Enhancing Assistive Robotics with Simplified Shared Control via Imitation Learning

by

Riley James Zilka

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science
University of Alberta

Abstract

Autonomous manipulation systems have achieved remarkable capabilities, yet integrating human expertise with diffusion-based policies in shared control remains relatively unexplored. Additionally, finding efficient combinations of human input with autonomous policies remains an unsolved problem. Physically impaired individuals demonstrate a strong preference for maintaining control authority and personal agency in their daily lives. To strike this balance, this thesis proposes a novel diffusion-assisted shared control approach that reduces mental workload while maintaining an intuitive control interface and user autonomy.

We present two complementary contributions. First, Human-In-The-Loop Diffusion (HITL-D) is a shared control framework dividing the 6 degrees of Cartesian control into two distinct components. Users control end effector position via a joystick while a diffusion policy autonomously determines orientation based on scene point clouds and Cartesian position. In a multi-task user study with 12 participants, HITL-D reduced average task completion times by 40%, decreased perceived workload by 37%, and improved Likert-scale ratings for independence, intuitiveness, and confidence compared to traditional teleoperation methods.

Second, we address the generalization limitations of shared control systems by proposing Human-In-The-Loop Highly Generalizable Diffusion (HITL-HGD), which leverages privileged spatial information via prompted image segmentation to condition the diffusion policy. Using alignment constraints and object

centroids, we create an autonomous system that encodes implicit rules about the world during inference. HITL-HGD achieves up to 300% improvement in generalization compared to HITL-D, reaching 100% success with as few as 7 demonstrations. Additionally, we demonstrate the ability to switch object instances under a single policy or task context while maintaining high success rates.

Together, these contributions demonstrate that shared control fundamentally enables practical assistive robotics by combining human strengths in task reasoning with autonomous capabilities in fine manipulation, reducing operator workload while preserving independence and enabling real-world deployment.

Preface

My work on this thesis was spurred by a comment from Dylan Miller, a fellow graduate student, about in painting human teleoperation data with generative diffusion policies. After working with diffusion policies and shared control architectures for some time I built up the experience to attempt the work in this thesis. Dr. Jagersand was an amazing source of information helping bridge the gap between autonomous methods and traditional tele-operation with his experience working with impaired individuals and foundational understanding of geometric constraints.

Parts of this work, authored by Riley Zilka, and supervised by Dr. Martin Jagersand, have been published and presented at IEEE/ICRA International Conference on Robotics and Automation and submitted to IEEE/ICRA International Conference on Robotics and Automation as:

- R. Zilka, S. Khlynovskiy, A. Wang, M. Jagersand. HITL-D: Human In The Loop Diffusion Assisted Shared Control. In International Conference on Robotics and Automation (ICRA), June 2026
- R. Zilka, and M. Jagersand. HITL-HGD: Highly Generalizable Human In The Loop Diffusion. Submitted to International Conference on Robotics and Automation (ICRA), July 2027

All Experiments were approved by University of Alberta Research Ethics and Management under permit Pro00054665

To my Mom and Dad

Thank you for giving me every opportunity to become the person I am today.

"Not all treasure is silver and gold, mate."

- Captain Jack Sparrow

Acknowledgements

Firstly, I want to give my biggest thanks to my supervisor, Dr. Martin Jagersand, who always encouraged problem solving in unique ways to solve seemingly complex problems. His deep curiosity and expertise in robotics, mathematics, and practical methods along with his near-religious commitment to *Scientific Computing: An Introductory Survey* by Michael T. Heath, ultimately guided me to the completion of this thesis.

I would also like to thank all of the friends I made along the way from the robotics lab to extra curricular activities, who always encouraged me, provided new perspectives and ideas and best of all being able to laugh with and make everything just a bit more enjoyable.

I am extremely appreciative of every opportunity I have been given to get to this point in my life and look forward to showing everyone who supported me that I am thankful.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Wheelchair-mounted robotic arms (WMRAs) hold significant promise in enhancing independence and quality of life for individuals with motor impairments [2–4]. Systems such as the MANUS and JACO robots extend the functional capabilities of powered wheelchairs by enabling users to perform activities of daily living (ADLs) such as grasping objects, opening doors, or manipulating devices [1]. In the United States alone, more than 9 million people live with a self-care disability and over 17 million people have independent living disabilities [5]. This represents a little under 5% of the US population who require daily assistance, and innovating solutions that enable self-care and autonomy would have a large impact. However, despite decades of research and technological advancement, WMRAs unfortunately remain underutilized in practice due to barriers such as cost and accessibility [6]. Another major barrier is the difficulty of control without training. Operating a high-degree-of-freedom robotic arm through small and limited interfaces, such as a standard 2-or-3-degree-of-freedom wheelchair joystick, imposes substantial mental workload on the user [7]. Therefore, developing new systems to reduce barriers of adoption like mental workload and effort during operation has the potential to improve the quality of life for millions of people.

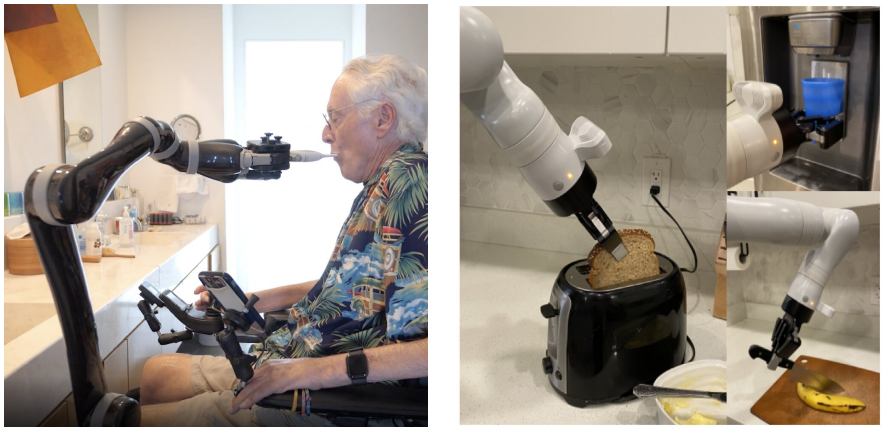


Figure 1.1: *Left:* The JACO robot arm (WMRA) being used by an elderly man in his home. *Right:* Example activities of daily living (ADLs) where robot arms may be used to help impaired individuals from [1].

Shared control is a popular field aiming to reduce workload, improve the ease of control, and minimize degrees of control input. It addresses tasks that could be performed by either a robot or a human independently, but accomplishes them through various combinations of autonomous and manual inputs from both parties. This approach reduces operator effort by decreasing the necessary dimensions of control and mental effort compared to traditional methods such as full Cartesian control [7–9].

The evolution of shared control has taken place over several decades. Early work in the 1990s established supervisory frameworks in teleoperation [11], which extended to arbitration strategies in the 2000s for assistive WMRAs that prioritized human input in the control and removed complete autonomy [12]. More recent approaches in the 2010s and later have matured into probabilistic and intent-aware models [**dragan2013policy**, 10], and increasingly leverage learning-based methods for personalization and high-dimensional control [13]. Learning-based methods are particularly valuable in their ability to adapt to user preferences and environmental changes while providing autonomous feedback to the system [14]. However, a well-documented challenge in shared control is that restricting user input, optimizing control mappings, or providing

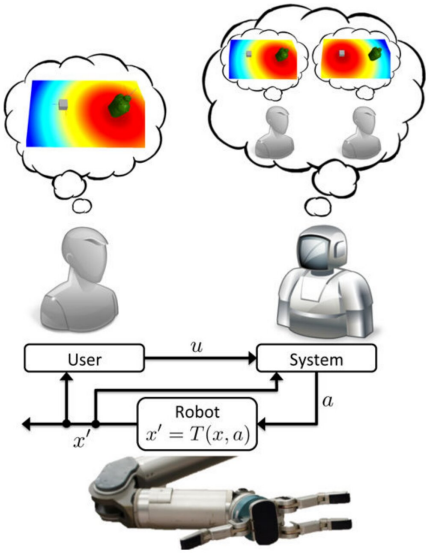


Figure 1.2: Javdani et al's shared control architecture illustrating the blending between human input and autonomous predictions. Here the system predicts possible goals and aligns with the user's behavior to output a corresponding action. [10]

autonomous control can sometimes result in a perceived loss of authority over the system [10]. Kim et al. state that non able-bodied individuals see WMRA's as a quintessential tool for enabling independent interaction but with too much assistance, WMRA's can be seen as just another agent to do work for them. This tension between autonomy and assistance is particularly important in assistive robotics, where maintaining user agency is as critical as improving task performance.

Another significant challenge limiting the wider adoption of learning-based shared control is the requirement for policies to generalize robustly across diverse environments, task variations, and workspace configurations. Recent work has argued that achieving such generalization requires developing large-scale standardized datasets containing millions of diverse tasks and demonstrations, paralleling successful approaches in vision, language, and robotics [15–18]. Yet learning robust sensorimotor policies remains fundamentally challenging due to high computational demands, the complexity of real-world physical

interactions, and the difficulty of capturing sufficient behavioral diversity in training data [18–20]. Consequently, the data requirements and computational barriers for developing generalizable shared control systems have become a significant obstacle to practical deployment in real-world assistive applications.

Recent approaches to improving policy generalization have explored several conditioning mechanisms. Point cloud-based conditioning, which leverages inherent spatial understanding from 3D data, has shown promise in improving generalization [21, 22]. However, these approaches typically require manual preprocessing steps such as hand-defined 3D region cropping to filter irrelevant points and precise camera extrinsic calibration—requirements that are impractical in real-world assistive settings where workspace configurations and camera placements vary. Alternative approaches condition policies on language instructions [23] or object-centric segmentation masks [24], yet these methods do not directly utilize geometric relationships between the end effector and task-relevant objects.

This thesis develops a novel split-control framework that integrates human teleoperation with diffusion policies, and investigates how geometry-aware conditioning enables generalization across diverse manipulation tasks with minimal demonstrations. The methods developed in this work address the tension between autonomy and assistance in shared control, with the goal of reducing barriers to adoption for modern WMRA systems.

1.2 Research Goals

To realize this vision of a practical shared control system, we ground our work in the perspective of individuals with motor impairments and prioritize concerns relevant to real-world deployment rather than theoretical convenience. This grounding reveals three critical design constraints that emerge directly from the barriers identified in the Background section: the system must be

intuitive and easy to learn (addressing the cognitive workload barrier of controlling high-degree-of-freedom arms through limited interfaces), it must generalize across diverse tasks and workspace configurations (addressing the data efficiency barrier that limits current learning-based approaches), and it must be robust to variability in user capabilities, physical setups, and environments (addressing the practical deployment barrier). These constraints motivate the following research objectives for our core thesis:

Stage 1: Human-in-the-Loop Diffusion Policies for Shared Control

How can we effectively integrate human teleoperation with diffusion-based autonomous assistance in a way that reduces cognitive burden without sacrificing user agency? Specifically:

- What is the relationship between shared autonomy and user perception of control and independence?
- How do we achieve technical performance improvements without sacrificing user experience and sense of agency?
- Can diffusion-assisted shared control maintain effectiveness across experiments grounded in everyday activities of daily living of varying complexity?

Stage 2: Generalization and Robustness of Shared Control Policies

How can we enable diffusion-based shared control policies to generalize beyond training data without requiring extensive manual scene engineering while minimizing data requirements? Specifically:

- What geometric relationships enable policies to reason about task constraints in a generalizable and interpretable manner?

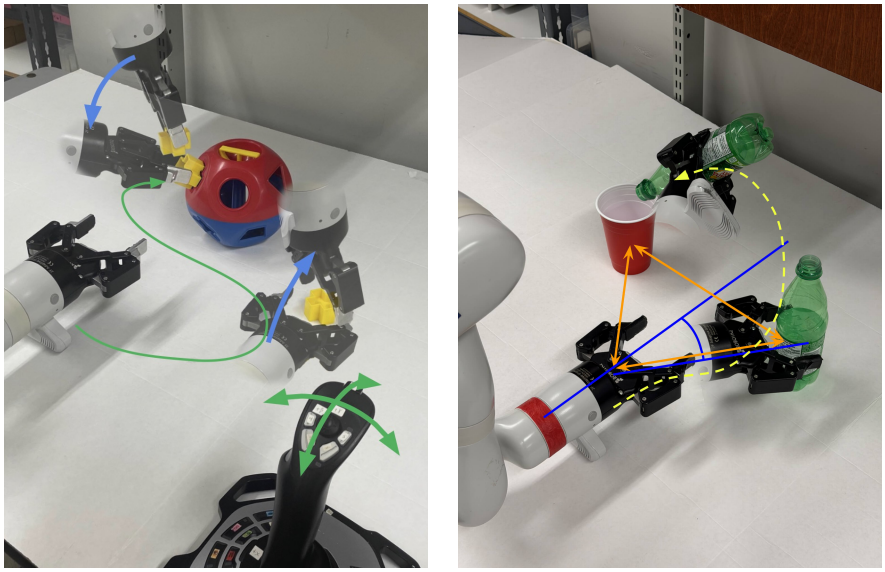


Figure 1.3: *Left:* HITL-D translates the end effector (green arrow) with 3 joystick-controlled axes, while our shared-control diffusion integration continuously adjusts orientation (blue) based on demonstrations. *Right:* HITL-HGD uses 3D point-to-point and alignment constraints to give the policy even more spatial context.

- Can geometric constraints additionally reduce the data burden required for capable autonomous assistance?
- To what extent do geometric constraints improve generalization beyond the training distribution to consistently complete tasks in unseen or shifted environmental conditions?
- What is the minimal data requirement for developing generalizable shared control systems, and how does this scale to new environments and tasks?

To address Stage 1 objectives, we introduce HITL-D (Human In The Loop Diffusion Assisted Shared Control) and for Stage 2 objectives we introduce its successor HITL-HGD (Human In The Loop Highly Generalizable Diffusion). This family of human-in-the-loop methods provides a foundation for building general, robust, and user-centered shared control systems for assistive teleoperation.

As seen in Figure 1.3, HITL-D was designed with the primary motivation of maintaining user authority at all times while providing meaningful autonomous assistance to reduce operator workload. It achieves this through a simple yet effective approach: leveraging expert human knowledge and task context while harnessing the capacity of diffusion policies to represent diverse robot behaviors in a flexible and generalizable manner. Diffusion policies have demonstrated significant promise in learning complex behavioral distributions from demonstrations [19, 25]. Following the control paradigm established in related work on orientation control in wheeled mobile manipulators [26–28], HITL-D employs a split-control approach for Cartesian motion. The user maintains direct control of translation (x, y, and z axes), while the diffusion policy governs the control of orientation (yaw, pitch, and roll). This decomposition allows human operators to focus on simple end effector translations while the policy handles fine orientation adjustments.

Building on HITL-D's foundation, HITL-HGD extends the framework with spatial constraint conditioning to improve generalization and enable robust performance across diverse workspace configurations. HITL-HGD is motivated by the goal of enabling individuals with motor impairments to access elegant, adaptable shared control systems. While maintaining the same split-control architecture as HITL-D, HITL-HGD enhances the underlying diffusion policy with explicit geometric constraints. We demonstrate that these spatial constraints provide the model with improved geometric scene understanding while remaining invariant to distribution shifts—a critical property for deployment in novel environments and with new object instances.

