Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels

by

Abhay Deshpande

Supervised by Siddhartha S. Srinivasa

A senior thesis submitted in partial fulfillment of

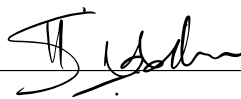the requirements for the degree of

Bachelor of Science

With Departmental Honors

Computer Science & Engineering

University of Washington
2024

Presentation of work given on: _____May 31, 2024_____

Thesis and presentation approved by: _____

Date: __May 31, 2024__

University of Washington

Abstract

Data Efficient Behavior Cloning for Fine Manipulation via Continuity-based Corrective Labels

Abhay Deshpande

Supervised by:

Professor Siddhartha S. Srinivasa
Computer Science and Engineering

We consider imitation learning with access only to expert demonstrations, whose real-world application is often limited by covariate shift due to compounding errors during execution. We investigate the effectiveness of the Continuity-based Corrective Labels for Imitation Learning (CCIL) framework in mitigating this issue for real-world fine manipulation tasks. CCIL generates corrective labels by learning a locally continuous dynamics model from demonstrations to guide the agent back toward expert states. Through extensive experiments on peg insertion and fine grasping, we provide the first empirical validation that CCIL can significantly improve imitation learning performance despite discontinuities present in contact-rich manipulation. We find that: (1) real-world manipulation exhibits sufficient local smoothness to apply CCIL, (2) generated corrective labels are most beneficial in low-data regimes, and (3) label filtering based on estimated dynamics model error enables performance gains. To effectively apply CCIL to robotic domains, we offer a practical instantiation of the framework and insights into design choices and hyperparameter selection. Our work demonstrates CCIL's practicality for alleviating compounding errors in imitation learning on physical robots.

Acknowledgements

I would like to thank Prof. Siddhartha Srinivasa for the opportunity to do meaningful research in the Personal Robotics Lab and for his valuable guidance as I began my research career. Similarly, Prof. Abhishek Gupta's supervision and unique perspective have been indispensable. I would also like to thank Liyiming Ke for her continuous mentorship and support throughout my time in the lab, which have contributed greatly to my growth as a researcher. Finally, I am grateful to Yunchu Zhang for his collaboration and camaraderie, from whom I learned a lot during our time working together.

Contents

List of Figures

Chapter 1

Introduction

Imitation learning has shown remarkable scalability with large datasets [Pomerleau, 1988; Floridi and Chiriatti, 2020], emerging as a promising paradigm for robots to mimic complex behaviors from demonstrations [Padalkar et al., 2023; Chi et al., 2023; Billard and Grollman, 2012]. However, usage of imitation learning in robotics is often constrained by covariate shift and the scarcity of demonstrations [Osa et al., 2018; Spencer et al., 2021]. Real-world robotic policies can suffer from an accumulation of sensor and joint noises when a robot executes a trajectory. These compounding errors can cause the agent to encounter out-of-distribution states, resulting in task failures.

Though various methods have been proposed to address this issue, their application can be limited by requirements such as the availability of interactive experts [Ross et al., 2011a; Bryan, 2019], simulators [Wang et al., 2024], a large-batch of sub-optimal data [Chang et al., 2021], and task-specific domain knowledge [Venkatraman et al., 2014; Ke et al., 2021b; Zhou et al., 2023a; Chen et al., 2023; Block et al., 2024]. To achieve broad applicability without complex prerequisites, we focus on imitation learning that operates under practical assumptions, relying solely on offline demonstrations without requiring interactive experts or simulators.

In this context, the framework of Continuity-based Corrective labels for Imitation Learning (CCIL) [Ke et al., 2023] has demonstrated promising results in simulations across multiple domains. CCIL combats compounding errors in imitation learning by generating corrective labels that bring the robot back to expert states. To generate such labels without the ground truth dynamics model, CCIL learns a dynamics model
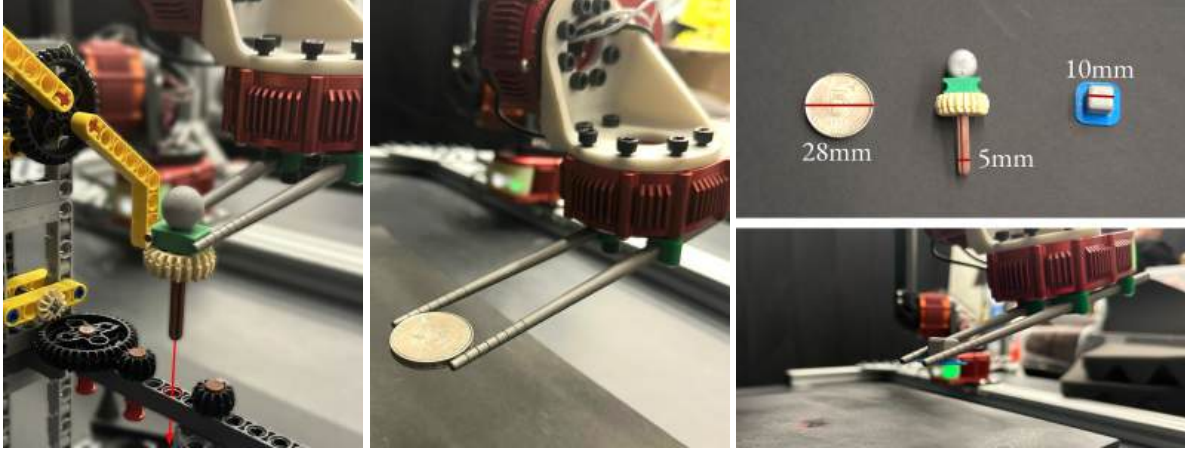
Figure 1.1: The three tasks we consider, *GearInsertion*, *GraspCoin*, and *GraspCube*, along with our three task objects: a coin, a Lego gear, and a small cube.

from the demonstrations and uses it to synthesize labels. Importantly, it leverages local Lipschitz continuity in the system dynamics, which allows fitting a dynamics model while having explicit guarantees on where the learned model has bounded errors, i.e., in the neighborhood near the demonstration data. This insight enables corrective label generation with theoretical guarantees, potentially alleviating compounding errors in imitation learning.

Despite CCIL's effectiveness in simulation domains, there is an *absence of empirical evidence* supporting its real-world applicability, and there remains a *gap in understanding its parameter choices for practical application*. Critically, CCIL assumes the presence of local Lipschitz continuity Sastry [2013] in the system dynamics—a physical property that can vary across tasks and datasets. Many tasks, such as manipulation, involve complex contact interactions that exhibit discontinuities in dynamics, raising doubts about CCIL's efficacy under such challenging real-world conditions. Therefore, research questions persist about required design decisions, parameter choices, and tuning procedures to enable its success.

