Cross-Entropy-based Method for Large-Scale Multi-Armed Bandits”. In: (2017). Ed. by Markus Wagner, Xiaodong Li, and Tim Hendtlass, pp. 353–365.
- [82] Shaoxiong Wang et al. “TACTO: A Fast, Flexible, and Open-source Simulator for High-resolution Vision-based Tactile Sensors”. In: *IEEE Robotics and Automation Letters (RA-L)* 7.2 (2022), pp. 3930–3937. ISSN: 2377-3766. DOI: 10.1109/LRA.2022.3146945. URL: <https://arxiv.org/abs/2012.08456>.
- [83] Yikai Wang et al. “Elastic tactile simulation towards tactile-visual perception”. In: *Proceedings of the 29th ACM International Conference on Multimedia*. 2021, pp. 2690–2698.
- [84] Yizao Wang, Jean-Yves Audibert, and R. Munos. “Algorithms for Infinitely Many-Armed Bandits”. In: (2008).
- [85] Jonathan Weisz and Peter K Allen. “Pose error robust grasping from contact wrench space metrics”. In: *2012 IEEE international conference on robotics and automation*. IEEE. 2012, pp. 557–562.
- [86] Bowen Wen et al. “You Only Demonstrate Once: Category-Level Manipulation from Single Visual Demonstration”. In: *Robotics: Science and Systems (RSS)* (2022).
- [87] Philipp Wu et al. “Masked Trajectory Models for Prediction, Representation, and Control”. In: *International Conference on Machine Learning*. 2023.

- [88] Tete Xiao et al. “Masked visual pre-training for motor control”. In: *Conf. on Robot Learning (CoRL)* (2022).
- [89] Tianhe Yu et al. “Scaling robot learning with semantically imagined experience”. In: *arXiv preprint arXiv:2302.11550* (2023).
- [90] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. “Gelsight: Highresolution robot tactile sensors for estimating geometry and force”. In: *Sensors 17* (2017).
- [91] Tony Z Zhao et al. “Offline meta-reinforcement learning for industrial insertion”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6386–6393.