1.3 Thesis Organization

The rest of this thesis is as follows: Chapter 2 summarizes the literature review, highlighting fields in assistive robotics and machine learning, and

investigating the specific relationships to this thesis. Chapter 3 presents the first stage of this thesis: the combination of human input and diffusion policies in shared control. This chapter covers the methods and experiments for this stage, demonstrating its success and motivation for further exploration. Chapter 4 investigates generalization of the initial methodology, motivated by the need for practical application across a wide variety of ADLs. This chapter discusses the simple but effective changes made to the policy and control pipeline, along with the corresponding experiments and results. Chapter 5 discusses the implications of incorporating geometric information into diffusion-based robotic manipulation policies. It analyzes how structured spatial conditioning influences policy behavior, generalization, and design trade-offs, and what this suggests for the broader design space of shared control systems. Lastly, Chapter 6 draws the final conclusion and discusses future work. Our code and system is publicly available at: https://github.com/rileyzilka01/human-in-the-loop-diffusion.git

Chapter 2

Background Methods

This chapter provides an overview of the foundational methods and prior work that inform this thesis. We first review shared control approaches in assistive robotics in section 2.1, then examine imitation learning as a mechanism for learning from demonstrations in section 2.2. Finally, we discuss challenges of developing generalizable systems and explore conditioning mechanisms that enable adaptable policies in section 2.3. These three areas of literature form the basis for understanding the methods and evaluation in this thesis.

2.1 Shared Control for Assistive Robotics

In practice, current WMRAs, such as the Kinova Gen3 that we use for our research and experiments, have 6-7 degrees of freedom (DoFs). Individuals typically control these high-DoF WMRAs via a 2-3 DoF joystick on a wheelchair armrest, creating difficulty in efficient manipulation. Shared control—the combination of autonomous and user input—has been investigated as a means of reducing mental demand while improving usability. Explored since the 1990s [11], shared control has been utilized across automotive, medical, space, and assistive robotics [29]. Research in shared control has demonstrated promising results in making teleoperation easier and more efficient.

Despite decades of research, real-world adoption remains limited due to two

key barriers. First, users express concern about loss of agency and control authority when autonomous assistance is introduced [4, 10, 13, 30]. Second, many existing methods lack the expressiveness and generality required for real-world deployment across diverse tasks and environments [31–33]. These limitations motivate the need for shared control systems that can maintain user agency, reduce mental workload, and generalize effectively with minimal data requirements.

This section surveys the landscape of shared control approaches: supervisory methods (section 2.1.1), intent detection mechanisms (section 2.1.2), learning-based approaches (section 2.1.3), and control splitting strategies (section 2.1.4).

2.1.1 Supervisory Methods

Early research into supervisory frameworks was presented through Sheridan et al.'s [11] investigation into how humans could teleoperate vehicles on the moon's surface through communication delays and outages. This idea was conceived as a solution to the "stop and wait" control methods that were traditionally used. The traditional methods have an operator control movement for a brief moment, wait for the system to relay the data and repeat. Sheridan et al.'s solution was to remove the operator from the control loop and instead give the remote vehicle a list of instructions to complete a desired task while supervising the incoming data to interrupt the cycle if needed.

Supervisory control appears in everyday commercial systems such as airplanes. For example, pilots enter system parameters into flight computers enabling autopilot to perform autonomous course adjustments during the flight; the pilots retain the ability to take back control authority at any moment. Modern supervisory control examples in academia leverage natural language instructions to prompt autonomous systems to complete the task. [20, 34–36].

These methods demonstrate promising results in the ability for humans to instruct robots to autonomously grasp and maneuver in complex environments. However, beyond being an example of supervisory control, these methods have not been studied in shared control contexts involving humans and we are limited in our critique.

Another supervisory control approach is the one-click approach where a user indicates their target object by clicking on the screen to trigger autonomous retrieval [31, 32]. Dune et al. [31] use this method in assistive systems for severely handicapped individuals. Using the MANUS arm and epipolar based visual servoing techniques, they enable object grasping in complex environments. Together, an eye-in-hand camera and eye-to-hand camera estimate a depth map that is used to plan retrieval paths. However, despite demonstrating a successful method, it suffers from problems like obstacle avoidance during the path planning estimation and requires image templates of the desired objects for the tracking algorithm. Additionally, they fail to validate the system on end users. Similarly, Tsui et al. [32] implement the one-click approach while also testing it on end users of varying cognitive ability and report positive perceptions among users with medium-low cognition. However, they encountered the same technical limitations regarding template-based tracking and obstacle avoidance, highlighting persistent challenges in supervisory one-click approaches.

2.1.2 Intent Detection

Intent detection predicts user intent by integrating various components such as user input, external dynamics, and environmental information during task execution, enabling the system to autonomously provide situational assistance through various strategies [10, 12, 37].

An early version of intent detection was implemented in 2010 by Erden et al.

Figure 2.1: Erden et al. explored an intent aware system to assist humans with industrial vehicle disassembly and repair tasks. [38]

[38] for industrial robotic vehicle assembly and repair. Motivated by physically demanding tasks, they used intent detection to switch between fixed-position impedance control and interactive control by measuring human-applied momentum at the end effector. However, the approach has two significant limitations. First, it lacked a user study to validate how operators perceived this type of assistance. Second, the system is sensitive to load dynamics: it requires calibration for the mass in hand, and if the mass changes (e.g., if a door panel falls during disassembly), the momentum calculations become unreliable, degrading intent inference.

Beyond measuring interaction forces, Dragan et al. [39] formalized intent detection through control arbitration, as seen in figure 2.2. They call this framework policy blending, where a user and autonomous system jointly control the robot according to:

$$a_{robot} = (1 - \alpha)a_{human} + \alpha a_{robot} \tag{2.1}$$

Arbitration is the blending of a user action (a_{human}) and a robot action (a_{robot}) controlled by α (bounded between 0 and 1). Dragan et al. [39] propose that arbitration should be moderated by confidence in the prediction of user intent. They explore two prediction strategies: memory-based prediction,

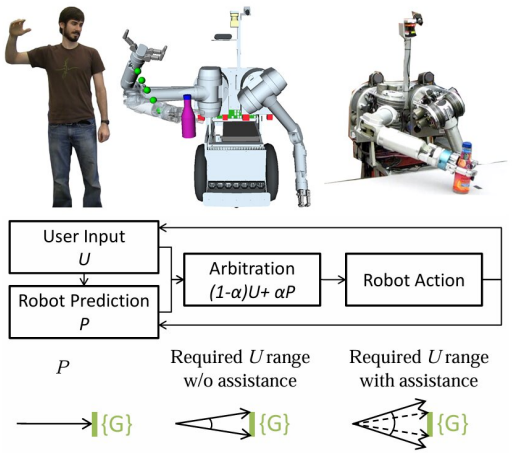


Figure 2.2: Dragan et al. formalized assistive teleoperation via arbitration of control determined by user intent. [39]

which uses all previous trajectory data to estimate the probability of future trajectories via cost functions, and amnesic prediction, which uses only current state to identify the closest goal. Their experiments revealed a key insight: users occasionally opted for less assistance even when it increased task completion time. They concluded that less blending is preferred when the task is easy for the human or when the robot lacks confidence. This finding directly motivates our shared control design, where we prioritize user agency by maintaining human authority over the robot's primary motion at all times.

Javdani et al. [10] build on the idea of modeling trajectory history for predicting the next robot action by formalizing shared control as a Partially Observable Markov Decision Process (POMDP). They assume users operate within goal specific distributions without external assistance, allowing them to create probability distributions over possible goals. The assumption allows them to satisfy the theoretical necessities of the POMDP to create probability distributions over the possible goals. When the policy is confident of a single goal, it will take over authority and autonomously completes the task; in cluttered scenarios with multiple possible goals, it blends control with the user. Their results showed that their system was able to provide input earlier,

allowing users to grasp objects faster and require fewer inputs to complete the task. However, users expressed dissatisfaction when the policy converged on strategies different from their own, reinforcing the need to maintain user authority.

Jain et al. supports user's concern for strategy misalignment by implementing a Bayesian filtering framework to infer over a number of possible goals [37]. By modeling human behavior as a Bayesian filtering in a Markov model, they are able to encode the users intent or goal changing over time. This is the conditional transition with the following probabilistic definition $P(g_t|g_{t-1})$ where g_t is the goal at time t and g_{t-1} is the goal at time $t-1$. By encoding dynamic goals and measuring uncertainty, their approach outperformed the results of both the memory based and amnesic methods from Javdani et al. [10].

Carlson et al. [12] implement adaptive intent detection allowing users to control how much assistance the system provides. They tested variable assistance in semi-autonomous wheelchairs for navigating through doorways and hallways. Users define waypoints and tasks while the system generates goal hypotheses based on user's intent; when difficult goals are presented, the system offers assistance that users can accept or decline. Their results demonstrate that their assistive system generates and follows better trajectories than non assisted trials. However, a lack of data on user preferences regarding control and independence makes it difficult to determine the paradigm's effectiveness for maintaining autonomy.

Despite showing more efficient task completion in most scenarios, intent detection typically falls short when measured on scales of user independence and authority. Javdani et al. states that users often feel like they lose authority of the robot, negatively affecting the perception of autonomy and independence. [10]. This observation is supported by Kim et al. [4], who suggests that users prefer to have control authority over easier control methods. They showcased

that only one thing significantly changed; users demonstrated less variability with the assistive mode.

2.1.3 Other Learning-Based Approaches

Beyond the intent detection paradigm, several learning-based shared control methods have explored alternative strategies for combining human and autonomous inputs. These approaches reveal important design considerations and limitations that shape the landscape of shared control systems.

Abi-Farraj et al. [33] propose a method that encodes demonstrated behavior as trajectory distributions, enabling the system to generalize to new situations while balancing autonomous assistance with user preferences. The learned distributions are continuously refined through collaborative execution between the human and robot. A key insight from their work is the use of distribution variance as an implicit measure of user preference: high variance indicates weak preference for a particular trajectory, while low variance suggests strong preference for a specific behavior. The system communicates this understanding to the operator through force feedback that indicates paths of least resistance. However, their approach relies on time-based Gaussian distributions over trajectories, which introduces a fundamental limitation when users deviate from the expected temporal progression. If an operator needs to correct a mistake mid-task, the distribution changes, potentially rendering the guidance unhelpful. Their evaluation demonstrated successful trajectory prediction and refinement through user interaction, though experiments were limited to object approach only and lacked subjective user ratings on autonomy and control perception.

Michel et al. [40] address the temporal dependency limitation by proposing a time-independent guidance approach using vector fields. Rather than conditioning on time, their method learns demonstration distributions and guides

the operator's movements within that distribution space. The guidance pushes the operator back toward it when outside and accelerates them along it when inside. By leveraging task dynamics such as forces and stiffness profiles, their method applies to tasks requiring physical interaction. Their evaluation on drawing tasks (S-shapes) and wiping motions (M-shapes) demonstrated lower mean errors, faster completion times, and reduced user workload compared to baseline methods. Importantly, their user study showed higher preference scores for the assistance and greater reported sense of control, suggesting that well designed autonomous assistance can enhance rather than diminish user agency.

Muelling et al. [13] investigate how adjustable levels of control authority can enable personalization in high-dimensional manipulation tasks. Their approach combines computer vision and arbitration strategies applied to brain-computer interface (BCI) control of a robotic manipulator. They model the scene using robot joint positions, gravity-compensated forces, and continuous object tracking via computer vision, defining capture envelopes around grasp approaches to favor accessible orientations. Following the arbitration framework formalized in prior work, they map neural activity from a BCI to motion commands. In experiments with tetraplegic participants controlling a Barrett WAM robot on door opening, pouring, and object manipulation tasks, their results demonstrated improvements in BCI-based control. However, the work also revealed a critical barrier: managing multiple control modes (separate modes for translation and rotation) imposed substantial mental load on users, significantly impacting task success rates. This finding suggests that reducing the number of distinct control modes is beneficial for user experience in assistive systems.

Another shared control method aims to reduce the degrees of freedom the human must explicitly control by learning state-conditioned mappings from

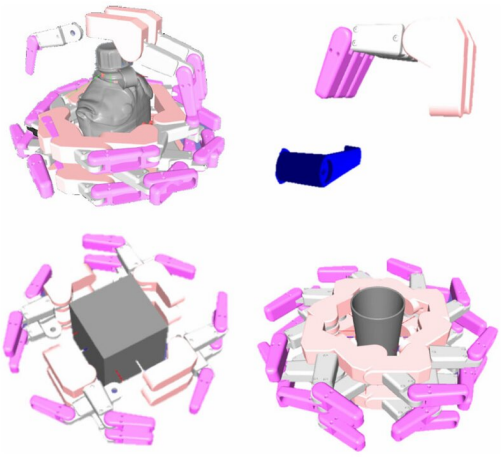


Figure 2.3: Muelling et al. [13] demonstrate stable grasp predictions for 3D objects using learned representations and arbitration-based shared control.

low-dimensional inputs to high-dimensional robot commands. Przystupa et al. [41] implement state-conditioned linear mappings learned from demonstrations, enabling control of a 7-DoF robot arm using a 2-DoF joystick. These locally linear maps compose to form non-linear control across the task space and allow for soft reversibility, permitting users to recover from unexpected robot configurations not seen during training. Their method outperforms autoencoder and PCA baselines on pick-and-place tasks and performs comparably to mode-switching approaches on pouring tasks. However, this approach exhibits reduced predictability outside the training distribution and applies reliably only to tasks similar to those in the training data [42], potentially creating confusion for users when the system behaves unexpectedly in novel situations.

While learned approaches to autonomous assistance show promise, grasp pose estimation represents a particularly constrained form of autonomous support. Mahler et al. [43] trained grasp pose prediction on synthetic datasets of millions of point clouds with analytically derived grasp metrics, demonstrating 99% precision on novel household objects including deformable items.

Ten Pas et al. [44] similarly developed grasp detection methods that localize robotic grasp configurations from RGBD images and point clouds without requiring CAD models, achieving grasp success rates between 75% and 95% on novel objects in light clutter. Although grasp pose estimation successfully predicts stable end-state orientations, these static goals often conflict with the user's immediate control objectives. Even when providing analytically stable poses, such rigid autonomous goals can diminish the user's sense of agency by overriding their intended manipulation strategy [30].

While autonomous solutions can be highly useful, a well-documented challenge in shared control is that restricting user input, optimizing control mappings, or providing autonomous control can sometimes result in a perceived loss of authority over the system [10]. This tension between autonomy and assistance is particularly important in assistive robotics, where maintaining user agency is as critical as improving task performance. The learning-based approaches reviewed in this section illuminate this tension: successful shared control requires careful design choices about when and how to provide autonomy, the dimensionality of control modes, and the robustness of learned behaviors beyond training distributions.

2.1.4 Architectural Paradigms: Splitting Control

We have reviewed methods that discuss adding assistance to the system in various ways but have not explicitly investigated literature for splitting control between both parties. This is different from control arbitration and blending because we do not allow the human and autonomous system to control the same motions, and instead delegate specific dimensions to both the human and the robot allowing each to control them individually. We investigate split control methods that are similar to our own, assigning the translational control to the user and the orientation and other dimensions to the system.

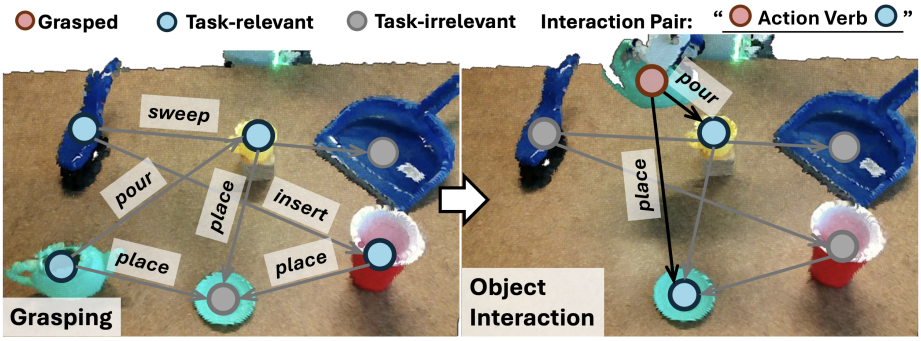


Figure 2.4: Fu et al. [28] showcase the benefits of a foundation model in pre-execution task planning.

