



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Collaborative Robotics: From Predictive Planning to Adaptive Grasp

Dissertation

zur Erlangung des akademischen Grades

Dr. rer. nat

an der Fakultät für Mathematik, Informatik und

Naturwissenschaften

der Universität Hamburg

Eingereicht beim Fachbereich Informatik

von **Jianzhi Lyu**

Januar, 2025

Gutachter: Prof. Dr. Jianwei Zhang
Prof. Dr. Frank Steinicke

Tag der Disputation: 20.06.2025

Abstract

Collaborative robots could revolutionize industrial and domestic environments by enabling seamless human-robot collaboration. However, ensuring both high efficiency and safety in dynamic, tightly shared workspaces remains a significant challenge. This thesis addresses these challenges through an integrated suite of frameworks that enhance both predictive motion planning and dexterous grasping capabilities, paving the way for more fluid, efficient and safe human-robot collaboration.

We begin with the development of Pred-HRC-Body, a novel pipeline that leverages recurrent neural networks and Gaussian mixture models to forecast human multi-joint arm trajectories and estimate intended targets early in the motion. By fusing observed and predicted hand palm trajectories, the framework generates collision-free, goal-oriented robot trajectories with trajectory optimizer that improve both safety and operational efficiency in tightly shared environments.

Building upon this foundation, Pred-HRC-EEG integrates electroencephalography signals using a recently developed brain-computer interface paradigm. This system capitalizes on gaze direction derived from steady-state visual evoked potentials to anticipate human actions even before human start moving. Robot velocity limits are also adjusted with vigilance metrics calculated from EEG signals to enhance human-robot collaboration safety. By dynamically modulating the robot’s velocity and behavior based on real-time cognitive state assessments, this approach not only enhances the fluency of interactions but also further elevates safety standards in collaborative tasks.

To address the parallel critical challenge of dexterous manipulation for collaborative robots, we introduce ADG-Net, a sim-to-real multimodal learning framework designed for adaptive grasping. By integrating RGB-D image, joint angles, and tactile feedback—including fingertip deformation of the robotic hands—ADG-Net predicts and optimizes grasp parameters in an adaptive manner. Extensive benchmark experiments in both simulated and real-world settings demonstrate that our approach achieves superior grasp success rates in scenarios involving both isolated and cluttered objects.

Finally, to seamlessly integrate motion planning and grasping in dynamic human-robot collaboration environments, we propose an integrated framework—RL-DNLS, which combines Actor-Critic reinforcement learning with differentiable nonlinear least-squares optimization. In this approach, reinforcement learning and differentiable trajectory optimization complement each other: the RL module explores optimal goal positions and adjusts cost weights, while the DNLS optimizer refines trajectories to satisfy kinodynamic and safety constraints. This synergistic combination ensures robust and optimal joint planning for reaching and grasping actions in the dynamic shared workspace.

Collectively, the contributions of this thesis establish a comprehensive approach to developing collaborative robots with human-like predictive intelligence and dexterous manipulation skills, fostering safer, more efficient, and fluent human-robot collaboration in both industrial and everyday settings.

Zusammenfassung

Kollaborative Roboter könnten industrielle und häusliche Umgebungen revolutionieren, indem sie eine nahtlose Mensch-Roboter-Kollaboration ermöglichen. Allerdings stellt die Gewährleistung sowohl hoher Effizienz als auch Sicherheit in dynamischen, eng genutzten Arbeitsbereichen eine erhebliche Herausforderung dar. Diese Arbeit geht diese Probleme durch ein integriertes Framework an, das sowohl die prädiktive Bewegungsplanung als auch die geschickte Greiffähigkeit verbessert und so den Weg für eine flüssigere, effizientere und sicherere Zusammenarbeit zwischen Mensch und Roboter ebnet.

Zunächst wird Pred-HRC-Body vorgestellt, eine neuartige Pipeline, die rekurrente neuronale Netzwerke und Gaußsche Mischmodelle einsetzt, um menschliche Mehrgelenkarmtrajektorien vorherzusagen und beabsichtigte Ziele bereits zu Beginn der Bewegung zu schätzen. Durch die Fusion von beobachteten und prognostizierten Handflächenbewegungen erzeugt dieses Framework kollisionsfreie, zielgerichtete Robotertrajektorien mithilfe eines Trajektorienoptimierers, wodurch sowohl die Sicherheit als auch die Betriebseffizienz in eng geteilten Umgebungen verbessert wird.