Our work aims to answer these questions by comprehensively analyzing CCIL's performance on real-world fine manipulation tasks. Through extensive ablation studies, we explore CCIL's assumptions and examine the impact of design choices on its practicality. These studies give rise to our key insight, namely, that *even in contact-rich manipulation domains with varying discontinuities in dynamical systems, it is feasible and practical to use CCIL to generate corrective labels in the sub-space that contains local continuity*. To this end, we propose a practical instantiation of the CCIL framework tailored to identify and fit dy-

namics models for domains with mixed local Lipschitz continuity. Our ablation study tests vary controlled variables—i.e., data quantity, dynamics model learning, and hyperparameter sensitivity—to produce practical guidelines for applying CCIL. Our experiments provide *empirical evidence that generating corrective labels using the CCIL framework significantly benefits imitation learning*.

We focus on the domain of fine manipulation, an area that encompasses a wide range of real-world applications, from manufacturing to assistive robotics [Sakurai et al., 2016; Joseph et al., 2010; Yamazaki and Masuda, 2012]. Fine manipulation tasks are notably challenging even for human experts due to their demand for precise control and susceptibility to failure from minor inaccuracies, resulting in limited data availability. However, learning from demonstration is often preferred over model-based methods for many fine manipulation tasks with hard-to-define cost functions (e.g., stitching wounds). Thus, the inherent difficulties of fine manipulation underscore a critical need for strategies to mitigate compounding errors in imitation learning, making it an ideal testing ground.

In summary, we contribute:

- Experiments that demonstrate CCIL's capability to improve the performance of imitation learning agents on real-world fine manipulation tasks with limited data availability. CCIL boosts the success rate for GraspCube from 23% to 83%, for GearInsertion from 58% to 72%, and for GraspCoin from 17% to 48%.

- Experiments with varying availability of expert data that validate with statistical significance the performance boost CCIL offers in low-data regimes.

- Extensive ablation tests that analyze how design choices and hyperparameters affect the local Lipschitz continuity and label errors, key conditions for CCIL to succeed. Our insights provide practical guidance on choosing Lipschitz constraints for training dynamics and choosing error bounds to generate corrective labels.

Chapter 2

Related Work

Imitation Learning and Behavior Cloning. *Imitation learning* can enable robots to mimic complex behaviors from expert demonstrations [Billard et al., 2008; Osa et al., 2018]. *Behavior cloning* (BC) [Pomerleau, 1988] is a simple and practical imitation learning method that requires only demonstration data to learn a robotics policy, remaining a strong candidate for practical application and therefore gaining popularity over the years [Bojarski et al., 2016; Zhang et al., 2018; Kalashnikov et al., 2018; Rhinehart et al., 2018; Chi et al., 2023]. Empirical studies have reported performance gaps between simulation and real world results and highlighted the importance of real-world evaluation [Mandlekar et al., 2021; Zhou et al., 2023b].

Covariate Shift and Data Augmentation. Compounding errors in robot execution can cause agents to encounter unexpected states from expert demonstrations [Osa et al., 2018]. Various works that address covariate shift often request additional information, including data from interactive experts [Ross et al., 2011b; Laskey et al., 2017; Sun et al., 2017], environmental samples [Ho and Ermon, 2016; Ke et al., 2021a; Fu et al., 2017], alternative sources [Chang et al., 2021] and domain knowledge [Ke et al., 2021b; Zhou et al., 2023a; Chen et al., 2023; Block et al., 2024; Ke et al., 2023]. To achieve broad applicability without such complex prerequisites, we focus on imitation learning that operates under minimal assumptions, relying solely on offline demonstrations without requiring interactive experts or simulators. We focus on CCIL that makes a light assumption about the presence of local Lipschitz continuity. Despite promising results, there is scant empirical evidence for CCIL's real-world applicability. We aim to provide this evidence.

Local Lipschitz Continuity in Dynamics. Classical control methods and modern robot applications [Bon-

nard et al., 2007; Li and Todorov, 2004; Kahveci, 2007] often assume local Lipschitz continuity in system dynamics to guarantee solution existence and uniqueness. However, they operate with pre-specified models and cost functions. Most works on learning dynamics from data neither leverage nor enforce the local Lipschitz continuity constraint [Hafner et al., 2019; Wang et al., 2022; Kaufmann et al., 2023]. In contrast, this work focuses on learning locally continuous dynamics models from data without requiring human-specified models. While some prior works have enforced Lipschitz constraints during modeling of training dynamics [Shi et al., 2019; Pfrommer et al., 2021], their application focuses on global continuity assumptions or simulation scenarios. It remains unclear how to learn dynamics models that capture the complex contacts and high precision requirements of real-world fine manipulation tasks while achieving local Lipschitz continuity, a key focus of our exploration in this paper.

Chapter 3

Continuity-Based Corrective Labels

This section describes the framework we use to train imitation learning policies and highlights the design choices we made. Our training framework uses behavior cloning [Pomerleau, 1988] and Continuity-based Corrective Labels for Imitation Learning (CCIL). *Behavior cloning* is an imitation learning algorithm that trains a policy from expert demonstrations, and *CCIL* is a data augmentation framework to enhance the robustness of imitation learning agents. Our derivation follows the existing CCIL framework, but we provide a self-contained instantiation in Section 3.2. Choosing variants and parameters for this framework is largely domain-specific and can be costly to tune in the real world. We detail our practical instantiation in Section 3.3.

3.1   Notation for Imitation Learning and Behavior Cloning

We consider a finite-horizon Markov Decision Process (MDP), $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, f, P_0\}$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, f is the ground truth dynamics function, and P_0 is the initial state distribution. A *policy* maps a state to a distribution of actions $\pi : s \rightarrow a$. We assume a *deterministic* world dynamics function, where f maps a state s_t and an action a_t at time t to the change in state such that the next state is $s_{t+1} = s_t + f(s_t, a_t)$. Following the common setting in imitation learning, the true dynamics function f is unknown, and we have only expert demonstrations $\mathcal{D}^*$ as a collection of transition triplets: $\mathcal{D}^* = \{(s_j^*, a_j^*, s_{j+1}^*)\}_j$, where $s_{j+1}^* = s_j^* + f(s_j^*, a_j^*)$, i.e., for state s_j^* the expert provided an action label of a_j^*. Behavior cloning learns a policy from these traces by maximizing the likelihood of the expert actions

---

Algorithm 1 Our Instantiation of CCIL: **C**ontinuity-based **C**orrective labels for **I**mitation **L**earning