A recent work called Task Aware Shared Control [28] explores the same orientation–translation split that we define in this thesis, but leverages foundation models and a grasp planner for zero-shot shared control. Fu et al. utilize the foundation model to construct an open-vocabulary interaction graph capturing possible object to object relationships and the corresponding interaction constraints. The results of their work demonstrate generalization across multiple manipulation tasks and object configurations without task-specific training. Additionally, the interaction graph technique influences success in multi-step tasks by providing long horizon interaction steps between objects, which is an important class of ADLs. Beyond achieving success across tasks, this split control architecture showcased reduced user input effort compared to existing baselines.

Jin et al. [26] developed an intent aware split control method for inspecting and viewing objects from different angles. In their work, the human operator directs initiatives through translational control to approach objects, the system then facilitates task execution through autonomous orientation control. The core idea is that the autonomous system will orient about an object as the user moves to see it from different angles. Using computer vision techniques, they track and focus on particular objects within the camera view. The orientation execution is determined through an intent aware policy. Intent is measured by

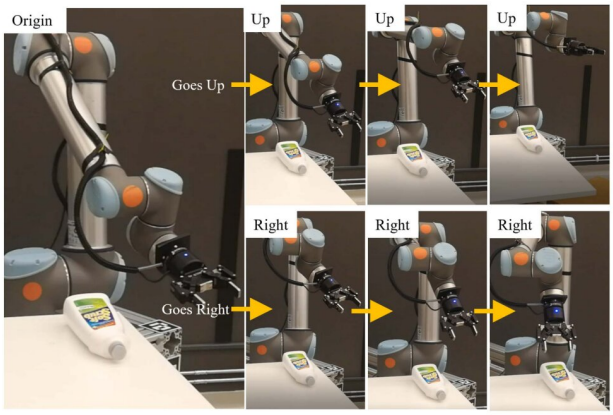


Figure 2.5: Sun et al. [27] demonstrate their split control key-pose guided assistive teleoperation system.

directly comparing calculated trajectories for seeking a particular object with the current user trajectory. While this work does implement a split control setup, its experiments were limited and show no indication of user preferences.

Sun et al. [27] raise concerns about bilateral teleoperation between differing devices; that is, when one device has different kinematics, number of joints or sizes than the other. This work establishes a set of keypoints for a task, for example, the pose of picking up a cereal box, the pose pouring it over the bowl, and the pose of putting it in a cupboard. To maneuver between those poses they utilize dynamic movement primitives (DMP) to generate trajectories between the poses. The operator moves the leader device translationally, while the robot performs orientation updates along the DMP trajectory. Their results showcase success in bilateral teleoperation of devices with different structure. However, this method does not preserve a natural control mapping in the task space; sideways movements on the leader device may not correspond to sideways movements on the follower device. This can potentially lead to increased mental workload for the operator.

2.2 Imitation Learning and Diffusion Policies

Behavior cloning (BC) is a foundational imitation learning approach where the objective is to learn observation-to-action mappings directly from demonstrations. However, BC struggles with multi-modal behaviors common in manipulation tasks, motivating more expressive alternatives. This section reviews BC and its limitations, then examines diffusion policies as a solution. We conclude by exploring why diffusion policies are particularly suited to shared control contexts through inpainting and other capabilities.

2.2.1 Behavior Cloning

Torabi et al. [45] learn policy behavior strictly from the state gathered during demonstrations, that is, their method is action agnostic. While most works are concerned with mimicking actions, in reality, humans do not know the specific action a demonstrator is taking. They demonstrate success in simulated tasks with their BC policy. Mandlekar et al. [46] were focused on creating replicable testing environments and datasets for behavior cloning and imitation learning research. They make an observation that history dependent BC has more success in task completion than batched reinforcement learning based imitation learning.

Codevilla et al. [47] explore the limitations of BC for autonomous driving. They demonstrate their adoption of BC on human driving datasets in the CARLA simulator. They identify limitations in BC. One of them is bias, especially in driving datasets. Most of the data consists of a few simple behaviors such as accelerating, stopping and more while there is a lot of different complex reactions to unseen events such as a pedestrian running into the road. This results in performance degradation because the diversity of the dataset does not grow in response to the amount of data. Another limitation is causal

confusion [48]; the experiments in the paper highlight causal confusion such as stopping at a red light, the probability of staying stopped is high, so this creates issues with restarting and acceleration.

Despite a widespread adoption of behavior cloning, there are well documented limitations [47]. Standard BC typically learns a unimodal mapping from observations to actions ($o \rightarrow a$). This presents issues with dataset bias and overfitting when learning actionable policies. In shared control settings where behavior is often multi-modal, a single user input could validly lead to multiple distinct manipulation strategies. Common BC models often average these modes and suffer from multi-valued function discontinuities [49], resulting in physically inconsistent or ineffective trajectories. Motivated by the limitations of mode averaging, we move away from standard BC models and move towards energy based models such as implicit behavior cloning [49] and diffusion for learning expressive policies.

2.2.2 Diffusion Models

Diffusion models are an increasingly popular class of generative policy in imitation learning. They work by treating policy generation as an iterative denoising process [50]. This approach is learned by systematically adding noise to sampled data and learning a function to remove it. By following the gradient of the data density, diffusion models capture multi-modal distributions. Unlike traditional behavior cloning, which predicts a mean trajectory from a unimodal distribution, diffusion policies sample from the joint distribution of entire trajectories, $p(\tau)$.

Chi et al. [19] demonstrate multi-modal expressiveness in their work called Diffusion Policy. They formulate a conditional de-noising process by conditioning action predictions on a history of data including robot positions and image data. They utilize an action-sequence prediction formulation to capture addi-

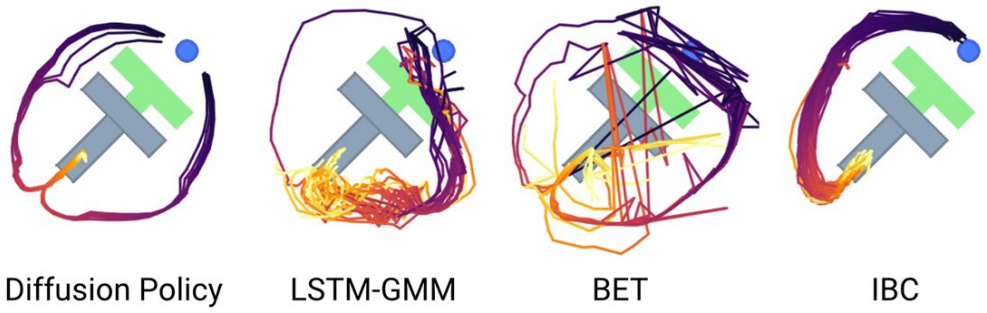


Figure 2.6: Chi et al. [19] visualize the multi-modal abilities of diffusion policies compared to other behavior cloning methods. The goal is for the blue dot to move to the other side of the "T" and push it into place.

tional robustness in temporal consistency and robustness to idle actions such as pouring liquids where you must wait for the pour to be finished. Their results showcase success in simulated and real-world experiments, demonstrating large improvements over traditional behavior cloning methods like IBC [49].

Their work yields several important findings for policy design. First, they demonstrate that diffusion policies can effectively capture multi-modal distributions, enabling more expressive action predictions than unimodal approaches. Second, they show that diffusion policies are more effective with position control than velocity control which contrasts with prior work [46, 49], where velocity control showed advantages. Chi et al. speculate that this difference arises because multi-modal distributions are more pronounced in position control. Additionally, position control avoids compounding errors that accumulate with velocity-based predictions. This finding has direct implications for our shared control design, where position control of translation aligns naturally with the autonomous control of orientation.

2.2.3 Why Diffusion Policies for Shared Control

Beyond their ability to capture multi-modal distributions, diffusion policies offer a second key advantage for our approach: inpainting. Inpainting is the

process of filling in or adding new content to a piece of data in the region specified. This capability directly aligns with our assistive split control design. In our approach, the human provides a Cartesian position (x, y, z), while the remaining 3 DoFs making up end effector orientation (θ, ψ, ϕ), are predicted by the policy. The policy's role is to inpaint an orientation that remains consistent with both the environmental observations and the user's position input.

Inpainting first demonstrated relevance in image generation, where researchers conditioned the inpainting process with the partially known image content [51, 52]. Notably, this generative process is not trained directly on the task of inpainting. Instead, the DDPM [50] is trained to generate an image that lies within a learned distribution, which naturally generates consistency during denoising. By specifying known values in a particular region, the model uses these constraints as distributional guidance during generation, a principle we leverage in this thesis.

To the best of our knowledge, there are few works investigate inpainting for robotic manipulation tasks. Hao et al. [53] explore inpainting in the context of a vision language model (VLM) to generate actionable 3D pose key frames from language prompts to guide policy generation. Since key frame poses can be inaccurate, they utilize an optimization strategy that balances key frame adherence with the learned motion priors. Their technique is marginally better on in-distribution tasks for experiments like grasp pose estimation and the Push-T task [19]. They additionally demonstrate significant improvements in generalization to out of distribution tasks, motivating our investigation of inpainting for autonomous policies.

Relating to the work from Hao et al. [53], we treat the user's translational input as a constraint while the policy inpaints an end effector orientation consistent with the task context. The user provides authoritative positional

guidance, and the policy's role is to reactively generate task-appropriate orientation that respects both this guidance and the learned behavioral distribution. For practical deployment in everyday scenarios, however, expressive policies must also generalize reliably across diverse environments and tasks.

2.3 Enabling Generalization

While diffusion policies and split control provide expressive learning and intuitive interfaces, enabling generalization across diverse environments and tasks remains a critical challenge. Generalization depends on key factors: structured conditioning data that captures task-relevant information, and invariance to distribution shifts such as object arrangement or coordinate frame changes. This section reviews conditioning mechanisms that encode rich environmental information and strategies for achieving robustness to such shifts.

2.3.1 Conditioning Mechanisms

It is one thing to increase the size of the model, how it learns the distribution or the capabilities of the hardware, but eventually we are limited by the data we feed the system. The ability to seamlessly condition our prediction on high dimensional data [54, 55] creates numerous possibilities for synthesizing expressive policies. This section examines two promising conditioning approaches: point cloud representations and geometric constraints.

2.3.1.1 Point Clouds and 3D Representations

A primary challenge in learning visuomotor policies is generalizing beyond the training data and task contexts while also maintaining data efficiency. Research into visual representations for conditioning manipulation policies has shown that 3D representations such as point clouds or voxels exhibit much stronger spatial generalization than 2D image or depth-based approaches [21–

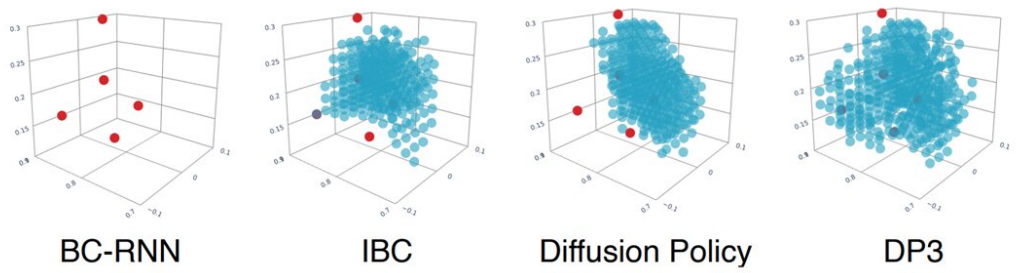


Figure 2.7: Ze et al. [21] showcase a motivating example for using point-clouds as a conditioning mechanism. The figure represents successful evaluation points (blue) from a set of 5 demonstrations points (orange). Their results surpass that of behavior cloning methods [49] and the image-based Diffusion Policy [19]

23, 56]. Point clouds naturally capture scene geometry, allowing architectures like PointNet++ [57] and Perceiver-based networks [23] to extract task-relevant structural features.

Shridhar et al. [23] developed PerAct, a transformer-based manipulation policy using a novel problem formulation. PerAct encodes language goals with RGB-D voxel observations using a Perceiver transformer to output discretized actions. The actions are decoded and passed into a motion planner to satisfy translation, orientation, gripper and collision constraints. By reformulating the problem with 3D structure and language conditioning, they enabled PerAct to perform at a state-of-the-art level surpassing policies which used unstructured image-to-action policies..

Ze et al. [21] built on the Diffusion Policy [19] framework by substituting image conditioning with encoded pointclouds conditioning along with the robot state. Their approach, DP3, takes a single view pointcloud, down samples it to 512 or 1024 points using farthest point sampling (FPS) and then encodes it to a 64 dimensional latent vector. This simple modification significantly improved performance. Across 72 tasks, DP3 demonstrated significantly higher success rates over baseline methods. DP3 additionally demonstrated that conditioning directly on the 3D data improves zero-shot generalization to novel

object poses. Their generalization extends to spatial (moving objects), appearance (flashing disco lights at the scene), instance (switching the objects) and view generalization (moving the camera). Beyond conditioning on pointclouds, they conducted ablation studies on their own method switching the pointcloud for voxel, RGB-D and image representations and concluded that pointclouds remained superior.

Li et al. [24] further showcased that when we can remove the noise of the scene via object centric segmentation, generalization abilities are increased even more. Their method Lan-o3dp follows a very similar setup to DP3, however, their differences arise in the point cloud conditioning. While DP3 requires view cropping to remove unwanted noise and points, Lan-o3dp uses language conditioned segmentation to get object masks and project these onto the pointcloud. This eliminates the need for manual viewpoint cropping while enabling broader generalization. Lan-o3dp demonstrates competitive results in all tasks compared with Diffusion Policy [19] and DP3 [21].

2.3.1.2 Geometric Constraints and Spatial Reasoning

Beyond capturing 3D structure of the environment, it is crucial to understand how objects interact with each other through geometric relationships such as alignment constraints and spatial dependencies. We draw inspiration from humans' innate ability to reason about object interactions [58]. Object interactions such as pouring water require geometric understandings: the bottle's pivot point, hand trajectory relative to that point, and the height difference between bottle and cup. Mitko et al. explain that humans have "intuitive dynamics", we judge trajectories intuitively without explicit calculation, we can draw conclusions based on intuition and spatial factors. Extending this capability to robotic policies motivates our investigation of geometric constraints.

Borghesan et al. [59] introduce formalized rules for incorporating geometric

constraints into robotic manipulation. Although they lack real-world evaluation on robots, they establish the idea of constraints such as distances, alignment angles, approach axes, normal axes and more. They formalize these constraints in the focus of expressions between entities. Additionally, they introduce a schematic for constraining behaviors such as moving by specifying a direction or rate of motion rather than the final position. Positioning behavior requires object centric geometry such as point to point differences or point to line projections. This work motivates our exploration of implementing such formalized constraints.

Sun et al. [60] incorporate geometric constraints using projective geometric algebra (PGA). PGA is grounded in Euclidean space using planes as primitives, it uses these planes to form a 4D basis representation for motion. They pass object poses as plane primitives along with robot forward kinematics into their diffusion policy, which results in a vectorized sequence of constraints that they can model motion with. While their method achieves high success rates, it requires 200 demonstrations for real-world tasks and depends on accurate pose estimation to extract plane primitives. Furthermore, their experiments use neatly structured rectangular objects, and it is unclear how the approach generalizes to everyday items like kitchen utensils and tools.

Li et al. [61] propose an alternative way to model spatial relationships, through their method ADPro, which constrains the diffusion denoising process to a spherical manifold. Rather than conditioning on spatial information, ADPro constrains the denoising process to respect underlying spatial rules, leveraging the fact that end effector pose naturally provides gradient direction. While ADPro improves backtracking efficiency and shows minor improvements over baselines, its geometrically constrained denoising does not explicitly encode spatial relationship understanding. This limitation motivates our goal: developing policies with the same "intuitive dynamics" that humans possess.