Auf dieser Grundlage integriert Pred-HRC-EEG Elektroenzephalographie-Signale mittels eines kürzlich entwickelten Brain-Computer-Interface-Paradigmas. Das System nutzt die aus steady-state-visuell evozierten Potentialen abgeleitete Blickrichtung, um menschliche Aktionen bereits vor deren Einsetzen zu antizipieren. Zudem werden die Geschwindigkeitsbegrenzungen des Roboters anhand von Wachsamkeitsmetriken, die aus EEG-Signalen berechnet werden, dynamisch angepasst, wodurch die Sicherheit in der Mensch-Roboter-Kollaboration weiter erhöht und die Interaktionsflüssigkeit verbessert wird.

Um die ebenso wesentliche Herausforderung der geschickten Manipulation durch kollaborative Roboter zu bewältigen, wird ADG-Net vorgestellt – ein sim-to-real, multimodales Lernframework für adaptives Greifen. Durch die Integration von RGB-D-Bildern, Gelenkwinkeln und taktilem Feedback – einschließlich der Deformation der Fingertips der Roboterhände – prognostiziert und optimiert ADG-Net adaptiv Greifparameter. Umfangreiche Benchmark-Experimente in simulierten und realen Umgebungen belegen, dass dieser Ansatz überlegene Erfolgsraten beim Greifen in Szenarien mit isolierten sowie in Unordnung befindlichen Objekten erzielt.

Abschließend wird zur nahtlosen Integration von Bewegungsplanung und Greifen in dynamischen Mensch-Roboter-Kollaborationsumgebungen ein integriertes Framework namens RL-DNLS vorgeschlagen, das Actor-Critic-Verstärkungslernen mit einem differenzierbaren nichtlinearen Kleinste-Quadrate-Optimierer kombiniert. In diesem Ansatz ergänzen sich das Verstärkungslernen und die differenzierbare Trajektorienoptimierung: Das RL-Modul ermittelt optimale Zielpositionen und passt die Kostenfaktoren an, während der DNLS-Optimierer die Trajektorien verfeinert, um kinodynamische sowie sicherheitsrelevante Vorgaben zu erfüllen. Diese synergetische Kombination gewährleistet eine robuste und optimale gemeinsame Planung von Reich- und Greifbewegungen im dynamischen, geteilten Arbeitsbereich.

Insgesamt etabliert diese Arbeit einen umfassenden Ansatz zur Entwicklung kollaborativer Roboter, die über eine menschenähnliche, prädiktive Intelligenz und

geschickte Manipulationsfähigkeiten verfügen. Damit wird eine sicherere, effizientere und flüssigere Mensch-Roboter-Kollaboration in industriellen wie auch alltäglichen Umgebungen gefördert.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Focus	4
1.3	Scientific Contributions	5
1.4	Thesis Structure	7
2	Related Work	9
2.1	Online Motion Planning	9
2.1.1	Human Motion Prediction and Intention Estimation	9
2.1.2	Online Trajectory Optimization	13
2.2	Static Dexterous Grasping	13
2.2.1	Simulation of Dexterous Grasping	13
2.2.2	Sim2Real Learning of Dexterous Grasping	14
2.2.3	Multimodal Sensing for Dexterous Grasping	15
2.3	Joint Motion and Grasping Planning	15
2.3.1	Decoupled Grasp Generation and Motion Planning	15
2.3.2	Joint Grasp and Motion Planning	16
3	Predictive Planning for HRC Using Human Motion	18
3.1	Introduction	18
3.2	Methodology	21
3.2.1	Human Trajectory Prediction and Target Estimation	21
3.2.2	Online Trajectory Generation	23
3.3	Experiments	26
3.3.1	Trajectory Prediction and Target Position Estimation	26
3.3.2	Online Trajectory Generation	28
3.3.3	HRC Efficiency	31
3.4	Summary	33
4	Predictive Planning for HRC Using EEG	34
4.1	Introduction	34
4.2	Methodology	35
4.2.1	Setup and task description	35
4.2.2	Participants	36
4.2.3	Experimental conditions	37