---

1: **Input:** $\mathcal{D}^* = (s_i^*, a_i^*, s_{i+1}^*)$
2: **Initialize:** $D^{\mathcal{G}} \leftarrow \varnothing$
3: `// Learn Dynamics`
4: $MSE \leftarrow \mathbb{E}_{(s_i^*, a_i^*, s_{i+1}^*) \sim \mathcal{D}^*} \left[\hat{f}(s_i^*, a_i^*) + s_i^* - s_{i+1}^* \right]$
5: $\hat{f} \leftarrow \arg\min_{\hat{f}} MSE$ s.t. $\|W\|_2 \leq L$
6: `// Generate Labels`
7: **for** $i = 1..n$ **do**
8:     $(s_i^{\mathcal{G}}, a_i^{\mathcal{G}}) \leftarrow \text{GenLabels}(s_i^*, a_i^*, s_{i+1}^*)$
9:     **if** $\|J_{\hat{f}}(s_i^*, a_i^*)\|_2 \cdot \|s_i^{\mathcal{G}} - s_i^*\| < \epsilon$ **then**
10:         $\mathcal{D}^{\mathcal{G}} \leftarrow \mathcal{D}^{\mathcal{G}} \cup (s_i^{\mathcal{G}}, a_i^{\mathcal{G}})$
11:     **end if**
12: **end for**
13: $\mathcal{D} \leftarrow \mathcal{D}^* \cup \mathcal{D}^{\mathcal{G}}$
14: `// Learn Policy`
15: LearnPolicy π
16: **Function** GenLabels
17:     $a_i^{\mathcal{G}} \leftarrow a_i^*$
18:     $s_i^{\mathcal{G}} \leftarrow s_i^* - \hat{f}(s_i^*, a_i^*)$
19: **Function** LearnPolicy
20:     $L(a, \hat{a}) \leftarrow$ policy loss function, as defined in Eq 4.1
21:     $\pi = \arg\min_{\pi} \mathbb{E}_{(s_i, a_i) \sim \mathcal{D}} [L(a_i, \pi(s_i))]$

---

being produced at the expert states: $\arg\max_{\hat{\pi}} \mathbb{E}_{s_j^*, a_j^*, s_{j+1}^* \sim \mathcal{D}^*} \log(\hat{\pi}(a_j^* \mid s_j^*))$.

3.2   Continuity-based Corrective Labels

CCIL aims to reduce compounding errors in imitation learning by generating corrective labels that guide the robot back to expert-visited states and recovery from mistakes. For each transition in expert data, $(s_t^*, a_t^*, s_{t+1}^*)$, CCIL proposes to generate some state-action pair $(s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$ to bring the agent to expert state s_t^*. With a known dynamics model f, we can sample labels and query f to ensure $s_t^{\mathcal{G}} + f(s_t^{\mathcal{G}}, a_t^{\mathcal{G}}) - s_t^* \to 0$. Without it, CCIL must first train a dynamics model.

Training a Dynamics Model. We can train a dynamics model by minimizing the following loss:

$$\mathbb{E}_{(s_t^*, a_t^*, s_{t+1}^*) \sim \mathcal{D}^*} \left[\hat{f}(s_t^*, a_t^*) + s_t^* - s_{t+1}^* \right]. \tag{3.1}$$

A learned dynamics model can yield reliable predictions only near its data support, not on arbitrary states

12

and actions. CCIL decides where to query the learned dynamics models by leveraging the presence of local Lipschitz continuity in the system dynamics, as specified in Definition 3.1. If (s^*, a^*) belongs to a space with local Lipschitz continuity, given small changes in the input state or actions, the resultant changes in the system's state are bounded and predictable to a degree, as follows.

Definition 3.1 (Local Lipschitz Continuity). A function f is *locally Lipschitz continuous* around (s, a) with coefficient K if there exists a neighborhood of (s, a) of size δ such that for every $(\tilde{s}, \tilde{a})$ where $\|(s, a) - (\tilde{s}, \tilde{a})\| \le \delta$:

$$\|f(s, a) - f(\tilde{s}, \tilde{a})\| \le K\|(s, a) - (\tilde{s}, \tilde{a})\|. \tag{3.2}$$

CCIL encourages the learned dynamics function to exhibit local Lipschitz continuity by modifying the training objective. We choose spectral normalization regularization and elaborate on the reasons in Sec. 3.3. Specifically, to train a dynamics model $\hat{f}$ using a neural network of n-layers with weight matrices $W_1, W_2, \ldots, W_n$, we can iteratively minimize the training objective (Eq. 3.1) while regularizing the model by setting

$$W_i \leftarrow \frac{W_i}{\max(\|W_i\|_2, K^{-n})} \cdot K^{-n} \tag{3.3}$$

for all W_i, where K is the desired Lipschitz constant.

Label Generation. With a learned dynamics model $\hat{f}$, CCIL attempts to find a corrective label $(s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$ for each expert data point (s_t^*, a_t^*) such that $s_t^{\mathcal{G}} + \hat{f}(s_t^{\mathcal{G}}, a_t^{\mathcal{G}}) \approx s_t^*$. We consider the BackTrack label generator inspired by Backward Euler used in modern simulators:

$$\begin{aligned} s_t^{\mathcal{G}} &\leftarrow s_t^* - \hat{f}(s_t^*, a_t^*) \\ a_t^{\mathcal{G}} &\leftarrow a_t^* \end{aligned} \tag{3.4}$$

Filtering Generated Labels by Label Distance. The generated label $(s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$ aims to bring the agent to some expert state s_t^*. Its error on the true dynamics is $\|f(s_t^{\mathcal{G}}, a_t^*) - \hat{f}(s_t^{\mathcal{G}}, a_t^*)\|$. CCIL provides a bound for this error, as follows.

Theorem 3.2. *When the dynamics model has a bounded training error ϵ on the training data, if the learned dynamics $\hat{f}$ and true dynamics f are respectively locally K_1 and K_2-Lipschitz within some neighborhood*

of (s_t^*, a_t^*) of size δ, and $\|s_t^{\mathcal{G}} - s_t^*\| < \delta$, then

$$\left\| f\left(s_t^{\mathcal{G}}, a_t^*\right) - \hat{f}\left(s_t^{\mathcal{G}}, a_t^*\right) \right\| \leq \epsilon + (K_1 + K_2)\left\| s_t^{\mathcal{G}} - s_t^* \right\|. \tag{3.5}$$

To ensure the generated labels are of low error bound, CCIL sets a hyperparameter δ, denoting the desired size of the neighborhood for local Lipschitz continuity. It filters out generated labels outside the neighborhood $\|s_t^{\mathcal{G}} - s_t^*\| > \delta$. Sec. 3.3 describes our modification to this filtering.