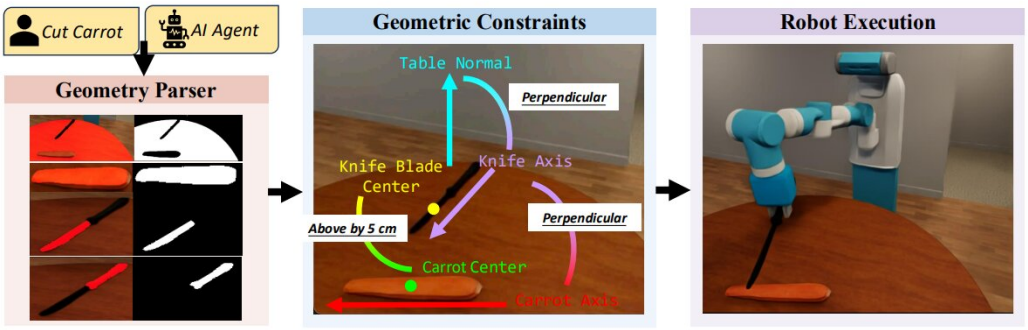


Figure 2.8: Tang et al. GeoManip method showcases an example of extracting various geometric constraints via VLMs [62]

Tang et al. [62] more directly incorporate implicit spatial knowledge in their method, GeoManip. Their method utilizes structured language commands to extract stage specific geometric constraints for the task context. A geometry parser identifies constraint primitives (planes, lines, edges, surface normals), which a vision language model then uses to generate cost functions for optimization. They achieve significant improvements in both simulation and real-world tasks like pick and place, pour, open, stir. However, it faces challenges in language-based problem specification, cost function generation, and constraint identification.

2.3.2 Invariance to Distribution Shifts

A key limitation of previous 3D learning methods like DP3 [21] is their dependence on hand-defined crop regions to eliminate noise and irrelevant data. These crops are sensitive to camera position changes meaning that new crops must be computed whenever the camera moves. This motivates two goals: eliminating manual cropping and achieving robustness to camera perturbations.

Li et al. [24] addressed cropping by using language-guided segmentation, which automatically isolates task-relevant objects and eliminates the need for manual preprocessing. This approach enables greater viewpoint generalization.

Alternatively, some methods transform point cloud coordinates into object-centric or robot-centric coordinate frames [23], allowing the policy to reason about spatial relationships independent of the camera coordinate system.

To achieve robust generalization, policies must be invariant to changes in scene configuration, object appearance, and camera viewpoint. Scene and object invariance can be achieved through language prompted segmentation, as demonstrated in prior work [23, 24]. Viewpoint invariance requires representing visual data in ways that are insensitive to camera motion. Translation invariance can be achieved by centering the point cloud on its centroid. Rotation invariance typically requires specialized network architectures such as Vector Neurons [63] or SE(3)-equivariant networks [64].

Alternatively, invariance can be achieved through data augmentation during training. Laskin et al. [65] demonstrated that augmenting training data with scene translations and arbitrary scalings improves generalization. This approach effectively expands the training distribution beyond the original demonstrations, enabling the policy to encounter and adapt to variations not explicitly captured in the data.

Chapter 3

Combining Human Input And Diffusion Policies

Now that we have built up a baseline of knowledge across all the necessary domains, we can begin building a system capable of the stage 1 research goals outlined in section 1.2. We introduce **Human In The Loop Diffusion (HITL-D)**, our first contribution towards enhancing assistive robotics with simplified shared control. This work does not address motion constraints, including intent detection or object avoidance. There is also no attempt to reduce the robots' degrees of freedom through a reduction mapping. Instead, motivated by our literature review on shared control, we introduce the usage of a split control paradigm that utilizes a learning-based method for the autonomous component. The user controls the robot's translational position at all times while a diffusion policy reactively inpaints a task-consistent orientation for the task. We chose diffusion policies because they have proven very beneficial in recent robotic manipulation research. They provide several advantages for shared control: naturally capturing multi-modal behavior distributions while seamlessly conditioning on high-dimensional observations such as point clouds and robot state. Our approach focuses on maximizing user independence while using shared autonomy to complete tasks faster and with reduced workload while being able to model human behavior appropriately.

3.1 Supplementary Knowledge

It is important to get a baseline understanding of how specific parts of our implementation work. We outline the general workings of a **Denoising Diffusion Probabilistic Model (DDPM)** [50]. This framework is the basis of Diffusion Policy [19] which is further developed in DP3 [21], one of our core influences.

Action Generation Starting from a sample drawn from a standard Gaussian distribution $\mathcal{N}(0, I)$, the model iteratively denoises this sample over K timesteps until convergence on a single action vector a^0. Let a^K denote the initial noisy sample. The denoising network ϵ_θ performs K sequential denoising steps, each guided by visual observations v (such as point clouds) and robot state s (joint positions, velocities, etc.), to produce a noise-free action according to:

$$a^{k-1} = \alpha_k(a^k - \gamma_k \epsilon_\theta(a^k, k, v, s)) + \sigma_k \mathcal{N}(0, I), \tag{3.1}$$

where $\mathcal{N}(0, I)$ is standard Gaussian noise added for stochasticity, and α_k, γ_k, and σ_k are coefficient functions of the timestep k defined by the noise scheduler. These coefficients control the magnitude of denoising at each step, early steps make large changes to the noisy sample, while later steps make fine refinements. The noise scheduler is designed such that the process smoothly transitions from pure noise to a valid action. This iterative procedure corresponds to the *reverse diffusion process* [50].

The key advantage of this formulation is that at each denoising step, the network conditions on both the current state of the sample and the observations, allowing it to generate actions that are simultaneously consistent with the learned behavior distribution and responsive to the current visual and state

context.

Training Objective To learn this denoising process, the network ϵ_θ is trained on a dataset $\mathcal{D}$ of demonstrated actions to predict noise that is artificially added to clean actions during the forward diffusion process. For each training sample, an action a_0 is pushed towards a standard gaussian distribution $\mathcal{N}(0, I)$ by adding an amount of noise corresponding to a random timestep k, creating a noisy version $\tilde{a}_k$. The training objective is to minimize the mean squared error between the true noise added and the noise predicted by the network:

$$\mathcal{L} = \text{MSE}\Big(\epsilon_k, \epsilon_\theta(\bar{\alpha}_k a^0 + \bar{\beta}_k \epsilon^k, k, v, s)\Big), \tag{3.2}$$

where $\bar{\alpha}_k$ and $\bar{\beta}_k$ are coefficients determined by the noise schedule that control how much clean signal versus noise is present in the noisy action. The network is trained to predict ϵ_k, the noise component, conditioned on the noisy action $\tilde{a}_k$, timestep t, visual observations v, and robot state s. By training on this objective across all timesteps, the network learns a conditional distribution over actions that respects both the structure of the demonstration data and the current observations and state.

Hyperparameter Tuning Hyperparameter tuning for diffusion policies is commonly adopted as follows;

- **Horizon** — the length of the action sequence the policy predicts in a single forward pass, determining the temporal window of planning. DP3 uses 4, Diffusion Policy uses 16-32.
- **Action Steps** — the number of action steps taken from the predicted sequence, allowing the policy to generate longer sequences but only ex-

ecute a subset, used for consistent generations. DP3 uses 3, Diffusion Policy uses Horizon/2 (8-16).

- **Inference Steps** — the number of denoising iterations performed during policy roll out, controlling the trade-off between inference speed and action quality; fewer steps are faster but may produce structurally inconsistent actions. Commonly chosen is 10.

- **Training Steps** — the number of discrete timesteps in the forward diffusion process during training, defining the noise schedule granularity; higher values provide finer noise schedules but increase computational cost. Commonly chosen is 100.

- **Noise schedule** — defines how much noise gets added at each training step (e.g., linear, or `squaredcos_cap_v2`), which is just a mathematical curve that controls noise progression. Linear often results in vanishing gradients while squared cosine presents more training stability via an adaptive schedule that avoids dumping lots of noise right away. Commonly chosen is `squaredcos_cap_v2`.

- **Learning rate** — the optimizer's step size controlling how aggressively the network weights are updated during training; typical values range from 10^{-5} to 10^{-3} depending on batch size and task complexity. Commonly chosen is 10^{-5}.

- **Learning rate scheduler** — how the learning rate changes during training (e.g., cosine annealing, linear decay, exponential decay) along with warmup steps; cosine annealing gradually decreases learning rate following a cosine curve, while warmup steps gradually increase the learning rate at the beginning of training to stabilize early learning. Commonly chosen is cosine with 500 warmup steps.

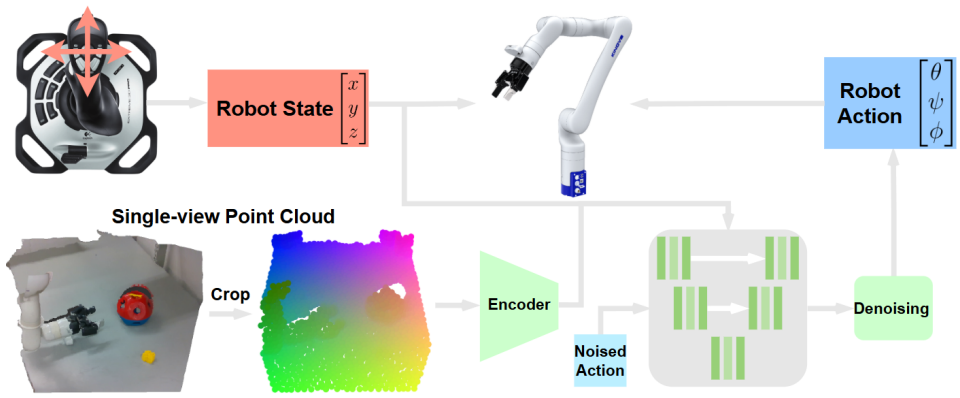


Figure 3.1: Overview of HITL-D methodology. During training, expert demonstrations are collected from a teleoperation interface, conditioning our robot action on a cropped and processed point cloud and robot state. During evaluation, a user will manipulate the robot by sending Cartesian position updates to the end effector. These updates run through our control loop along with point clouds to predict end effector orientations.

3.2 Methods

Human In The Loop Diffusion (HITL-D) is a purely reactive shared control approach where a diffusion policy conditions on sparsely sampled point clouds and end effector Cartesian positions to predict an end effector orientation as an assisting action. This combination of human input and autonomous output allows the user to maintain control while also benefiting from non-intrusive autonomous assistance.

Orientation was chosen as the action based on two considerations: safety and workload. The user always maintains primary control of the robot as the autonomous predictions cannot change the position of the end effector. Delegating orientation to the autonomous model simplifies system use, allowing the robot to perform fine, task-specific adjustments while the human focuses on coarse positional control. This is especially beneficial for individuals with physical impairments who struggle with fine manipulation. Additionally, by delegating only position control to the human operator, we eliminate the de-

manding mode switches present in other shared control methods, where operators must switch between different control modes [7–9].

In HITL-D, our diffusion policy maps observations $o \in O$ to an assisting action $a \in A$ while the user remains in primary control of the robot. Composing our system, there are 3 key parts: (a) **Autonomous Policy Training**, built upon a comparably small number of demonstrations and the perception-decision framework in [21], the action prediction is absolute end effector orientation in radians; (b) **Reactive Diffusion Policy**, adjusting certain hyperparameters ensures the policy reacts intuitively and preserves user control in real time; (c) **Human-Diffusion Interaction**, an assistive system is developed that reacts to the users motion using techniques that enable rapid, intuitive inference, allowing the system to react predictably to user input. Figure 3.1 illustrates the overall methodology and control loop.

3.2.1 Autonomous Policy Training

The autonomous component of HITL-D is a diffusion policy trained to map multi-modal observations into absolute end effector orientations. Each training step consists of three primary input components:

- **Perception**. HITL-D conditions on a colored point cloud to convey 3D spatial information, improving generalization and reducing demonstration requirements [21]. The point clouds are cropped with a hand picked bounding box based on the location of the camera relative to the scene. After cropping, farthest point sampling [66] is performed on the point cloud reducing the number of points down to 2048. This number balances inference speed and scene representation quality. The down sampled point clouds are subsequently encoded into a 128 dimension vector for simplified and faster training using the same 3 layer MLP in [21].

- **Robot State**. HITL-D also conditions on a robot state vector containing the 3 dimensional x-y-z position of the end effector relative to the Robot's base frame.

- **Robot Action**. HITL-D predicts a 3 dimensional end effector orientation in radians.

Our policy is designed as a conditional denoising diffusion model [19, 50] that we outline in section 3.1. We follow the same design choices and algorithm as Ze et al. [21] made for their DP3 model.

3.2.2 Reactive Diffusion Policy

In pursuit of a real-time, reactive control system, we adjusted the horizon to be 1 action prediction instead of the more commonly used, 4, 8 or 16 [19, 21]. This prioritizes reactivity over guidance to maintain user authority. We ignore any future predictions and only orient the end effector in response to the current Cartesian position and point cloud. Figure 3.2 demonstrates a motivating example for the change in horizon. When users have authority over the position of the robot, they can produce completely foreign movement sequences to the training data. This means that if we were to predict a horizon of 8, the policy expects the user to behave in similar sequences present in the training data, but if the human deviates, the orientation will not be consistent with what the user might expect. We also reduced the previous observation size for conditioning to just 1 due to the unpredictable nature of human control over the robot, making older observations less informative and perhaps destructive for accurate predictions. We recorded demonstrations with uniformly sampled sequence lengths of 256 to ensure the model observes fine-grained temporal transitions, allowing it to predict orientation corrections for a range of typical motions seen in the demonstrations.

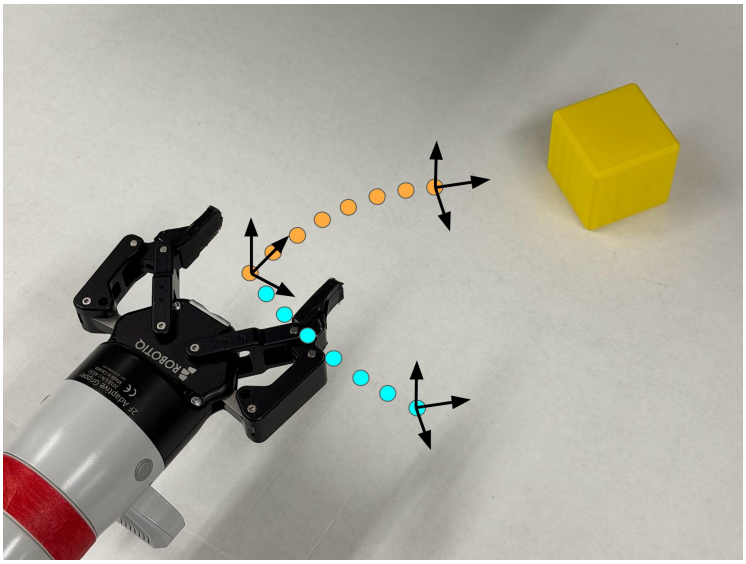


Figure 3.2: Expected path for policy (orange), user translation path (blue), end effector orientation (black axes). A justifying example for not using action chunking and instead using a horizon of 1; users can move in ways unexpected by the model resulting in an end state inconsistent with the user's path.

3.2.3 Human-Autonomy Interaction

The interaction between human control and orientation assistance via our diffusion policy is designed to be straightforward and intuitive. Since the policy predicts a desired absolute orientation, velocity commands are generated based on the difference between the prediction and current orientations. This provides rapid corrections when large discrepancies exist, while naturally slowing movements as finer adjustments are required. The gain for this simple proportional controller is 0.05. All motion is capped by a velocity limit to maintain safety of operation. This provides a smooth and responsive change for the users while not being quick enough to feel unnatural.