4.2.4	Brain-computer interface	37
4.2.5	Robot trajectory generation	38
4.3	Experiments	42
4.3.1	HRC performance	42
4.3.2	Safety distance analysis	43
4.3.3	Target prediction accuracy and vigilance	45
4.4	Discussion	47
4.4.1	BCI can improve HRC efficiency and enhance safety	48
4.5	Summary	49
5	Sim2Real Adaptive Dexterous Grasping	50
5.1	Introduction	50
5.2	Methodology	52
5.2.1	Motivations	52
5.2.2	Coordinate Systems	52
5.2.3	Two-Stage Grasp Simulation	52
5.2.4	Multimodal Learning for Grasp Prediction	53
5.2.5	Multimodal Adaptive Grasping	54
5.3	Two-Stage Grasp Simulation	54
5.3.1	Overview	54
5.3.2	Grasping Scenario Modeling	55
5.3.3	First Stage: Fast Contact	56
5.3.4	Second Stage: Grasp Optimization and Tactile Sensing	58
5.3.5	Tactile Sensing and Grasp Quality Evaluation	59
5.3.6	Dataset Generation	61
5.4	Multimodal Learning for Grasp Prediction	62
5.4.1	Architecture of ADG-Net	62
5.4.2	Training of ADG-Net	63
5.5	Multimodal Adaptive Grasping	66
5.6	Experiments	67
5.6.1	Experiments in Simulation	67
5.6.2	Experiments in the Real World	70
5.6.3	Failed Trials and Limitations	73
5.7	Summary	74
6	Joint Motion and Grasp Planning for HRC	75
6.1	Introduction	75
6.2	Methodology	76
6.2.1	General Formulation of MPC	77
6.2.2	Actor-Critic RL	77
6.2.3	Integrating Differentiable MPC with RL	78
6.2.4	Dynamic Grasp in Shared Human-Robot Workspace	80
6.3	Experiments	83
6.3.1	Simulation Evaluation	83
6.3.2	Real-World Experiments	88

6.4	Summary	91
7	Conclusions and Future Work	94
7.1	Limitations	95
7.2	Future Research	96
A	List of Abbreviations	99
B	Publications	101
C	Acknowledgements	103
	Bibliography	105

List of Figures

1.1	Fenced and fenceless HRC	2
1.2	HRC scenarios	3
1.3	Two strategies of HRC	4
1.4	The structure of the thesis.	8
2.1	Human and PR2 collaboration	10
2.2	Simulus and timing of EEG experiments	12
2.3	Grasps generated by DexGraspNet	14
2.4	SE(3)-DiF’s architecture	17
3.1	Side view of the HRC experiment	19
3.2	Placement of the target and initial positions	20
3.3	Collected human palm trajectories	20
3.4	The architecture of our proposed HRC pipeline	21
3.5	Encoder-Decoder neural network	22
3.6	Separation plane	25
3.7	Analysis of trajectory prediction and target estimation	27
3.8	Confusion matrices for intended position estimation	28
3.9	Experiments with a reactive controller	30
3.10	Experiments with our predictive controller	30
3.11	Shadowhand first finger tip trajectories	31
3.12	Goal scheduling and HRC performace	33
4.1	The proposed HRC framework. [68]	36
4.2	Relationship between the robot and human goals	39
4.3	Geometric model of the work cell	40
4.4	Example scenario of the HRC experiment	41
4.5	Robot relative performance	42
4.6	Safety distance analysis	43
4.7	Velocities between robot and human	44
4.8	BCI classification accuracy	45
4.9	Vigilance monitoring	46
4.10	Vigilance analysis	46
4.11	Relationship between vigilance and HRC performance	47
5.1	Overview of the proposed method	51