Learn Policy. A straightforward way to utilize the generated labels is to add them to the demonstration dataset to train any imitation learning agent. For simplicity, we consider the following behavior cloning objective to train a policy:

$$\pi = \arg\min_{\pi} \mathbb{E}_{(s_i, a_i) \sim \mathcal{D}} \left[L(a_i, \pi(s_i)) \right]. \tag{3.6}$$

3.3 Our Practical Instantiation of CCIL

CCIL's effectiveness critically hinges on (1) training a dynamics model for the locally continuous state-action space, and (2) using it to generate labels with reasonable error bounds. We recommend simple design choices based on practical insights. Alg. 1 summarizes our approach.

Training Dynamic Models. CCIL requires training a dynamics model that contains local Lipschitz continuity and proposes a few candidate loss functions. We aim to achieve this goal using only spectral normalization regularization. We highlight that the local Lipschitz constant K varies not just between domains, but also between different state-actions in the same domain. Regardless of the training objective, as long as $\hat{f}$ is modeled via a neural network, we can measure the local Lipschitz constant at any state-action by calculating the local Jacobian matrix $J_{\hat{f}}$ as follows.

Remark 3.3. For a continuous function $\hat{f}$, the local Lipschitz constant in a neighborhood around a given point (s, a) is

$$K(s, a) \approx \|J_{\hat{f}}(s, a)\|_2. \tag{3.7}$$

Training with spectral normalization imposes an *upper bound* on the Lipschitz constant. The learned model will still exhibit local Lipschitz continuity with varying Lipschitz constants at different state-actions. In practice, we can set a loose upper bound on the local Lipschitz continuity and still achieve local Lipschitz

14

continuity at various regions in the state space, thereby satisfying the condition required by CCIL.

Filtering Generated Labels by Label Error. CCIL proposes to filter out generated labels $(s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$ if the label distance $\|s_t^{\mathcal{G}} - s_t^*\|$ violates a set threshold $\|s_t^{\mathcal{G}} - s_t^*\| > \delta$. This implicitly assumes that the size of the neighborhood of local Lipschitz continuity stays constant throughout the state space, making it challenging to tune the threshold δ. We refine this rejection process, as indicated in line 9 of Alg. 1.

The error bound for each generated label (Theorem 3.2) depends not only on the label distance but also on the training error ϵ and the local Lipschitz coefficients of $\hat{f}$, K_1. To measure ϵ and K_1 (Remark 3.3), we can calculate an error bound for each generated label. For a synthetic label $(s_t^{\mathcal{G}}, a_t^{\mathcal{G}})$ generated from expert demonstrations (s_t^*, a_t^*), knowing $a_t^{\mathcal{G}} = a_t^*$ and assuming $\epsilon \approx 0$ and $K_2 \propto K_1$, we can set a threshold on the acceptable label error bound σ and reject the label if

$$\|J_{\hat{f}}(s_t^*, a_t^*)\|_2 \cdot \|s_t^{\mathcal{G}} - s_t^*\| \geq \sigma. \tag{3.8}$$

In practice, we first generate all labels and examine the distribution of label error bounds. We then define σ to be some quantile of the distribution. We effectively compute a "trust" region around the expert data to generate labels, and the size of the region varies for each data point. We visualize this region in empirical experiments in Fig. 5.2b.

Chapter 4

Experimental Design

4.1   Motivation

The CCIL framework has shown notable success in various simulation domains. However, its application to real-world scenarios hinges on several critical research questions:

Local Continuity in Dynamics and Real-World Application. CCIL relies on local Lipschitz continuity in system dynamics, yet real-world robotic tasks often involve discontinuities due to physical contact. Can this foundational assumption be validated in realistic domains? Can CCIL still enhance imitation learning agent performance amidst such challenges?

Data Scarcity and Augmentation Impact. Limited real-world robot demonstrations highlight the importance of efficient data augmentation. While abundant data typically guarantees better outcomes, the critical question lies in the low-data regime: How effectively can CCIL address data scarcity, and what data volume is required to observe a tangible impact?

Sensitivity to Hyperparameters. Tuning imitation learning algorithms on physical robots is logistically challenging. Direct execution on robots, while being the most reliable evaluation method, risks unpredictable behaviors and necessitates numerous real-world trials to achieve statistical significance. How sensitive is CCIL to hyperparameter variations? Can we offer guidelines for its real-world application that mitigate the high evaluation and tuning costs?
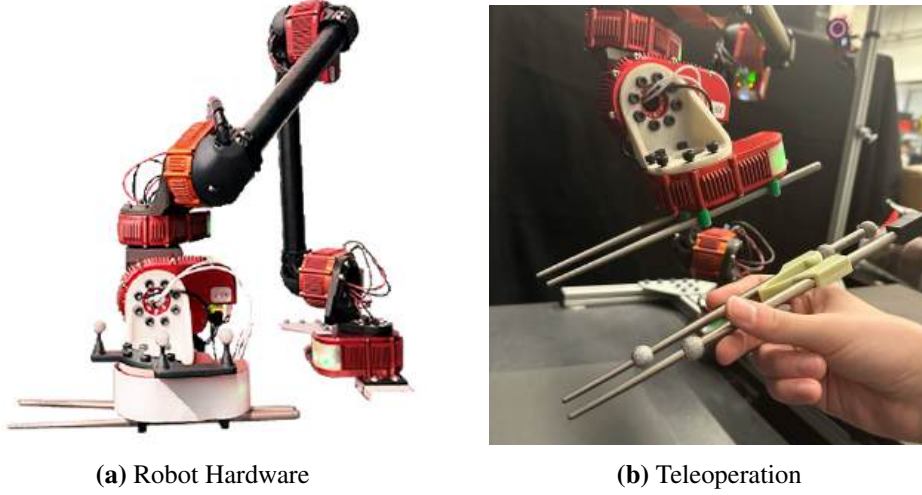
(a) Robot Hardware   **(b)** Teleoperation

Figure 4.1: System overview. (a) Our HEBI-based 7-DOF robot with a chopstick end-effector. (b) Teleoperation mimicking leader chopsticks tracked using a motion-capture cage.

4.2 Hypotheses

We explore the following hypotheses:

- **H1:** CCIL can improve imitation learning policies in real-world fine-manipulation tasks. This improvement is statistically significant, especially in low-data regimes.

- **H2:** Real-world tasks with complex contacts and discontinuities in dynamics can still exhibit local Lipschitz continuity to justify CCIL's assumptions.

- **H3:** The performance of CCIL is highly sensitive to the label rejection hyperparameter, which determines the acceptable error bound for generated labels.

- **H4:** The performance of CCIL exhibits relative robustness to variations in the local Lipschitz constraint enforced during the training of the dynamics model.

To evaluate **H.1**, we compare the success rate of a behavior cloning agent trained with and without CCIL-generated labels. We vary the amount of demonstration data to investigate the impact of data quantity on these methods.

To evaluate **H.2**, we measure the local Lipschitz continuity of the trained dynamics model for tasks with complex discontinuity. We also vary the hyperparameter of the upper bound on the local Lipschitz constant.