One significant positive result of our human-autonomy interactive control design is that the safety is guaranteed and determined by the human's control over the robot's Cartesian position; in response, the robot may rotate its joints at a slow speed but it is restricted from autonomously moving to unexpected positions in the workspace, preventing risk of damage to itself or surrounding

persons.

3.3 Experiments

The most important metric for evaluating the viability of our system is performance with real human users compared to other telemanipulation methods. These experiments were approved by University of Alberta Research Ethics and Managements (Pro00054665) and signed consent was obtained from the 12 non-disabled participants (ages 19-29, 9 male, 3 female). Experiments followed a single-blind format with control method order randomized across trials. Each participant used each control method for each of the three tasks.

- **Cartesian Control**: This method provides the user with 2 modes, one controls the 3 dimensions of Cartesian space (x, y, z) and the other mode controls end effector orientation (yaw, pitch, roll). Users switch between these modes by pressing a button on the joystick.

- **Point and Go Control**: This method [42] remaps the coordinate frames of the end effector creating a more intuitive "point and go" behavior, it was shown that Point and Go outperformed Cartesian control in most tasks while also being better than the learned SCL method [41]. Similar to Cartesian control, users can switch modes by pressing a button on the joystick.

- **HITL-D**: Our unique shared control method combining diffusion assisted orientation with human translational control.

Implementation details HITL-D adopts a convolutional network-based diffusion policy [19], using DDIM [67] as the noise scheduler. Each policy is trained on 3000 epochs with a demonstration timestep length of 256 for all of

(a) Unstack Cubes Task

(b) Screwdriver Task

(c) Shape Matching Task

Figure 3.3: *Left*: The Unstack Cubes task consists of 3 unaligned cubes and a plate, the goal is to grab each cube on their faces and place them onto the plate such that they are not stacked on the plate. *Middle*: The Screwdriver Task consists of a screwdriver and a gray hollow cylinder, the goal of the task is to grab the screwdriver and put it away inside the cup with the handle up. *Right*: The Shape Matching task consists of a cross shape that needs to be precisely aligned with the corresponding hole in the shape matching container.

our tasks, a horizon of 1 and a previous observation size of 1. Hyper-parameter explanation is given in section 3.2.2.

The three tasks used in our study (Fig. 3.3) were selected to capture different categories of ADLs [1] and differing levels of orientation demand that the average WMRA user may face during day to day activities. The unstacking task primarily tested translational, while the screwdriver and shape matching tasks required frequent orientation adjustments. This experiment design provides insight into whether the benefits of our shared control method extended to both orientation-intensive and orientation-light tasks.

HITL-D's diffusion policy was trained by first performing a single demonstration with an Xbox 360 controller, with Cartesian position mapped to the left stick and orientation mapped to the right stick with various functions spread across the buttons. Our control layout for both the user study joystick and Xbox controller are provided in figure 3.4. We converged on these control layouts through testing and consultation, the joystick layout minimizes operator effort and need to lift their fingers or hand off of the joystick during

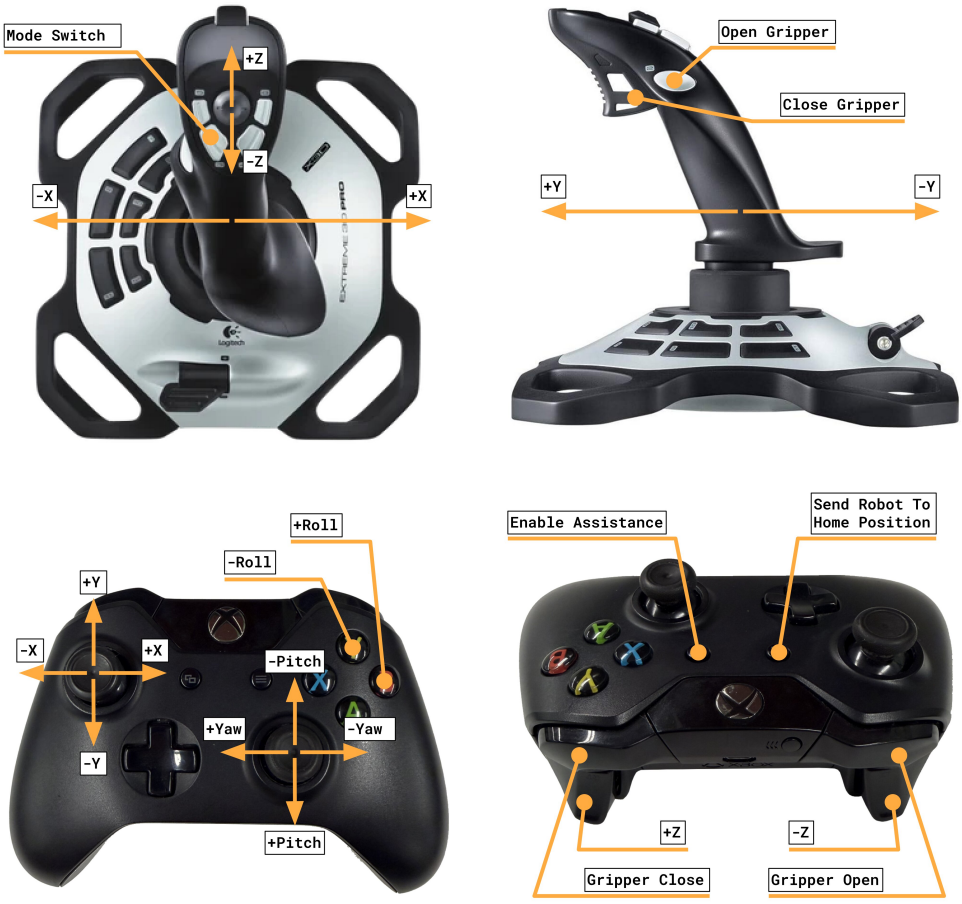


Figure 3.4: Control devices used in the study. *Top row*: Logitech Extreme 3D Pro Joystick interface used by study participants. *Bottom row*: Xbox 360 controller used by the author during demonstrations.

execution. We decided on the Xbox controller layout because it was most comfortable and easiest to use. Using the Xbox controller for demonstrations allowed simultaneous control over all degrees of freedom of our 7 DoF Kinova Gen3 robot arm while doing the demonstrations. The advantage of this control setup is that our demonstrations feel smoother and more natural as there is no need to pause and switch between the translation and orientation modes. All demonstrations were performed by an expert researcher to ensure high-quality trajectories. We evaluated and trained our policy on a GeForce RTX 3090Ti.

At this stage we are only concerned with evaluating the effectiveness of a

human and diffusion combination for manipulation, not the generalization of our policy. While broader research in robotic diffusion policies tests many demonstrations for generalization and robustness, we only test this framework with one demonstration per task which proved enough to make accurate action predictions. Consistent with popular research [19, 21], we expect that increasing the demonstration count would further enhance policy capability and adaptability to novel situations.

3.3.1 User Study Protocol and Metrics

At the start of each individual study, each participant was given a brief description of the controls and whether or not it had mode switching, but no context on what the controls did in terms of manipulating the robot. The users had 60 seconds to test and familiarize themselves with each system before attempting to complete 3 trials for each of the 3 tasks using each system. This was designed to counteract the effect of in-trial learning, where a user may improve on each trial and their subjective ratings may be impacted.

Beginning each trial, the robot was always in the same starting position and each scene setup was replicated for each user in the study. At the end of each task-method pair, the users were asked to fill out 2 surveys to measure subjective attitude towards the system and subjective workload ratings. The surveys are defined as:

1. **Likert Survey:** A 7 point survey measuring the following metrics;
    - *Confidence*: how confident are you in using the teleoperation mapping?
    - *Easier*: how easy was completing the task with this method?
    - *Independence*: how independently do you feel you can control the robot?

- *Intuitive*: how intuitive was the system?
- *Precision*: how precise were the commands in the system?
- *Predictable*: how predictable was the robot's behavior?
- *Undo*: how easy was it to undo mistakes with this system?

2. **NASA Task Load Index (TLX):** Hart et al. [68] developed a standardized system to measure subjective workload across different tasks. In this study, it is used to enable meaningful comparisons between methods.

Trial videos, task completion time, successes and failures, mode switches, completion times and the number of resets were collected for this study. However, mode switches were always 0 for our HITL-D method as there is only 1 mode. Resets were made available to the user in two forms; 1) required for the tasks when the cubes were knocked over unintentionally, the screwdriver was not dropped in the cylinder and became unreachable, or if the cross shape was dropped and became unreachable; 2) users could request a reset if the their control resulted in a robot configuration they could not understand or recover from.

Comparisons of quantitative metrics were achieved with two-way repeated-measures ANOVA. Binary success rates were evaluated with Cochran's Q test. For discrete data from the Likert assessments, Wilcoxon signed-rank tests were conducted. The threshold for significance was set at 0.05.

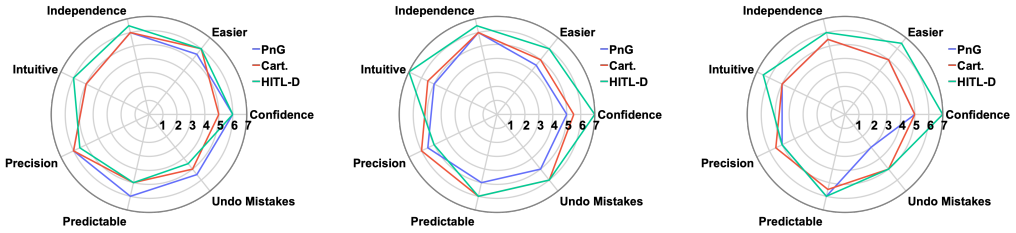
3.4 Results

Table 3.1 shows completion times, NASA-TLX workload scores, number of resets and success rates for each task-method pairing. Figure 3.5 displays the Likert scale metrics for both robot behavior and perceived competence. Figure 3.6 displays the bar chart for mean mode switches across methods on each task.

Table 3.1: Quantitative results[1] from the user study comparing control systems for three tasks.

Experiment \ Metric	Completion Time (s)			NASA-TLX Workload (/100)			Resets (#/Trial)			Success Rate (%)		
	HITL-D	PnG	Cart.	HITL-D	PnG	Cart.	HITL-D	PnG	Cart.	HITL-D	PnG	Cart.
Unstack	**95.5±7.1**	110.4±8.0	124.2±9.9	**32.5±4.8**	42.1±5.9	43.8±3.8	0.11±0.09	0.06±0.04	0.28±0.09	100.0	91.7	86.1
Screwdriver	**36.5±5.8**	88.1±9.5	72.9±9.3	**25.5±3.4**	53.6±5.2	42.1±6.4	0.22±0.11	0.22±0.09	0.17±0.08	100.0	94.4	97.2
Shape Match	**56.7±6.7**	126.1±7.0	96.9±7.7	**32.0±6.0**	57.6±6.1	52.5±7.2	0.33±0.10	0.44±0.09	0.25±0.10	97.2	77.8	97.2
Overall	**62.9±19.1**	108.2±21.1	98.0±17.4	**30.0±14.5**	51.1±16.4	46.1±13.8	0.22±0.25	0.24±0.16	0.23±0.24	99.1	88.0	93.5

[1] Results are reported with standard error. Bold = Shared significantly better than Point and Go; Underline = Shared significantly better than Cartesian.



Unstack Cubes Task | Screwdriver Task | Shape Matching Task

Figure 3.5: Radar plots for Likert scale survey measuring individual ratings for different parts of the system. (1-least favorable, 7-most favorable)

Overall we observed a significant decrease in completion times and perceived workloads across all tasks with HITL-D. Notably, standard deviations were lower for each metric with the exception of average resets per trial, indicating that performance improvements were consistent across participants. Figures 3.7 and 3.8 demonstrate the visual significance of HITL-D's improvements over Cartesian and Point and Go. We also noticed the number of mode switches with Point and Go to be greater than that of Cartesian control for the Screwdriver (15.47% more) and Shape Matching task (16.82%). This contradicts the results of Wang et al. [42]. We attribute this increase in mode switches to the type of task and movements required by the user, for example, if a user needed to move laterally without "pointing" they would have to switch between modes often to maintain the same end effector orientation. These types of motions were especially relevant in the screwdriver and shape matching task.

Unstack Task: HITL-D showed significant improvements with a time

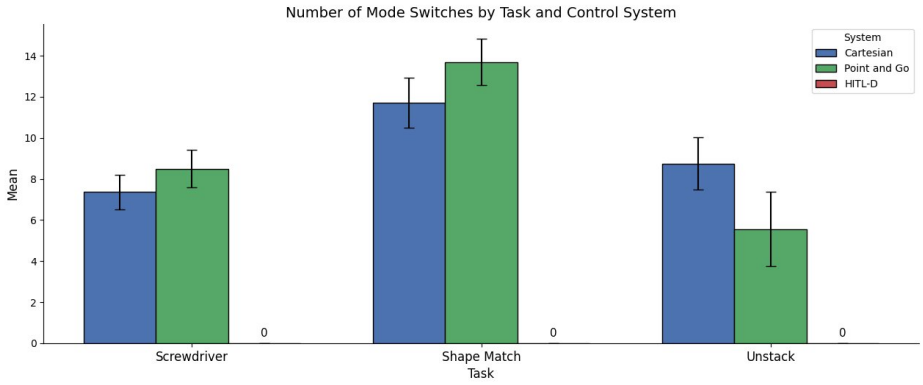


Figure 3.6: Graph displaying the average number of mode switches per method per task with standard deviation indicated by the error bars, notably HITL-D has 0 mode switches.

reduction of 13.5% and a NASA-TLX workload improvement of 22.8% in comparison to Point and Go. HITL-D was 23.1% faster and had a 25.8% lower workload than Cartesian. HITL-D maintained competitive in the Likert scores of independence and predictability and while not statistically significant, HITL-D showed trends of being more intuitive to the users. We also observed that HITL-D had a 100% success rate while Point and Go and Cartesian methods did not.

Screwdriver Task: We observe significant improvements of HITL-D over both PnG and Cartesian methods in completion times and NASA-TLX workload scores. HITL-D was nearly twice as fast as Cartesian, reducing average completion time by 49.9%, the workload score was 39.4% lower. HITL-D was more than twice as fast as PnG improving the average time by 58.6% and more than halving the workload score by with an improvement of 52.4%. HITL-D again maintained a success rate of 100% while Cartesian and PnG were close by at above 94% success rates in all trials. HITL-D also showed higher scores, and while the differences were not statistically significant, nearly all Likert survey metrics, except precision, indicate that users preferred HITL-D for this task.

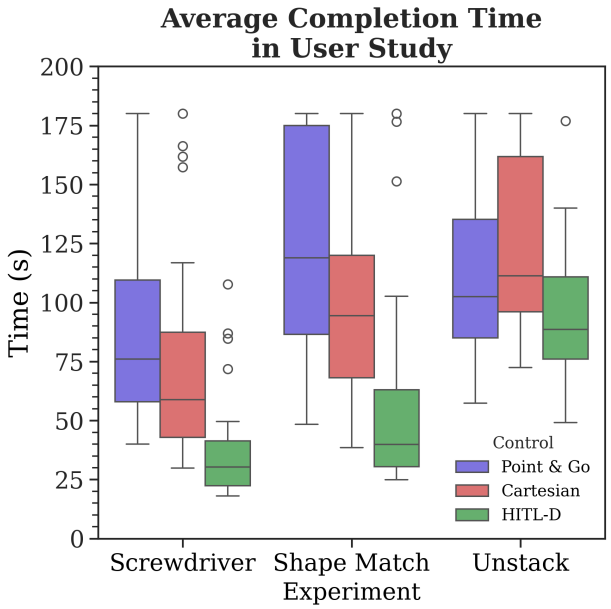


Figure 3.7: Box plot for completion times showcasing clearer differences in means, deviations and outliers

Shape Match Task: Again, we observe significant improvements of HITL-D over both PnG and Cartesian methods in completion times and workload scores. In comparison with Cartesian, HITL-D reduced average completion time by 41.2%, the workload score was 39.0% lower. HITL-D was more than twice as fast as PnG improving the average time by 55.0% and reducing the workload score by 44.4%. HITL-D had a very high success rate of 97.2% and while the PnG struggled to keep up at 77.8% success rate, Cartesian tied HITL-D at 97.2%. HITL-D improved in most metrics in the Likert scale survey except precision which still matched the other two methods.