5.2	Coordinate systems for dexterous grasp	53
5.3	Framework of the two-stage grasp simulation	55
5.4	24 DOFs of a left ShadowHand. [147]	56
5.5	An example of pre-grasp initialization	57
5.6	Grasping trials	58
5.7	Tactile sensing via FEM	59
5.8	Basic information on a contact point. [147]	61
5.9	Architecture of the proposed ADG-Net	64
5.10	Experiment setups	68
5.11	Prediction errors of joint angles	69
5.12	Part of objects for grasping experiments. [147]	70
5.13	Grasping demos with ADG-Net	71
5.14	Failed grasping trials	73
6.1	Separation plane between Kuka and human	80
6.2	Method overview	82
6.3	Simulation environment	84
6.4	Robot motion in different grasp phases	87
6.5	Value maps from critic network	88
6.6	Jerk magnitude and time cost for computation	89
6.7	Experiment setup	90
6.8	Object pose estimation accuracy	91
6.9	Human body tracking using MediaPipe Pose	92
7.1	Overview of the research in this thesis	95

List of Tables

3.1	Comparison between predictive and reactive controller	31
3.2	Experiment workflow	32
4.1	Trajectory optimizer parameters.	40
5.1	Basic information of the grasping dataset	62
5.2	Prediction errors of two ADG-Nets	69
5.3	Results of the Benchmark Experiments [147]	72
6.1	PPO Hyperparameters	83
6.2	Kinodynamic Parameters	84
6.3	Grasping performance in simulation	85
6.4	Ablation study	86
6.5	Experiment results for different objects	92

Chapter 1

Introduction

1.1 Motivation

Collaborative robots (cobots) are designed to work alongside humans to complete complex tasks. Equipped with advanced sensing technologies—such as force/torque sensors, vision systems, and tactile sensors—these robots continuously monitor and respond to their surroundings. Their sophisticated control systems, which include motion control, collision detection and avoidance, and Human Robot Interaction (HRI) mechanisms, enable safe and effective operation in shared workspaces.

One of the key advantages of cobots is their adaptability and ease of deployment. They can be quickly installed and integrated into a variety of operational scenarios, making them highly versatile for industries such as manufacturing, healthcare, and household robotics. In these settings, cobots assist humans with tasks that require precision, adaptability, and contextual awareness.

A critical feature of cobots is their ability to interact with humans efficiently and safely. Advanced Human Robot Collaboration (HRC) frameworks integrate sensing, planning, and decision-making capabilities so that robots can operate seamlessly and intuitively in shared workspaces, thereby enhancing productivity and user experience. The growing deployment of cobots underscores their potential to revolutionize industries by improving efficiency and enabling safer, more effective collaboration.

HRC can be broadly categorized into two configurations: fenced and fenceless. In fenced HRC (see Fig.1.1 [47]), robots operate within designated workspaces separated from human workers by physical barriers. This arrangement ensures high safety levels by preventing direct contact between humans and robots, allowing robots to operate at higher speeds. However, physical barriers also increase spatial requirements, prolong workflows, and necessitate time-consuming setups or relocations.

In contrast, fenceless HRC allows robots and humans to share the same workspace (see Fig.1.2a and Fig.1.2b). In these scenarios, safety is maintained through advanced sensing systems that detect proximity or contact forces, enabling the robot to halt immediately upon a potential collision. Although this approach maintains safety, frequent stop-and-restart cycles can disrupt the fluidity of interactions and reduce overall collaboration efficiency.