17

To evaluate **H.3** and **H.4**, we run ablation studies varying the hyperparameters of the label rejection threshold and the Lipschitz constraint to investigate (1) their impact on the success rate of CCIL, and (2) the interplay between these hyperparameters.

4.3   Hardware

We conducted experiments on a fine manipulator platform introduced in Ke et al. [2021b] and shown in Fig. 4.1a. The robot has 7 joints and is equipped with a pair of chopsticks as its end-effector. It runs a hierarchical controller, where the high-level command consists of a 6-DOF end-effector pose target plus 1-DOF for the last joint to open or close the gripper chopsticks. A policy sends high-level commands at 20Hz. The robot computes inverse kinematics to send each joint a low-level positional command, which is tracked by a positional PID controller running at 1KHz. For perception, we use a motion capture system. We follow Ke et al. [2020] to also collect human teleoperation demonstrations (Fig. 4.1b).

4.4   Tasks and Data Collection

We consider three fine manipulation tasks that require a millimeter level of precision.

1. *GraspCube*: Grasps and lifts a tiny 1cm cube above the table. Despite being the easiest of all three tasks, the task is non-trivial with scarce data (e.g., 100 trajectories).

2. *GearInsertion*: Inserts a Lego gear into a hole on a board. This mini-gear insertion task (Fig. 1.1) is inspired by industrial assembly Gubbi et al. [2020]. Proper insertion leaves a gap of less than 0.2 mm.

3. *GraspCoin*: Uses chopsticks to grasp and lift a metal coin lying flat on the table. The coin's thin round shape and slippery texture makes it difficult even for human experts to pick up using chopsticks.

For all tasks, we vary the initial positions of the object. To collect demonstrations, we use a mix of heuristic controllers and human teleoperation. For *GraspCube* and *GearInsertion*, we designed heuristic controllers and collected 500 trajectories for *GraspCube* and 100 trajectories for *GearInsertion*. Note that the heuristic controllers did not have perfect success rates, and we filtered out failed demonstrations. For *GraspCoin*, designing a heuristic controller was complicated due to the difficulty of the task, so we instead used successful demonstrations from an expert teleoperation to obtain 200 trajectories.

4.5 Training

For each task, we train two behavior cloning agents, with and without CCIL-generated labels. The training follows Alg. 1, and we formulate the action loss for our hardware and detail the parameter tuning procedure below.

Loss Formulation. To train a policy using behavior cloning following Eq. 3.6, we need to design an action loss $L(a^*, \hat{a})$. We denote the action a for our hardware as (x, q, c), which represents the end-effector xyz location, orientation, and chopstick angle. Given the target action $a^* = (x, q, c)$ and the predicted action $\hat{a} = (\hat{x}, \hat{q}, \hat{c})$, the action loss is:

$$L(a, \hat{a}) = \alpha_1 \|x - \hat{x}\|^2 + \alpha_2 \theta^2 + \alpha_3 (c - \hat{c})^2, \tag{4.1}$$

where θ is the angle between the orientations q and $\hat{q}$ and $\alpha_1, \alpha_2, \alpha_3$ weight each component of the loss function. In our experiments, we use $\alpha_1 = 10, \alpha_2 = 1, \alpha_3 = 10$.

Parameter Tuning. Applying CCIL introduces two key parameters: (1) the Lipschitz constraint for training the dynamics model, and (2) a threshold for filtering the generated labels. The Lipschitz constraint, K (Eq. 3.3), upper bounds the Lipschitz continuity of the learned dynamics model. A loose upper bound makes the training process easier and more likely to yield low training error. Conversely, a tighter bound is more likely to yield a dynamics model with lower Lipschitz coefficients, enabling CCIL to generate higher quality corrective labels. The label error bound is defined in Theorem. 3.2, and the threshold σ controls the acceptable bound.

To set the Lipschitz constraint K, we first train an unconstrained model and analyze the distribution of local Lipschitz coefficients on the learned model. If this distribution suggests that the learned model has limited local Lipschitz continuity, we choose a tighter Lipschitz constraint.

To set threshold σ, we first generate corrective labels without filtering and analyze the distribution of label errors. Based on the distribution, we choose a label rejection quantile (between 0 to 1, where 0 means to filter out all generated labels) that filters out outliers or long tails.

4.6　Evaluation

We test each agent's success rate by conducting 48 trials per agent to establish statistical significance. To ensure fairness, for each task we select 16 fixed initial conditions (i.e., for grasping agents, we place the object to grasp at 16 fixed positions across the workspace). For each initial condition, we test the learned policy for 3 trials. For each trial, we denote the success as a binary variable. To report the statistical significance of the empirical results, we compare the success rates between the two policies by performing two-proportion z-tests.

Chapter 5

Results

Across three tasks, we trained 36 agents and tested their success rate in the real world (Fig. 5.1a). For ablation, we further evaluated 33 agents on the GraspCube task.

5.1 Corrective Labels' Improvement to Imitation Learning

H.1 motivates the investigation of whether CCIL can improve imitation learning performance in real-world fine-manipulation tasks. These tasks involve complex contact dynamics between objects, end-effectors, and the workspace, making it unclear if CCIL's assumptions hold.

CCIL yields an increase in performance for behavior cloning. Fig. 5.1a shows that training with labels generated by CCIL generally improves the performance of behavior cloning over all three tasks, as measured by success rate. We validated the statistical significant of the improvement. The performance boost is significant at the $p < 0.05$ level for the cube grasping and coin grasping tasks.

The CCIL boost is significant in the low-data regime. The empirical performance gain from applying CCIL is statistically significant in low-data regimes, as shown in Fig. 5.1a. CCIL boosts behavior cloning performance from 23% to 83% (using 100 GraspCube trajectories), from 6% to 17% (using 20 GearInsertion trajectories) or 58% to 72% (using 100 GearInsertion trajectories), and from 17% to 48% (using 200 GraspCoin trajectories). These limited-data regimes face substantial challenges from covariate shift due to limited data support from expert demonstrations. CCIL demonstrated promising results to alleviate this problem.
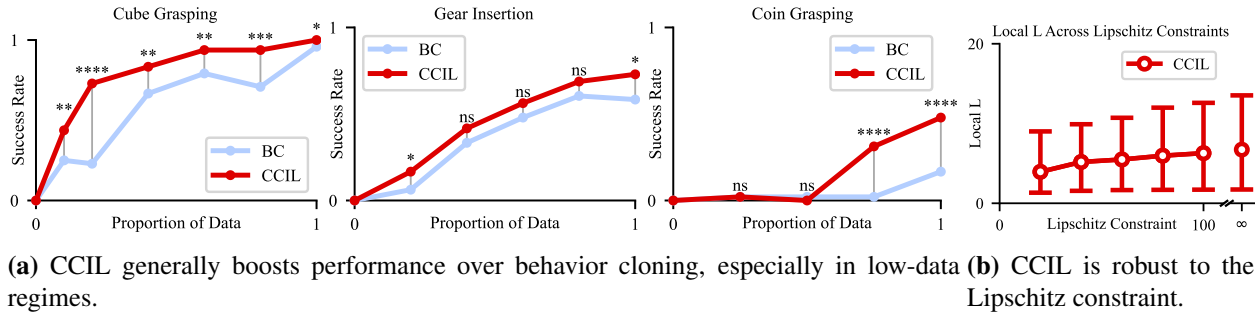
(a) CCIL generally boosts performance over behavior cloning, especially in low-data regimes.