3.5 Discussion

Unstack Task: The comparatively smaller performance gain of HITL-D over Cartesian and PnG methods in the unstacking task can be explained by the fact that this task does not require large or frequent orientation changes, de-

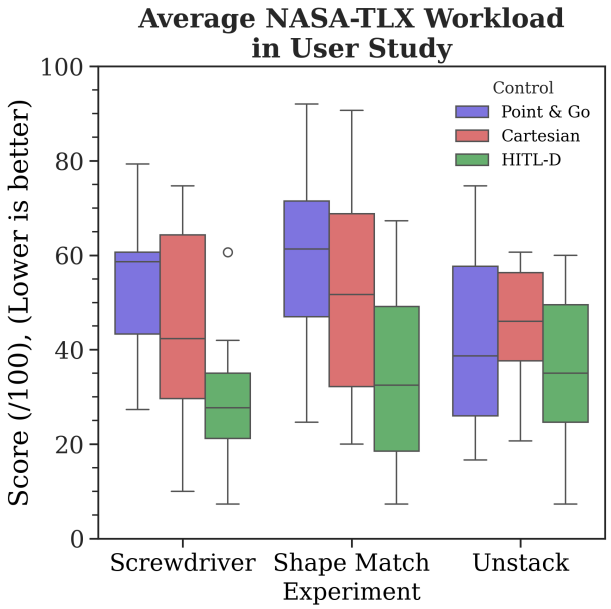


Figure 3.8: Box plot for NASA-TLX Workload ratings showcasing clearer differences in means, deviations and outliers

pending on the user's approach. We observed that some users bumped or pushed the middle cube to rotate it into a position they could grasp while others spent the time switching modes to get the correct orientation. The statistically significant improvement provided by HITL-D arises from its ability to automatically orient with respect to the cubes correctly, skipping the bumping step and reorienting toward the plate. We make the claim that our method is at least as effective as traditional teleoperation methods. Supporting this claim, the results from the unstack cubes task demonstrate that in situations where orientation adjustments aren't necessary, our method maintains or improves almost all metrics measured in the Likert survey of our method in comparison to the other two.

Screwdriver Task: Our evaluation indicates that users significantly benefited from the shared control framework in this task. Some users picked up the screwdriver completely perpendicular to the table and others picked it up at quite a shallow orientation before the autonomous component could reach its

prediction. Despite having only one demonstration, the model was still capable of assisting the user effectively with variations in user strategy leading to a 100% success rate. HITL-D's autonomous component for orientation proved significantly useful in user workload ratings as well as the empirical evidence in completion times. We noticed that the areas users spent the most time in while using Point and Go was when the forearm joints needed to flip due to the change of control frames, which led to visible frustration. Users who did not hit this case of joint flipping were not as frustrated and were able to complete the task with competitive times. The Cartesian control method also proved to be complex for users who did not use the mode switching for the Unstack task before this one. They were not as familiar with the orientation mode, causing them to perform worse. Users typically failed at this task or needed resets due to dropping the screwdriver outside of the tube. One interesting set of trials during the study led to a false negative for the HITL-D method, during the Cartesian experiment the user would lightly pick up the screwdriver so that it had enough friction to stay in the grippers but just low enough so that it would rotate within them to be completely vertical. When they were unable to utilize this technique with HITL-D they assumed that it was the control method not doing what they desired and continued attempting this behavior and needing to reset. Figure 3.7 demonstrates these times as outliers as they were the only inconsistent trials during the experiment. Subsequently this false negative resulted in the only outlier for the workload rating seen in Figure 3.8. Even with this outlier HITL-D remains significantly well suited for a task like this that necessitates large orientation changes.

Shape Match Task: This task posed the greatest difficulty for users, as achieving a successful insertion required the end effector orientation to be nearly exact in order to align with the shape-matching container. Users commonly struggled with the roll of the end effector for the Cartesian method

which caused the cross to be unaligned. Similar to the screwdriver task, users would pick up the cross at a very shallow angle, resulting in them not having enough pitch available in the workspace to orient the end effector so the cross had the right orientation. We also noticed a particular behavior only for the HITL-D method due to its shared control nature. Users inexperienced with the method and task often attempted to pick up the plus shape at a shallow angle. Upon trying to insert it they found that they were constrained by the end effectors ability to orient correctly with various pick up orientations and would have to reset. Evidence for this behavior is demonstrated in Figure 3.7 which visualizes the outliers in trial times. Once users recognized that the autonomous model required an additional period to achieve an optimal orientation when picking up the cross, they were able to complete the task more quickly, and no resets were needed after the first trial.

Further, we noticed significant bias in the shape matching task, users often expected to be able to complete the task in the way they desired, using different end effector orientations with respect to the table to pick up the cross. We speculate that in some cases this may have been due to the order at which the methods were trialed, if users did not use HITL-D first, they were free to pick up the cross any way they wished and attempt to insert it. Due to the demonstration for HITL-D being done by one of the authors, their internal bias towards task completion strategy was baked into the models predictions. This resulted in users often needing to learn how to interpret the policy on their first trial using HITL-D. Despite this need for interpretation and minor learning hurdle, most users still indicated that this method made them feel independent, and with a lower mental demand as well. Further investigation should go into user studies involving user created policies, where the user performs a trial as they would in a real world system and then use it to see if there is any difference in their understanding of task completion.

Across All Tasks: Overall, the observations demonstrate the efficacy of our method in time to completion, perceived workload, and on average higher user preferences for our method in the Likert metrics such as intuitiveness, ease of control and independent ability to move the robot. An interesting result we observed, although not statistically significant, was that Point and Go showed trends in higher task completion times and workload ratings than Cartesian in the Screwdriver and Shape Match tasks. We attribute this to the fact that Point and Go is a recent novel control method [42] and that while it may require more time to learn, it shows promising results for experienced users. Most importantly, we proved our claim that our method does not reduce human autonomy: users experienced increased autonomy and independence compared to that of Point and Go and Cartesian methods.

3.6 Stage 1 Concluding Statements

To conclude stage 1 of this work, we revisit our research goals introduced in section 1.2. Firstly, we investigated a novel way to combine human teleoperation input with the expressive power of diffusion policies. This investigation demonstrated that human in the loop diffusion significantly reduces users mental demand while preserving their agency during task execution. Second, we utilized various study protocols and surveys to measure this subjective workload and perception of the method. Grounding our method in everyday ADLs we were able to successfully evaluate its real world applicability. Lastly, we achieved the important goal of not sacrificing user control and authority for improved success rates. In fact, we saw a trend of users feeling like they had more independence and authority while also improving completion times and success rates. We conclude stage 1 as a success motivating our next chapter on stage 2.

Chapter 4

Using Geometric Relationships To Learn Implicit Relationships For Generalization

Following the successful completion of our stage 1 goals (section 1.2), we can begin our experimentation with the research goals outlined in stage 2 (section 1.2). To achieve this, we introduce **Highly Generalizable Human In The Loop Diffusion (HITL-HGD)**, our second contribution towards enhancing assistive robotics with simplified shared control. HITL-HGD builds on components of the literature we discussed in section 2.3.1 about conditioning mechanisms. Primarily, we investigate conditioning on spatial and alignment constraints to give the assistive policy implicit rules about the environment. Building on the evidence that 3D spatial information is invaluable for data efficiency and generalization, we contribute the idea that conditioning on 3D spatial constraints will further increase generalization and give the robot a geometric understanding of the objects and task. In this work, we utilize the same control setup as HITL-D. We aim to demonstrate that our new autonomous system with spatial constraints, when combined with intuitive and general human input, can result in a largely generalizable system that HITL-D lacked when presented with new scene and task setups.

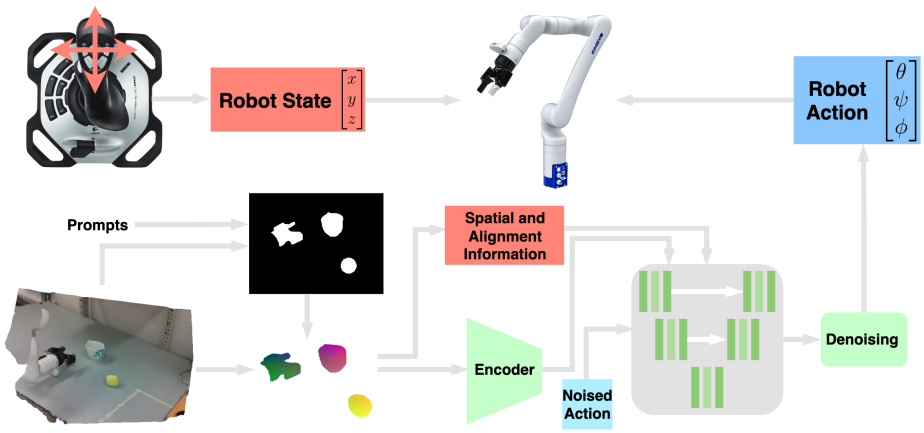


Figure 4.1: Overview of HITL-HGD methodology. A teleoperation interface is used to collect expert demonstrations. Our robot action is conditioned on the segmented point cloud, along with the spatial and alignment constraints. During operation, the human user will manipulate the end effector translationally while the diffusion policy predicts the continuous 6D SO(3) representation and converts it into Euler angle differences for the end effector.

4.1 Methods

HITL-HGD maintains the reactive shared control framework of HITL-D while replacing the cropped point cloud and end-effector position conditioning with a segmented point cloud, centroid-difference constraints, and alignment constraints derived from SAM-3 segmentation, giving the policy an explicit geometric understanding of the scene in camera coordinates.

In HITL-HGD there are again 3 key components: (a) **Point Cloud Segmentation and Geometric Constraints**, using recent advancements from Meta's SAM-3 [69] we segment our point clouds and gather our constraints for training; (b) **Diffusion Policy Training**, efficiently using a small number of demonstrations and the same perception-decision framework in [21]. The action prediction is the continuous 6D SO(3) representation [70]; (c) **Reactive Human-Diffusion Interaction**, the same assistive system formulation used in HITL-D that reacts to users motion using techniques that provide rapid and

intuitive inference. Figure 4.1 illustrates the overall methodology and control loop.

4.1.1 Point Cloud Segmentation and Spatial Constraints

We use Meta's SAM-3 [69] to automatically segment task-relevant objects, eliminating the need for hand-defined workspace crops required by prior work [21, 22]. To do this, we built a live implementation of the SAM3 code base, allowing us to pass a list of prompts to the SAM3 segmentation model which then returns masks for each prompt individually. For objects that SAM-3 cannot segment from a single prompt (e.g., the end effector), we apply sub-prompting: a broader prompt such as "robot arm" first isolates the arm region, and a second prompt applied only within that mask reliably extracts the gripper. In cluttered scenes where a prompt matches multiple objects, we retain only the mask whose 3D centroid is closest to the end effector.

Once we have segmented each of our prompts, we can extract centroid differences, defined as the vectors between the centroids of our objects, along with alignment constraints. First, we use each 2D image mask to mask our 3D point cloud; this mask allows us to calculate a centroid of the remaining points. We repeat this for each prompt. We only take the centroid difference between our end effector and each object of interest in our task context. We also perform another sub-segmentation of a region on the robot forearm just behind the end effector, this region is not involved in the centroid differences but is needed for the alignment constraints.

The alignment error measures how well the end effector is oriented relative to the target object. We compute two unit vectors in camera coordinates. First, the unit vector from the end effector ($\mathbf{p}_{forearm-region}$) to the forearm region ($\mathbf{p}_{ee}$), representing the end effector's current orientation frame. Second, the unit vector from the end effector ($\mathbf{p}_{ee}$) to the object ($\mathbf{p}_{obj}$), representing

the desired approach direction. The alignment error is then:

$$e_{align} = \left\| \frac{\mathbf{p}_{forearm-region} - \mathbf{p}_{ee}}{\|\mathbf{p}_{forearm-region} - \mathbf{p}_{ee}\|} - \frac{\mathbf{p}_{ee} - \mathbf{p}_{obj}}{\|\mathbf{p}_{ee} - \mathbf{p}_{obj}\|} \right\|_2 \tag{4.1}$$

This error is bounded between 0 (perfectly aligned) and $\sqrt{2}$ (completely misaligned). Each position ($\mathbf{p}_{ee}$, $\mathbf{p}_{forearm-region}$, $\mathbf{p}_{obj}$) is a 3D centroid obtained from segmentation. We compute this constraint for each prompted object to encourage the policy to maintain proper orientation throughout task execution.

After gathering our constraints, we merge the point cloud and perform voxel sub sampling to quickly reduce the number of points. We then apply farthest point sampling [66] to obtain a uniformly distributed set of exactly 1024 points. We reduced this number from 2048 points in HITL-D because the segmentation filters out irrelevant information, resulting in a very high fidelity scene representation with 1024 points. The down sampled point clouds are subsequently encoded into a 64-dimensional vector for simplified and faster training using the same 3 layer MLP in [21]. This new point cloud processing method is an improvement over our stage 1 methodology. Performing a large trim using voxel sub sampling and then doing FPS allows us to process pointclouds up to 15x faster on the same hardware and as a benefit improves inference speed.

4.1.2 Diffusion Policy Training

We utilize the same diffusion policy formulation specified in section 3.1 to map multi-modal observations into the continuous 6D SO(3) representation space. We adopt the 6D representation of SO(3) following Zhou et al. [70], which provides continuous, differentiable rotation representations suitable for neural networks.

6D SO(3) Representation Since orientation is a fundamental component of manipulation tasks, we represent end-effector rotation using the 6D rep-

resentation of the special orthogonal group SO(3). Rather than using Euler angles, which suffer from singularities and discontinuities [70], the 6D representation leverages two orthonormal 3D vectors that form the first two columns of the rotation matrix. Given a rotation matrix $R \in \mathrm{SO}(3)$, we extract the representation as:

$$\mathbf{r}_{6\mathrm{D}} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \end{bmatrix} \in \mathbb{R}^6, \tag{4.2}$$

where $\mathbf{r}_1$ and $\mathbf{r}_2$ are the first two column vectors of R. The third column can be recovered via the cross product: $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$. To ensure orthonormality during training, we apply Gram-Schmidt orthogonalization to the predicted 6D vectors, projecting them back onto SO(3). This representation provides several advantages: it is continuous and differentiable, avoiding discontinuities present in Euler angle representations, and it naturally integrates with neural network training objectives.

Training Each training step of the diffusion policy consists of three primary input components:

- **Spatial Information**. HITL-HGD conditions on the segmented and colorless point cloud described above to convey 3D spatial information.
- **State Vector**. We also condition on the 3D spatial constraints. This vector is a length of $3 * (n - 1)$ for the centroid differences and the alignment errors which is length $(n - 2)$. The number of prompts is represented by n. We have 1 prompt that is for the forearm region, 1 prompt for end effector and the remaining $n - 2$ prompts for our task context.
- **Action Vector**. HITL-HGD predicts the continuous 6D SO(3) representation.

Additionally, another change we make for our phase 2 goals is that we no longer condition on end effector position as this is a non invariant data point. End effector position is grounded in the robots base frame, every position is unique for the end effector. If we move the camera or scene layout, the end effector position needed to complete this task changes. Thus, we eliminate absolute end effector position from our conditioning to reduce the dependency on non invariant data.