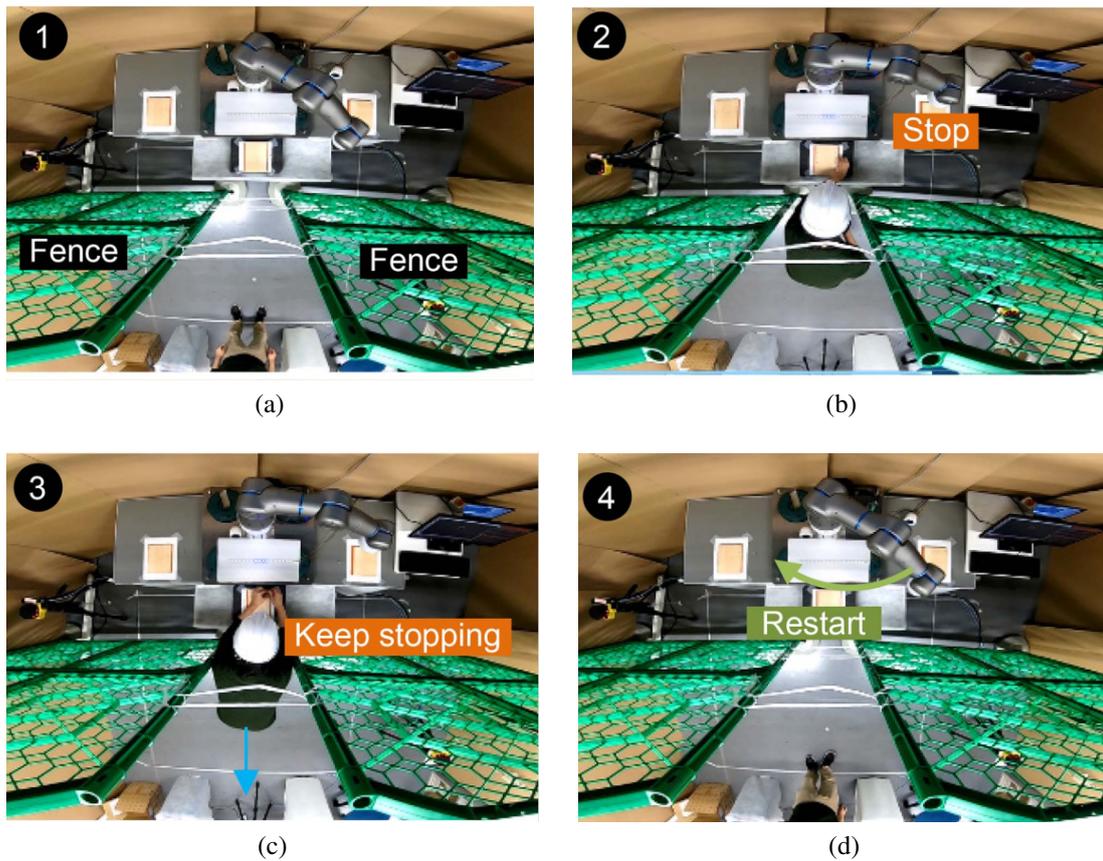


Figure 1.1: HRC with fence [47]. (a) When human is not in the robot workspace, the robot works at its full speed; (b) When human is detected to be inside of robot workspace, the robot will stop moving; (c) The robot will keep static as long as human is inside its workspace; (d) When the human is detected to be outside of the robot workspace, the robot will restart to work again.

To improve the fluency and efficiency of HRC, cobots must move beyond reactive behaviors—such as simply stopping and restarting—and incorporate predictive capabilities. For example, in an assembly line, a cobot working with a human must continuously monitor and predict human actions, adjust its trajectory proactively to avoid collisions, and optimize its task schedule to enhance overall system performance. Achieving this seamless integration requires advanced predictive planning that combines human motion and intention prediction with proactive robot control.

Furthermore, to operate effectively in diverse scenarios—including everyday environments as illustrated in Fig.1.2c and Fig.1.2d—cobots must also possess advanced manipulation capabilities. A critical aspect is their ability to grasp a wide range of objects with varying sizes, shapes, and materials. Grasping is a fundamental yet challenging robotic skill, particularly when relying on a single sensory modality. Vision-based systems, for example, often face difficulties with occlusions, partial object views, and varying lighting conditions. Tactile sensors provide valuable information about contact forces and textures, but they are insufficient on their own. Proprioceptive sensors complement these modalities by offering data on joint positions and forces, ensuring that