(b) CCIL is robust to the Lipschitz constraint.

Figure 5.1: (a) As the amount of data increases, CCIL provides a smaller boost over behavior cloning. We use asterisks to denote statistical significance: * p < 0.1, ** $p < 0.05$, *** $p < 0.01$, and **** $p < 0.001$. ns denotes $p \geq 0.1$. (b) Mean and middle 95% of the local Lipschitz coefficients of the learned dynamics model across the demonstration dataset as the Lipschitz constraint increases. As the enforced constraint increases, the distribution of coefficients converges.
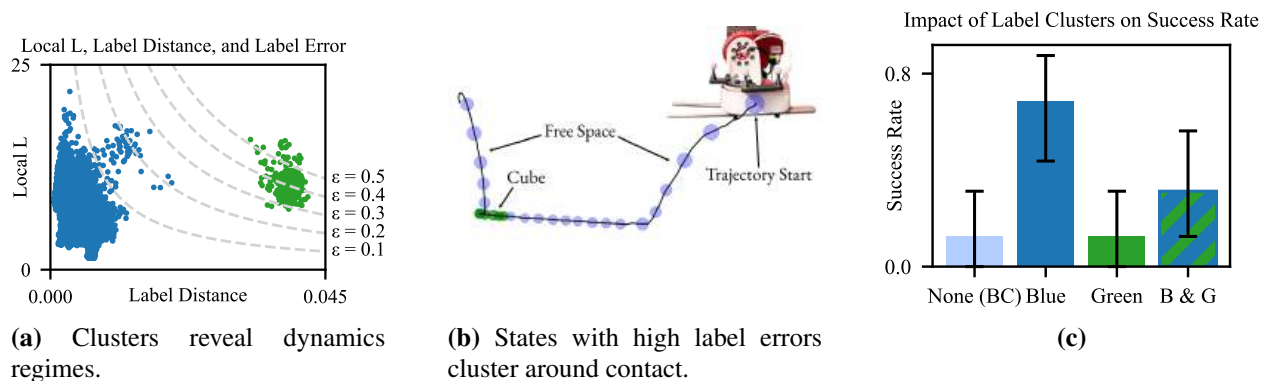

(a) Clusters reveal dynamics regimes.

(b) States with high label errors cluster around contact.

(c)

Figure 5.2: (a) The 20% dataset for the *GraspCube* task reveals two distinct clusters of corrective labels. (b) The green cluster mainly corresponds to labels where the cube is being manipulated, and the blue cluster to arm-free space motion. (c) Policies trained using just the blue cluster (low label rejection threshold) are more successful compared to those trained with the green cluster or both (high label rejection threshold.

Remark. Our experiments validate how CCIL can be applied to complex fine manipulation tasks in the real world despite the presence of discontinuity in the contact dynamics.

5.2 CCIL's Assumptions on Local Lipschitz Continuity

H.2 motivates the investigation of whether CCIL's local continuity assumptions are sufficiently valid and realistic to apply to real-world fine manipulation tasks.

Real-world tasks contain local Lipschitz continuity that can be realized by learning dynamics models. In training the dynamics model, we experiment with different Lipschitz constraints, measure the

resulting models' local Lipschitz coefficient, and plot the average of the coefficient in Fig. 5.1b. With a large upper bound on the Lipschitz constraint (loose constraint), the local coefficients increase but converge to those of an unconstrained model. These findings imply that our environments exhibit local continuity that the learned dynamics models are fitting to, even without explicitly enforcing the Lipschitz continuity coefficient.

The learned dynamics model and generated labels have varying continuity that correlates with the real world. For each generated label, we can measure the local Lipschitz coefficients using the learned dynamics model ("Local L"). We generate a scatterplot for the generated labels in Fig. 5.2a and see two clusters (green and blue). We then sample some of the labels for visualization and plot them along an expert trajectory in Fig. 5.2b. We observe that the green cluster corresponds to labels generated near the cube, and the blue labels are mainly in free space when the robot moves without collision. Intuitively, the learned model and generated label have higher errors for the green group that contains discontinuity.

We test how policy performance changes as we incorporate into the demonstration dataset (1) blue labels, (2) green labels, and (3) both labels, as shown in Fig. 5.2c. Training with blue labels improves the agent's success rate. However, training with the green labels associated with contact-rich states and higher errors is not as useful. Such labels actually diminish performance. This highlights how label quality can impact CCIL performance and the significance of filtering generated labels.

5.3 Impact of the Quality of Generated Labels

The label filtering threshold controls what generated labels are used when training a policy. It is therefore crucial to understand its effects and how to tune it properly. We propose a practical way to enforce a filtering threshold via label rejection quantile; **H.3** motivated exploring how this design choice affects CCIL's performance, which we do below.

We choose the GraspCube task with 100 trajectories. We evaluate running CCIL with different label rejection quantiles and report the success rate in Fig. 5.3a. The rejection quantile controls how many generated labels are used (25%, 75%, etc.) based on their computed error bound.

Low-quality labels can harm policy performance. Using all generated labels (i.e., choosing a quantile close or equal to 1) makes CCIL performance similar to or worse than behavior cloning. With this quantile,
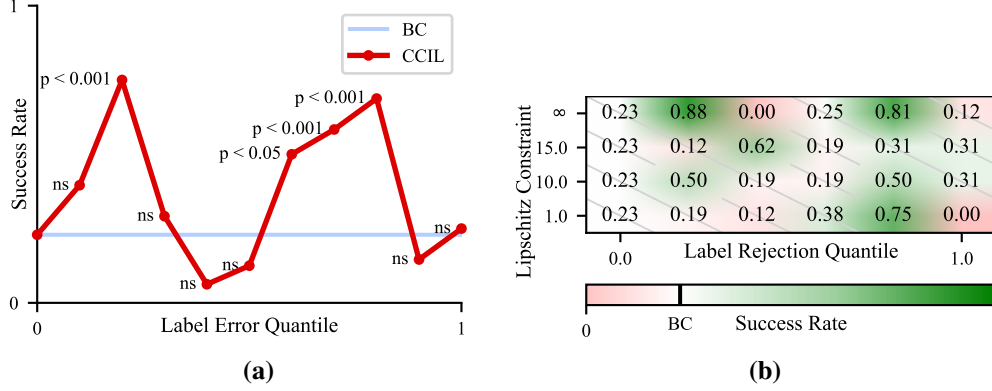
Figure 5.3: **(a)** Policy performance when filtering out varying fractions of the generated labels due to label error. **(b)** Hyperparameter ablation for CCIL, varying the Lipschitz constraint and the label rejection quantile. Green cells indicate a performance boost over behavior cloning, while red cells show worse performance. Crossed out cells are not significant at the $p < 0.05$ level, while the others are.

the generated labels have high error bounds and could be incorrect under the true dynamics. If these labels are included when training the policy, they could potentially diminish policy performance.