4.1.3 Reactive Human-Autonomy Interaction

HITL-HGD inherits the same reactive interaction design from HITL-D: an action horizon of 1 and observation window of 1 prioritize real-time responsiveness over trajectory guidance, and the proportional orientation controller with velocity capping. This design ensures the new spatial constraint conditioning integrates without altering the user experience validated in HITL-D. We further define how we utilize the 6D SO(3) action prediction in our control loop.

Angular Velocity Command Generation The predicted 6D orientation is converted to a rotation matrix R_{target} via Gram-Schmidt orthogonalization. Given the robot's current orientation as a quaternion $\mathbf{q}_{\text{current}}$ (converted to rotation matrix R_{current}), we compute the rotational error in the tool frame:

$$R_{\text{error}} = R_{\text{current}}^{-1} \cdot R_{\text{target}}, \tag{4.3}$$

which represents the relative rotation needed to reach the target orientation. We convert this error rotation to an angular velocity command using the exponential map:

$$\boldsymbol{\omega}_{\text{error}} = R_{\text{error}}.\text{as\_rotvec}() \in \mathbb{R}^3, \tag{4.4}$$

where as_rotvec() returns the rotation vector (axis-angle representation) in radians. A proportional controller with gain K_p generates the desired angular velocity:

$$\boldsymbol{\omega}_{\text{cmd}} = K_p \cdot \boldsymbol{\omega}_{\text{error}}, \tag{4.5}$$

with a dead band threshold τ to prevent oscillations at small errors:

$$\omega_i = \begin{cases} K_p \cdot \omega_{\text{error},i} & \text{if } |\omega_{\text{error},i}| > \tau \\ 0 & \text{otherwise} \end{cases}, \tag{4.6}$$

where $i \in \{x, y, z\}$ and $\tau = 0.5°$. The final command is scaled and converted to degrees for hardware compatibility.

4.2 Experiments

In this section, we experimentally show the dramatic increase in generalization over HITL-D using the same control method. All demonstrations for our experiments were gathered on the Kinova Gen3 robot using an Xbox 360 controller. All data collection and evaluation was done on the same controller setup seen in figure 3.4. Data collection and evaluation were performed by one of the authors.

Implementation details We trained each of our policies over 3000 epochs using an RTX 3090Ti, we evaluated all models and policies including the segmentation model simultaneously on the same machine, maintaining inference speeds close to 3 Hz for the system.

Since HITL-HGD uses the identical control mechanism as HITL-D, we hold the control scheme fixed and focus our evaluation exclusively on generalization — isolating the contribution of spatial constraint conditioning.

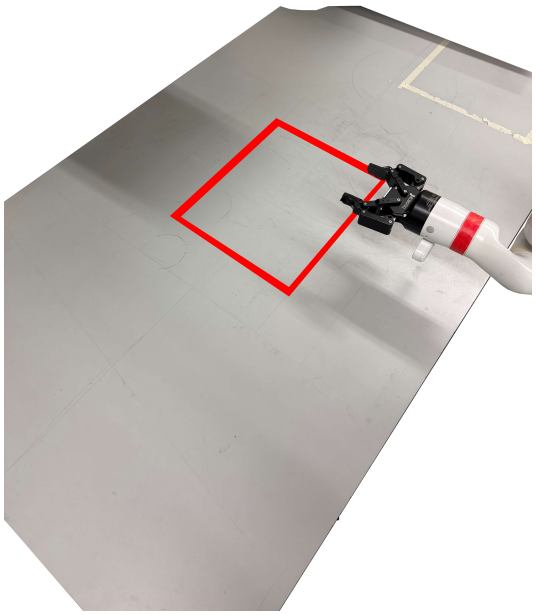


Figure 4.2: The red bounds indicate the border between in distribution (ID) and out of distribution (OOD) trials

(a) Pouring Task

(b) Scooping Beans Task

(c) Recycling Task

Figure 4.3: *Left*: The Pouring task's goal is to grab the bottle, orient towards the red cup and become fully inverted without spilling anything. *Middle*: The Scooping Beans task has a goal of scooping the beans from the green pot to the white bowl. A successful trial includes getting a moderately sized scoop of beans and getting them to the white bowl and orienting to dump them without spilling any. We modify a spoon for this task for the grippers ability to grab it. *Right*: The Recycling task's goal is to get all 3 cans (oriented differently) to the cardboard box to the side of the workspace. A successful trial includes getting all 3 cans into the cardboard box.

4.2.1 Generalization As A Function Of Demonstrations

In this experiment, our aim is to measure how the number of demonstrations affects our workspace generalization. We train 4 policies for each task for each method, each policy has 1, 3, 7 and 10 demonstrations respectively. We then measure the success rate across 10 in distribution (ID) trials, which are within the training distribution workspace. We also measure the success rate, as defined in Figure 4.3, over 10 trials in the out-of-distribution (OOD) workspace. See Figure 4.2 for the workspace bounds. Each policy will be measured on the same 10 trial layouts to ensure consistency and coherence in evaluation metrics. We conduct this experiment on 3 tasks that capture increasingly complex ADLs [1]; they can be seen in Figure 4.3.

4.2.2 Object Invariance

We further show that our SAM3 prompted image segmentation integration allows object generalization across the same policy for a single "skillset" such as pick and place. We train a single policy with point cloud centering and simulated rotations over 10 demonstrations for a pick and place task using a tennis ball and a mug. We then evaluate the performance of this policy under different pick and place tasks, conducting 10 trials over the training distribution workspace. The different pick and place tasks can be seen in Figure 4.4. We evaluate both HITL-HGD and HITL-D on all four object configurations to isolate the contribution of SAM-3 semantic prompting.

4.3 Results

4.3.1 Generalization As A Function Of Demonstrations

In Figure 4.5, we see that the ID trials for HITL-HGD follow a logarithmic trend indicating that we don't gain further generalization beyond 7-10 demon-

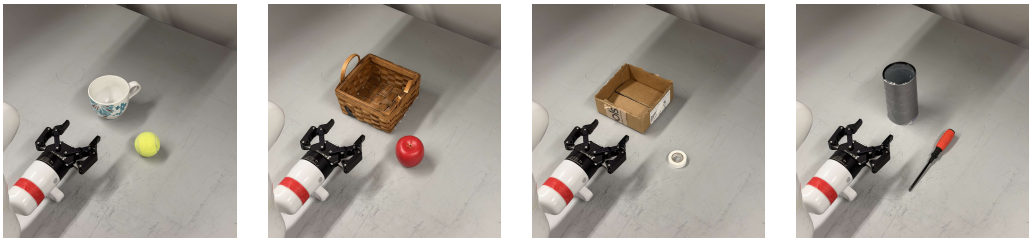


Figure 4.4: We present four variations of the same pick and place policy. We train the policy on the tennis ball and mug and evaluate on the same objects along with 3 other sets. Each requiring differing completion modalities.

strations. Meanwhile, HITL-D improves slightly but does not showcase any learning trends. During experiments with HITL-D, orientation predictions appeared to lack structure, successful trials did not exhibit consistent, task-aligned orientation strategies. In contrast, HITL-HGD's predictions showed clear geometric alignment with object configurations, suggesting that explicit spatial conditioning enables the policy to learn meaningful task structure. We notice the same behavior for HITL-D's OOD trials.

For the pouring task, HITL-D was completely unable to complete the task without spilling water as a result of trying to predict end effector orientations via the discontinuous Euler representation. We notice a significant improvement by switching to the continuous 6D SO(3) representation [70], which avoids the discontinuities that cause Euler-based predictions to collapse for full-inversion orientations. We can see that at 7 demonstrations, HITL-HGD has a 90% success rate for ID trials and 100% success rate for OOD trials. Strangely, HITL-HGD's OOD trials with the 10 demonstration policy only had a success rate of 20%. We notice that upon moving outside the red lines as seen in Figure 4.2, the end effector would roll 90 degrees preventing pickup of the bottle. We attribute this to the last 3 demonstrations; although not exhibiting any distinct changes in execution, they may have caused the learned distribution to be far different when moving into unknown space and converg-

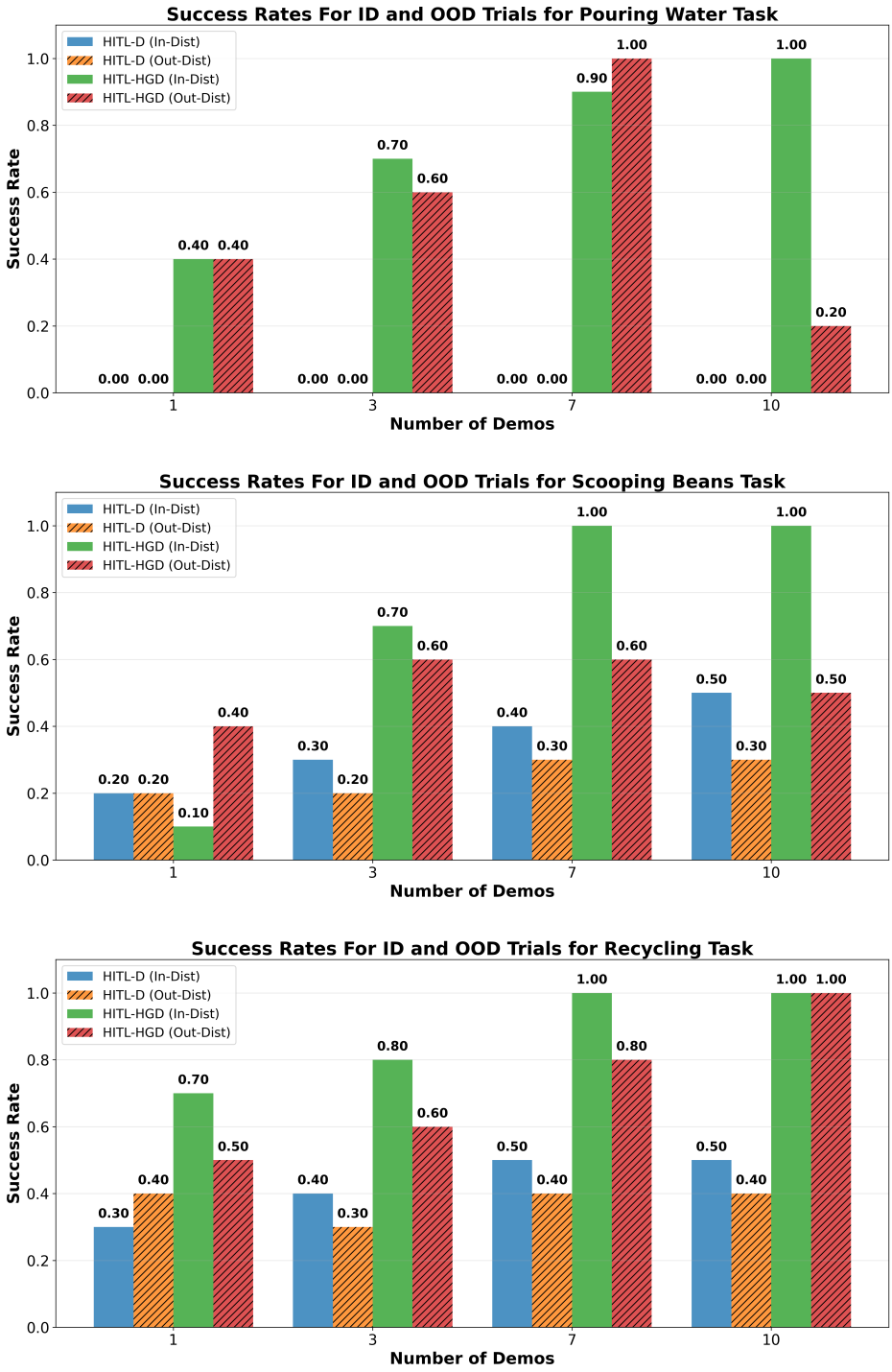


Figure 4.5: Histogram showcasing success rates for the pouring fluids task on ID and OOD trials with HITL-D (blue and orange respectively) and HITL-HGD methods (green and red respectively).

ing on one behavior. This sensitivity to additional demonstrations suggests that demonstration filtering or providing structured demonstrations could be a promising direction for future work.

For the bean scooping task, we see ID success rates top off at 100% at 7 demonstrations while the OOD trials only reach a 60% success rate between all demonstration amounts. We notice a linear trend in the success rate of HITL-D reaching a plateau at 50% success at 10 demonstrations while there was no significant change in the OOD trials success rates with the best being 30% success at 7 demonstrations.

For the recycling task, the limiting factor was being able to pick up the cans laying down at a variety of angles. HITL-HGD was more reliably able to orient with respect to the can it was closest to which we attribute to conditioning on the alignment error. This is demonstrated in the success rates where at minimum HITL-HGD was able to get 70% success on ID trials and 50% success on OOD trials on only 1 demonstration. Meanwhile, HITL-D's best performance at 10 demonstrations was 50% ID success and 40% OOD success.

4.3.2 Object Invariance

The results in Figure 4.6 demonstrate the robustness of HITL-HGD to unseen distributions. We see very similar performance between both methods for the tennis ball + mug and apple + basket, the "pick" items share a very similar shape, while the mug and basket are distinct to represent the "place". However, we see HITL-D drop off steeply for the thin tape roll + cardboard box. The strength of HITL-HGD's alignment information in the policy conditioning allows the end effector to rotate down when the tape roll is below the end effector. This is further enforced in the screwdriver + cylinder example, not only did the end effector have to rotate down for the handle but it also had to

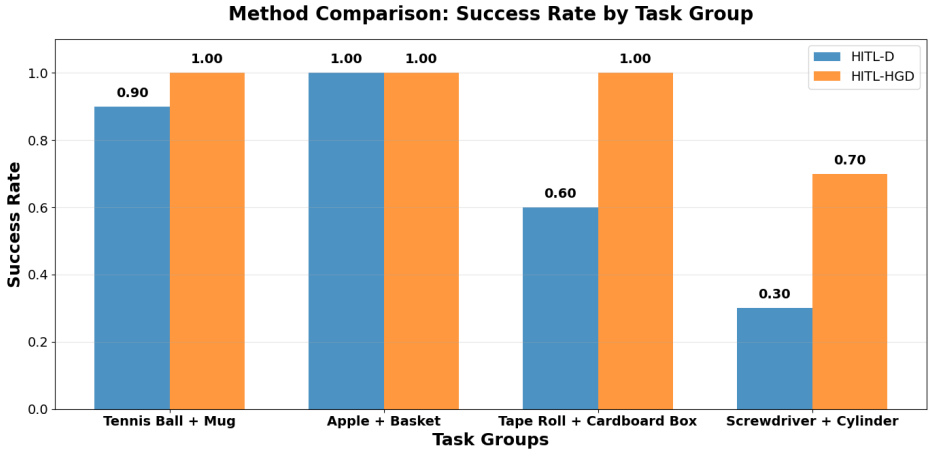


Figure 4.6: Histogram for the results of the object variance experiment

orient the screwdriver vertically to be able to drop it in the cylinder. We saw that HITL-D was unable to understand these geometric relationships, limiting its ability to adapt in different circumstances, while HITL-HGD was able to succeed with 70% success on a pick and place task with completely different constraints.

4.4 Stage 2 Concluding Statements

Concluding stage 2 of this thesis, we review the goals stated in section 1.2. Importantly, we enabled our diffusion-based shared control method to generalize across diverse workspace configurations without any hand engineering involved. Further, we investigated the contribution of spatial conditioning mechanisms showcasing great success in their ability to give the model implicit task constraints. We showcased that this contribution also extends far beyond the training distribution with consistency in the ability to complete the task in OOD trials. Our experiments highlight key success in only requiring 7 demonstrations for a highly generalizable model, reducing the data burden for deployment. Finally, stage 2 was a large improvement towards enhancing assistive robotics via imitation learning.