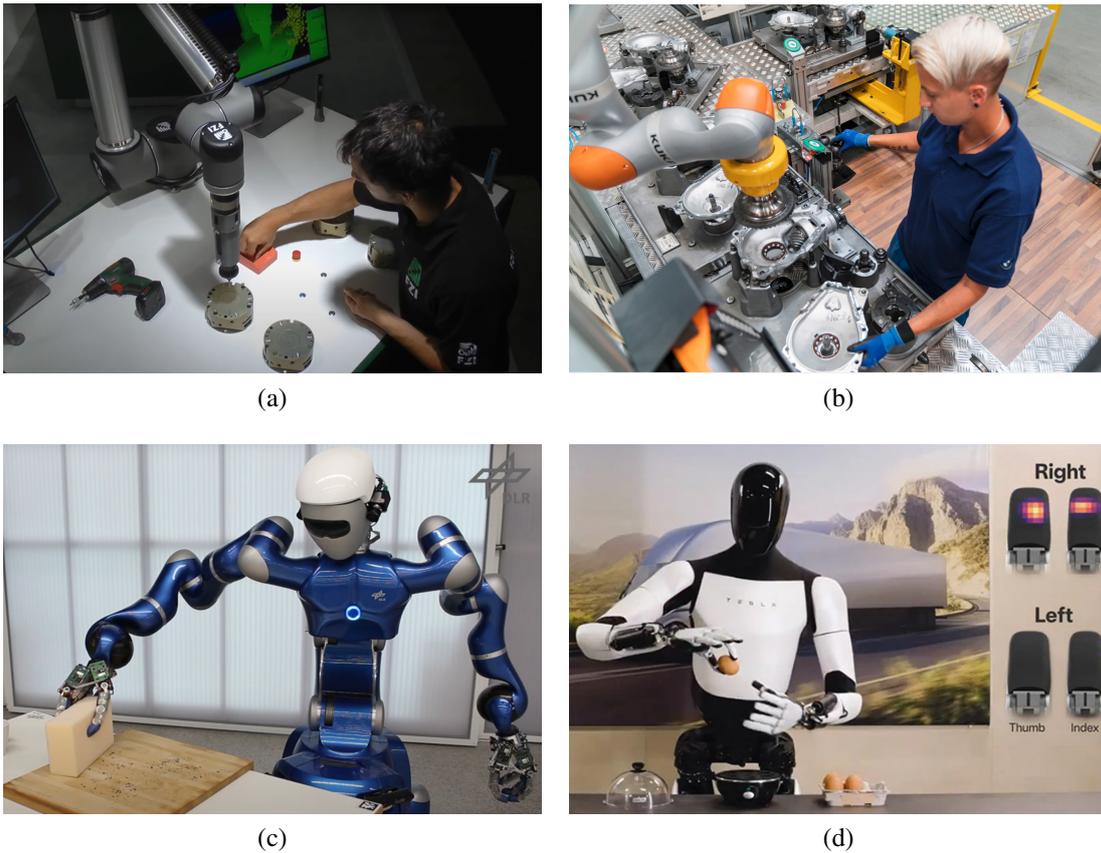


Figure 1.2: HRC without fence. (a) Collaborative screw assembly [44]; (b) HRC in the axle drive assembly [8]; (c) Daily cleaning tasks with dexterous hand [31]; (d) Tesla Optimus Gen2: doing housework with dexterous hand [125].

robots operate safely within their mechanical limits.

Integrating multimodal sensing—combining visual, tactile, and proprioceptive data—is essential to overcome these challenges. By leveraging the complementary strengths of these modalities, cobots can achieve a more comprehensive understanding of object properties and environmental conditions. Multimodal frameworks enable precise and stable grasps even in cluttered or dynamic environments. For instance, tactile feedback can enhance grasp stability by detecting slippage, while visual data assists in dynamic object localization and recognition. Proprioceptive information further ensures safe operation within the robot’s kinematic and dynamic limits.

The dual challenges of enhancing HRC efficiency through predictive planning and improving manipulation capabilities via multimodal sensing form the core motivations of this thesis. This work aims to advance collaborative robotics by addressing these critical areas, ultimately contributing to the development of cobots that are safer, more adaptive, and capable of dexterous grasping in real-world scenarios.