There is a need to balance label error and usefulness for CCIL to succeed. Labels with lower error bounds are more trustworthy. However, these labels tend to be closer to the demonstration data (small label distance). Including labels with conservative error bounds may fail to expand the data support. Conversely, a label that is further from the demonstration data can expand the data support while potentially introducing a higher associated error. Fig. 5.3a shows intermediate values of the label error quantile that achieve a higher success rate, indicating the need to balance label accuracy and usefulness to maximize CCIL's performance.

5.4 Impact of Continuity Constraint

H.4 asks how enforcing Lipschitz continuity for training the dynamics model impacts CCIL's performance.

We train the dynamics model using different Lipschitz constraints; for each model, we generate labels and test the performance of CCIL with various label error quantiles. All hyperparameters other than those ablated upon are fixed. We show the resulting success rate in Fig. 5.3b, crossing out the cells whose performance is not statistically significant compared to behavior cloning.

CCIL is robust to different Lipschitz constraints. Interestingly, we see that for each Lipschitz constraint, there is a label rejection quantile that achieves a significant boost in performance over behavior

cloning. This indicates that CCIL could work well for a wide range of constraints, making it robust to this parameter.

CCIL can work with an unconstrained dynamics model. Surprisingly, even using an unconstrained dynamics model, as indicated by the Lipschitz constraint of ∞, CCIL still performs well given a certain quantile. We investigate this phenomenon in Fig. 5.1b and observe that the unconstrained model still exhibits local continuity over the demonstration data, as shown by the relatively small local Lipschitz coefficients across the dataset. Interestingly, we observe that as the Lipschitz constraint increases, the distribution of local Lipschitz coefficients converges to that of the unconstrained model.

Remark. These findings together imply that in environments with sufficient local continuity, explicit Lipschitz constraint enforcement may be unnecessary and tuning the label rejection quantile can be more critical. When applying CCIL to new environments, we can start with unconstrained dynamics models for their simplicity.

Chapter 6

Conclusion

This paper demonstrated the effectiveness of corrective labels for offline Imitation Learning through real-world fine-manipulation experiments, highlighting its impact in the low-data regime. Our results validate that the continuity assumptions underlying CCIL are applicable even in the presence of discontinuous contact dynamics. Furthermore, we provide practical guidelines for choosing the continuity constraint and label quality threshold to enable CCIL's success. Future work can explore extending CCIL to high-dimensional state spaces, such as images, and investigating different policy classes, such as diffusion policies.

6.1 Limitations

CCIL is not without its drawbacks, however. Due to its minimal additional assumptions, CCIL does not provide any proven bounds on how its corrective labels will impact a policy's performance. Although these labels boost performance empirically, our experiments show that it is nontrivial to predict how different labels will impact a policy. This can be explained by the mismatch between the policy's test-time state distribution and that of the expert demonstrations. CCIL aims to expand the coverage of the expert demonstrations, but in the purely offline setting, we have no way of knowing the shift between this expanded distribution and the policy's true test-time distribution.

6.2   Future Work

This work, to our knowledge, marks the first deep investigation of continuity-based corrective labels for Imitation Learning on real-world platforms. However, there still remain many new and exciting avenues to further explore this problem setting.

6.2.1   Provable improvement of covariate shift

Understanding when CCIL is able to effectively decrease the shift between the policy's and expert's state distributions would allow CCIL to be applied with confidence to many new environments. This understanding is difficult to establish with the minimal assumptions associated with CCIL, but prior work [Laskey et al., 2017] has shown that training the policy in parallel with collecting disturbed demonstrations from the expert can provably decrease the divergence between the policy and expert distributions.

Following a similar method does not require a fully queryable expert, which would enable even stronger methods [Ross et al., 2011b], but instead assumes sample access to trajectories from a noised expert. By adopting this approach and generating corrective labels in addition to the noised samples from the expert, we may be able to show stronger guarantees about the divergence between distributions. Intuitively, we seek to expand the expert state distribution by combining disturbed demonstrations with the cheaply generated continuity-based corrective labels offered by CCIL.

Such a method would help us measure how well CCIL can expand the expert demonstrations' state distribution to match the learned policy, and provable bounds on that divergence would be invaluable to further understanding and expanding upon this framework.

6.2.2   Application to high dimensional states

We have shown that CCIL works well in real robotic tasks by exploiting the continuity in the system dynamics. This, however, relies on having access to a state representation with continuous properties. Images and other similar high-dimensional representations, for example, are very discontinuous with respect to the dynamics, even if the underlying system is very smooth. This issue fundamentally limits CCIL's applicability to these problem settings.

To address this, we need to extract a state representation with sufficient continuity properties from the

image data, which can then allow us to apply CCIL in the resulting latent space. We reason that when the underlying system itself is continuous, there exists some continuous state representation that is encoded within the image state space. Image embeddings for robotic tasks has been widely studied to great success Nair et al. [2022]; Radford et al. [2021], but importantly, these methods make no mention of continuity with respect to the system dynamics, which is crucial for CCIL to generate accurate labels. To achieve a suitable embedding of the high-dimensional image data, we can employ similar representation learning methods, while enforcing continuity constraints. By jointly training the image encoder and the latent dynamics model, we can encourage the encoder to learn an embedding that allows for accurate and continuous dynamics prediction in the latent space.

Enabling CCIL to generate corrective labels even with seemingly discontinuous and high-dimensional state representations would greatly increase its applicability. For example, this would allow for its usage in complicated environments with messy perception, where low-dimensional state cannot be accurately estimated. CCIL can improve data efficiency in these domains, making Imitation Learning much more robust and reliable in the real world.