Chapter 5

Discussion: Generalization and Structure in Learned Shared Control

5.1 Shared Control as a Solution to Temporal Task Constraints

A fundamental challenge in autonomous manipulation is handling tasks with temporal decision-making, knowing when to stop an action and transition to the next. Autonomous diffusion policies with action chunking [19, 21] attempt to achieve temporal consistency by predicting action sequences ahead of time. However, this approach creates a critical limitation: once the predicted action sequence completes, the policy may continue executing a trajectory before the actual task is finished. Consider pouring water from a bottle into a cup, the policy might predict a full pouring motion, but the human operator knows contextually when the cup is full and when to stop.

By integrating human input directly into the control loop, we resolve this tension. The human operator naturally possesses the task context required for temporal decision-making. In our framework, as long as the human maintains the end effector over the target (the cup), the robot continues its learned pouring motion. The moment the human moves away, the diffusion policy reorients the bottle back to an upright position. This reactive coupling ensures

fluid, goal-oriented assistance that adapts to real-time task progress rather than fixed action sequences.

Crucially, our approach is immune to distribution shifts caused by idle or unexpected states. Because the human can move out of any learned distribution at any time, the robot must continuously adapt to the human's real-time guidance. Our experiments demonstrate that the diffusion process successfully refines motion sequences to remain consistent with both the learned demonstration distribution and the operator's immediate input, eliminating the need for the policy to "know" when tasks end—the human provides that knowledge.

5.2 Data Efficiency Through Shared Control and Geometric Conditioning

A significant practical advantage of our approach is its dramatically reduced data requirement compared to fully autonomous baselines. DP3 [21] and Diffusion Policy [19] require 40 to 284 demonstrations for real-world tasks, whereas our method achieves high generalization with as few as 7 demonstrations and shows success on as little as 1 demonstration. While not directly comparable due to differing control assumptions, this efficiency stems from two complementary factors.

First, shared control inherently reduces the demonstration burden. The human operator covers the spatial manifold of a task through natural movement, broadening the distribution over which the autonomous policy must succeed. Rather than requiring demonstrations that exhaustively sample all possible states, the policy only needs to learn how to assist the human effectively within the human's natural action space.

Second, our geometric conditioning approach through centroid differences and rotation simulation provides the policy with implicit understanding of task structure. Unlike methods that condition on absolute positions in the

camera frame, we represent conditioning information as relative object distances. This fundamentally remodels the manipulation problem: the policy learns that certain actions should occur at specific geometric relationships—for instance, orienting the end effector relative to an object's orientation when the end effector is within a few centimeters. This geometric representation makes the policy invariant to object location within the point cloud, a property that classical approaches must learn implicitly from diverse demonstrations.

The benefits of this approach extend further through our training strategy. Rotation simulation provides the policy with implicit knowledge of how scenes behave under camera rotations, enabling generalization to camera viewpoint changes never seen in the demonstrations. Point cloud centering similarly allows the policy to generalize across scene arrangements: if objects maintain the same relative distances as in a training demonstration but are positioned differently in the workspace, the policy treats them as distributionally equivalent. The diffusion policy learns the underlying task manifold in states we do not explicitly demonstrate, and by encoding implicit geometric rules, its ability to generalize across that manifold grows substantially.

5.3 Accessibility and Usability for Non-Expert Operators

Beyond technical performance, a critical measure of success for assistive systems is their usability by the intended population. Our user study demonstrates that operators without prior robotics experience or training can efficiently complete complex manipulation tasks while experiencing significantly reduced mental workload compared to teleoperation. This finding directly addresses a major barrier to WMRA adoption: the requirement for extensive operator training. The result is a system that can be deployed "out of the gate", a step forward in making advanced robotic assistance accessible to in-

dividuals with motor impairments who would benefit most from such systems.

Chapter 6

Conclusion & Future Work

6.1 Conclusion

This work presents a novel framework combining diffusion policies with human input for assistive robot teleoperation. Through two complementary stages, we demonstrate that our shared control method-where a human operator provides Cartesian velocity commands and a diffusion policy reactively determines orientation—significantly reduces operator mental workload while maintaining user independence and improving task efficiency. Our initial framework (Stage 1: HITL-D) achieved statistically significant improvements in task completion times and perceived workloads through a user study with operators lacking prior robotics experience, establishing the benefits of shared control over purely human or fully autonomous approaches.

Building on this foundation, we enhanced the generalization capability of the policy (Stage 2: HITL-HGD) by conditioning on explicit spatial relationships—centroid differences and alignment constraints—rather than relying solely on implicit learning from diverse demonstrations. This geometric conditioning approach yielded up to 300% improvement in generalization performance, enabling high success rates with as few as 7 demonstrations and the ability to generalize across object instances and workspace configurations without explicit camera calibration. By centering point clouds in the cam-

era frame and applying rotation simulation during training, we provide the policy with implicit geometric understanding that extends beyond its training distribution.

These results underscore a fundamental insight: shared control leverages human strengths in task-level decision-making and spatial reasoning while the diffusion policy handles fine-grained manipulation, naturally covering the spatial manifold of tasks and reducing the demonstration burden. By integrating human input directly into the control loop, we transform manipulation from a fully autonomous problem into a collaborative one, reducing operator workload without sacrificing the expressive capacity of diffusion-based policies. This approach represents a significant step toward accessible, practical assistive systems that can be deployed without extensive operator training.

6.2 Future Work And Limitations

6.2.1 Limitations

Despite the generalization improvements demonstrated, our approach has several limitations that present opportunities for future work. First, the policy observes only the current frame with no temporal history, which can cause instability when a single noisy observation produces an outlier action. Second, incorporating SAM-3 segmentation reduces inference speed: In our setup the full system runs at approximately 3 Hz on an RTX 3090ti. Further optimization of the segmentation pipeline would be necessary for time-critical tasks. Third, depth images from the RealSense D435i are relatively noisy, introducing centroid estimation errors near reflective or geometrically thin objects and occasionally degrading alignment constraint quality.

6.2.2 Future Directions

Our work opens several avenues for future research. A critical near-term direction is conducting user studies with impaired individuals to validate usability and workload reduction for our intended demographic. Concurrently, evaluating a wider variety of activities of daily living would provide insight into the broader applicability of this method across diverse manipulation tasks.

On the other hand, a promising line of future work concerns the type of data required for training. Because our control loop is reactive to the human with a horizon of 1, we do not require temporally consistent demonstrations. This enables a novel data collection strategy we might call a "state space sweep", where we provide training data at specific states and behaviors rather than full temporal sequences. For example, in a pouring task, we would only need to demonstrate pouring from either side of the cup, data with the bottle in hand, and data with an empty hand. This approach significantly reduces the total number of timesteps needed while ensuring the distribution covers task-critical states, eliminating redundant behaviors in common situations.

Bibliography

[1] L. Petrich, J. Jin, M. Dehghan, and M. Jagersand, "A quantitative analysis of activities of daily living: Insights into improving functional independence with assistive robotics," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 6999–7006. DOI: 10.1109/ICRA46639.2022.9811960.

[2] B. Driessen, H. Evers, and J. v Woerden, "Manus—a wheelchair-mounted rehabilitation robot," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of engineering in medicine*, vol. 215, no. 3, pp. 285–290, 2001.

[3] V. Maheu, P. S. Archambault, J. Frappier, and F. Routhier, "Evaluation of the jaco robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities," in *2011 IEEE international conference on rehabilitation robotics*, IEEE, 2011, pp. 1–5.

[4] D.-J. Kim et al., "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 42, no. 1, pp. 2–14, 2011.

[5] *Centers for disease control and prevention*. [Online]. Available: https://www.cdc.gov/dhds/about/index.html.

[6] J. Bourassa, J. Faieta, J. Bouffard, and F. Routhier, "Wheelchair-mounted robotic arms: A survey of occupational therapists' practices and perspectives," *Disability and Rehabilitation: Assistive Technology*, vol. 18, no. 8, pp. 1421–1430, 2023.

[7] L. V. Herlant, R. M. Holladay, and S. S. Srinivasa, "Assistive teleoperation of robot arms via automatic time-optimal mode switching," in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2016, pp. 35–42.

[8] H. A. Tijsma, F. Liefhebber, and J. L. Herder, "Evaluation of new user interface features for the manus robot arm," in *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, IEEE, 2005, pp. 258–263.

[9] A. Campeau-Lecours, U. Côté-Allard, D.-S. Vu, F. Routhier, B. Gosselin, and C. Gosselin, "Intuitive adaptive orientation control for enhanced human–robot interaction," *IEEE Transactions on robotics*, vol. 35, no. 2, pp. 509–520, 2018.

[10] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *Robotics science and systems: online proceedings*, vol. 2015, pp. 10–15 607, 2015.

[11] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. MIT press, 1992.

[12] T. Carlson and Y. Demiris, "Human-wheelchair collaboration through prediction of intention and adaptive assistance," in *2008 IEEE international conference on robotics and automation*, IEEE, 2008, pp. 3926–3931.

[13] K. Muelling et al., "Autonomy infused teleoperation with application to brain computer interface controlled manipulation," *Autonomous Robots*, vol. 41, no. 6, pp. 1401–1422, 2017.

[14] A. Broad, T. Murphey, and B. Argall, "Learning models for shared control of human-machine systems with unknown dynamics," *arXiv preprint arXiv:1808.08268*, 2018.

[15] A. O'Neill et al., "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 6892–6903.

[16] B. Ai et al., "Towards embodiment scaling laws in robot locomotion," *arXiv preprint arXiv:2505.05753*, 2025.

[17] A. Kirillov et al., "Segment anything," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4015–4026.

[18] A. Kumar et al., "Pre-training for robots: Offline rl enables learning new tasks from a handful of trials," *arXiv preprint arXiv:2210.05178*, 2022.

[19] C. Chi et al., "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02 783 649 241 273 668, 2023.

[20] K. Black et al., "Pi0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[21] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[22] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024.

[23] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*, PMLR, 2023, pp. 785–799.

[24] H. Li, Q. Feng, Z. Zheng, J. Feng, and A. Knoll, "Generalizable robotic manipulation: Object-centric diffusion policy with language guidance," in *Workshop on Embodiment-Aware Robot Learning*, 2024.

[25] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," *arXiv preprint arXiv:2208.06193*, 2022.

[26] Z. Jin and P. R. Pagilla, "Collaborative operation of robotic manipulators with human intent prediction and shared control," in *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, 2020, pp. 1–6. DOI: 10.1109/ICHMS49158.2020.9209565.

[27] D. Sun and Q. Liao, "Asymmetric bilateral telerobotic system with shared autonomy control," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 1863–1876, 2021. DOI: 10.1109/TCST.2020.3018426.

[28] Z. Fu, P. Song, Y. Hu, and R. Detry, "Tasc: Task-aware shared control for teleoperated manipulation," *arXiv preprint arXiv:2509.10416*, 2025.

[29] D. A. Abbink et al., "A topology of shared control systems—finding common ground in diversity," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 509–525, 2018. DOI: 10.1109/THMS.2018.2791570.

[30] M. A. Collier, R. Narayan, and H. Admoni, "The sense of agency in assistive robotics using shared autonomy," in *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2025, pp. 880–888.

[31] C. Dune, C. Leroux, and E. Marchand, "Intuitive human interaction with an arm robot for severely handicapped people-a one click approach," in *2007 IEEE 10th International Conference on Rehabilitation Robotics*, IEEE, 2007, pp. 582–589.

[32] K. M. Tsui, D.-J. Kim, A. Behal, D. Kontak, and H. A. Yanco, ""i want that": Human-in-the-loop control of a wheelchair-mounted robotic arm," *Applied Bionics and Biomechanics*, vol. 8, no. 1, pp. 127–147, 2011.

[33] F. Abi-Farraj, T. Osa, N. P. J. Peters, G. Neumann, and P. R. Giordano, "A learning-based shared control architecture for interactive task execution," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 329–335.

[34] S. Tellex et al., "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 25, 2011, pp. 1507–1514.

[35] G. C. Kang, J. Kim, K. Shim, J. Lee, and B. Zhang, "Clip-rt: Learning language-conditioned robotic policies from natural language supervision,(2024)," *arXiv preprint arXiv:2411.00508*, 2024.

[36] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 139–13 150, 2020.

[37] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 9, no. 1, pp. 1–23, 2019.

[38] M. S. Erden and T. Tomiyama, "Human-intent detection and physically interactive control of a robot without force sensors," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 370–382, 2010.

[39] A. D. Dragan and S. S. Srinivasa, "Formalizing assistive teleoperation.," in *Robotics: Science and Systems*, Sydney, Australia, vol. 8, 2012, pp. 73–80.

[40] Y. Michel, Z. Li, and D. Lee, "A learning-based shared control approach for contact tasks," *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 8002–8009, 2023.

[41] M. Przystupa et al., "Learning state conditioned linear mappings for low-dimensional control of robotic manipulators," *arXiv preprint arXiv:2410.21441*, 2024.

[42] A. Wang, C. Jiang, M. Przystupa, J. Valentine, and M. Jagersand, "Point and go: Intuitive reference frame reallocation in mode switching for assistive robotics," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 1–7. DOI: 10.1109/ICRA55743.2025.11127426.

[43] J. Mahler et al., "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[44] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[45] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," *arXiv preprint arXiv:1805.01954*, 2018.

[46] A. Mandlekar et al., "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[47] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9329–9338.

[48] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.

[49] P. Florence et al., "Implicit behavioral cloning," in *Conference on robot learning*, PMLR, 2022, pp. 158–168.

[50] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[51] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 461–11 471.

[52] C. Corneanu, R. Gadde, and A. M. Martinez, "Latentpaint: Image inpainting in latent space with diffusion models," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2024, pp. 4334–4343.

[53] C. Hao, K. Lin, Z. Xue, S. Luo, and H. Soh, "Disco: Language-guided manipulation with diffusion policies and constrained inpainting," *IEEE Robotics and Automation Letters*, 2025.

[54] Y. Wang, Z. Liu, L. Yang, and P. S. Yu, "Conditional denoising diffusion for sequential recommendation," in *Pacific-Asia conference on knowledge discovery and data mining*, Springer, 2024, pp. 156–169.

[55] Z. Zhan et al., "Conditional image synthesis with diffusion models: A survey," *arXiv preprint arXiv:2409.19365*, 2024.

[56] S. Noh et al., "3d flow diffusion policy: Visuomotor policy learning via generating flow in 3d space," *arXiv preprint arXiv:2509.18676*, 2025.

[57] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[58] A. Mitko and J. Fischer, "When it all falls down: The relationship between intuitive physics and spatial cognition," *Cognitive research: principles and implications*, vol. 5, no. 1, p. 24, 2020.

[59] G. Borghesan, E. Scioni, A. Kheddar, and H. Bruyninckx, "Introducing geometric constraint expressions into robot constrained motion specification and control," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1140–1147, 2015.

[60] X. Sun, Y. Wang, S. Yang, Y. Chen, and D. Rakita, "Hybrid diffusion policies with projective geometric algebra for efficient robot manipulation learning," *arXiv preprint arXiv:2507.05695*, 2025.

[61] Z. Li, R. Yang, R. Chen, Z. Luo, and L. Chen, "Adpro: A test-time adaptive diffusion policy via manifold-constrained denoising and task-aware initialization for robotic manipulation," *arXiv preprint arXiv:2508.06266*, 2025.

[62] W. Tang et al., "Geomanip: Geometric constraints as general interfaces for robot manipulation," *arXiv preprint arXiv:2501.09783*, 2025.

[63] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas, "Vector neurons: A general framework for so (3)-equivariant networks," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 200–12 209.

[64] A. Simeonov et al., "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 6394–6400.

[65] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas, "Reinforcement learning with augmented data," *Advances in neural information processing systems*, vol. 33, pp. 19 884–19 895, 2020.

[66] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[67] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[68] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, Elsevier, 1988, pp. 139–183.

[69] N. Carion et al., *Sam 3: Segment anything with concepts*, 2025. arXiv: 2511.16719 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2511.16719.

[70] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5738–5746.