6.2.3   Application to time-chunking policies

In this work, we explored how CCIL can increase the robustness and data efficiency of behavior cloning methods leveraging an MLP architecture. Recent works [Chi et al., 2023; Zhao et al., 2023] have shown that alternative policy architectures, leveraging observation and action sequences, can be applied to Imitation Learning, resulting in performance gains in real-world robotic tasks. Applying CCIL to these novel policy architectures would be valuable to help understand if and how CCIL can improve the data efficiency and robustness of these methods.

Due to these methods use of observation chunking, CCIL cannot be directly applied as originally formulated [Ke et al., 2023]. This is due to the fact that the proposed label generation algorithms can only show how the policy can recover from out-of-distribution states back to the support of the demonstration data. This works well for one-step policies, but for those that use observation sequences, we must also show the sequence of states that brought the policy to the out-of-distribution state in the first place. We can do this by generating this out-of-distribution state with our learned dynamics model. By disturbing the actions

in the expert demonstrations, we can generate new out-of-distribution data, which we can then guide back to the support of the demonstration data using the the learned model. If the disturbances are small enough to stay within the region of local Lipschitz continuity around the demonstration data, this label generation algorithm can generate synethic observation sequences, which are suitable for use in observation chunking models.

This enhancement to CCIL will enable its application to a wider variety of policy classes. This will highlight CCIL's generalizability and broad applicability to many new policy architectures, allowing CCIL to inherit their associated advantages.

Bibliography

Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal. 2008. Survey: Robot programming by demonstration.

Aude Billard and Daniel Grollman. 2012. Imitation learning in robots. *Encyclopedia of the Sciences of Learning*.

Adam Block, Ali Jadbabaie, Daniel Pfrommer, Max Simchowitz, and Russ Tedrake. 2024. Provable guarantees for generative behavior cloning: Bridging low-level stability and high-level behavior.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars.

Bernard Bonnard, Jean-Baptiste Caillau, and Emmanuel Trélat. 2007. Second order optimality conditions in the smooth case and applications in optimal control. *ESAIM: Control, Optimisation and Calculus of Variations*.

Nicholas J Bryan. 2019. Data augmentation and deep convolutional neural networks for blind room acoustic parameter estimation.

Jonathan Chang, Masatoshi Uehara, Dhruv Sreenivas, Rahul Kidambi, and Wen Sun. 2021. Mitigating covariate shift in imitation learning via offline data with partial coverage.

Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. 2023. Genaug: Retargeting behaviors to unseen situations via generative augmentation.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*.

Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning robust rewards with adversarial inverse reinforcement learning.

Sagar Gubbi, Shishir Kolathaya, and Bharadwaj Amrutur. 2020. Imitation learning for high precision peg-in-hole tasks. In *ICCAR*.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2019. Dream to control: Learning behaviors by latent imagination.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *NeurIPS*.

Rohan A Joseph, Alvin C Goh, Sebastian P Cuevas, Michael A Donovan, Matthew G Kauffman, Nilson A Salas, Brian Miles, Barbara L Bass, and Brian J Dunkin. 2010. "chopstick" surgery: a novel technique improves surgeon performance and eliminates arm collision in robotic single-incision laparoscopic surgery.

Nazli E Kahveci. 2007. *Robust Adaptive Control For Unmanned Aerial Vehicles*. Ph.D. thesis.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. 2018. Scalable deep reinforcement learning for vision-based robotic manipulation. PMLR.

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. 2023. Champion-level drone racing using deep reinforcement learning. *Nature*.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. 2021a. Imitation learning as f-divergence minimization. Springer.

Liyiming Ke, Ajinkya Kamat, Jingqiang Wang, Tapomayukh Bhattacharjee, Christoforos Mavrogiannis, and Siddhartha S Srinivasa. 2020. Telemanipulation with chopsticks: Analyzing human factors in user demonstrations. IEEE.

Liyiming Ke, Jingqiang Wang, Tapomayukh Bhattacharjee, Byron Boots, and Siddhartha Srinivasa. 2021b. Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation. IEEE.

Liyiming Ke, Yunchu Zhang, Abhay Deshpande, Siddhartha Srinivasa, and Abhishek Gupta. 2023. Ccil: Continuity-based data augmentation for corrective imitation learning.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. 2017. Dart: Noise injection for robust imitation learning. PMLR.

Weiwei Li and Emanuel Todorov. 2004. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. 2021. What matters in learning from offline human demonstrations for robot manipulation.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. 2022. R3m: A universal visual representation for robot manipulation.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*.

Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. 2023. Open x-embodiment: Robotic learning datasets and rt-x models.

Samuel Pfrommer, Mathew Halm, and Michael Posa. 2021. Contactnets: Learning discontinuous contact dynamics with smooth, implicit representations. PMLR.

Dean A Pomerleau. 1988. Alvinn: An autonomous land vehicle in a neural network.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. 2018. Deep imitative models for flexible inference, planning, and control.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011a. A reduction of imitation learning and structured prediction to no-regret online learning. JMLR Workshop and Conference Proceedings.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011b. A reduction of imitation learning and structured prediction to no-regret online learning. JMLR Workshop and Conference Proceedings.

Haruka Sakurai, Takahiro Kanno, and Kenji Kawashima. 2016. Thin-diameter chopsticks robot for laparoscopic surgery. IEEE.

Shankar Sastry. 2013. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media.

Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. 2019. Neural lander: Stable drone landing control using learned dynamics. IEEE.

Jonathan Spencer, Sanjiban Choudhury, Arun Venkatraman, Brian Ziebart, and J Andrew Bagnell. 2021. Feedback in imitation learning: The three regimes of covariate shift.

Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. 2017. Deeply aggrevated: Differentiable imitation learning for sequential prediction. PMLR.

Arun Venkatraman, Byron Boots, Martial Hebert, and J Andrew Bagnell. 2014. Data as demonstrator with applications to system identification. In *ALR Workshop, NIPS*.

Jun Wang, Yuzhe Qin, Kaiming Kuang, Yigit Korkmaz, Akhilan Gurumoorthy, Hao Su, and Xiaolong Wang. 2024. Cyberdemo: Augmenting simulated human demonstration for real-world dexterous manipulation.

Tongzhou Wang, Simon S Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. 2022. Denoised mdps: Learning world models better than the world itself.

Akira Yamazaki and Ryosuke Masuda. 2012. Autonomous foods handling by chopsticks for meal assistant robot. VDE.

Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. 2018. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. IEEE.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. 2023. Learning fine-grained bimanual manipulation with low-cost hardware.

Allan Zhou, Moo Jin Kim, Lirui Wang, Pete Florence, and Chelsea Finn. 2023a. Nerf in the palm of your hand: Corrective augmentation for robotics via novel-view synthesis.

Gaoyue Zhou, Liyiming Ke, Siddhartha Srinivasa, Abhinav Gupta, Aravind Rajeswaran, and Vikash Kumar. 2023b. Real world offline reinforcement learning with realistic data source. In *ICRA*.