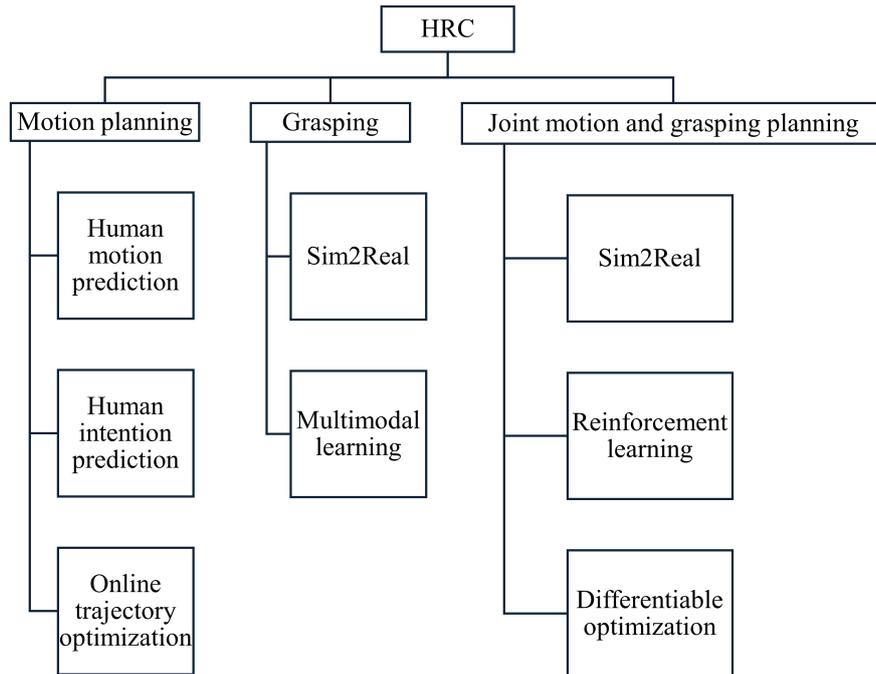


Figure 1.3: Research focuses of this thesis. The first focus of this thesis is to address motion planning and object-grasping tasks separately. The second focus is to develop integrated approaches that solve motion planning and object-grasping tasks jointly.

1.2 Research Focus

This thesis begins by addressing the critical challenge of enhancing the safety and efficiency of HRC within tightly shared workspaces. Achieving an effective balance between safety and efficiency is particularly challenging due to the unpredictable and rapid nature of human motions. In such environments, robots are often required to slow down or even stop to ensure safety, significantly impairing the efficiency of collaboration. The research focuses are illustrated in Fig. 1.3.

The first focus of this thesis is to address motion planning and object-grasping as separate problems. For motion planning, collaborative robots must be equipped with the ability to predict human motions and intentions, which enables predictive trajectory planning and adaptive task scheduling. This approach mitigates the negative impact of human unpredictability on HRC performance. In the object-grasping problem, the focus is on enabling cobots to grasp objects of various sizes and shapes in static scenarios.

Motion Planning To address the motion planning problem in dynamic HRC environments—without considering the object-grasping task—we introduce a framework composed of several modules in which the robot’s trajectory optimization and the human motion/intention prediction modules are treated independently. In this setup, human motion data and electroencephalography (EEG) signals are used to predict human trajectories and intentions. These predictions then update the robot’s task schedule and inform a trajectory optimization planner that generates safe and efficient robot trajectories.

By combining human intention and motion prediction with advanced planning and learning methodologies, this thesis provides a comprehensive framework for improving the fluency and effectiveness of HRC in tightly shared workspaces. This work lays the foundation for safer, more intuitive, and efficient collaboration between humans and robots. However, a significant challenge remains: seamlessly integrating the predicted human motions and intentions into the trajectory optimization process. Finding an optimal method to combine these predictions with trajectory planning remains complex.

Object-Grasping This section focuses on improving the general grasping ability of cobots' dexterous hands by leveraging multimodal information. Only the static object-grasping problem is considered here. Although similar work exists, previous studies have primarily focused on rigid robotic hands without considering fingertip deformation, and they often rely on pressure feedback rather than comprehensive tactile information for grasping. Moreover, training the grasping policy in those approaches tends to be time-consuming. In this thesis, not only is visual information utilized, but hand joint angles and tactile feedback are also incorporated, enabling the dexterous hand to adapt to dynamic environments during grasping. Furthermore, the quality of grasp candidates is calculated and predicted using the proposed method. In addition, a parallel GPU-based simulator is employed to collect a multimodal grasping dataset, and the grasping policy network is efficiently trained using supervised learning on this data.

The second focus of this thesis is to jointly address motion planning and object-grasping in dynamic HRC shared workspaces, thereby overcoming the limitations identified in the first focus. Reinforcement learning (RL) is employed to optimize the integration of human motion and intention predictions with the trajectory optimization process. Specifically, RL is used to determine the optimal goal positions and cost function weights for the trajectory optimizer, ensuring that task constraints are satisfied while maximizing both trajectory quality and grasping success. The trajectory optimizer, built on a differentiable optimization solver, is designed to be trainable jointly with the RL model. This cohesive design enables the system to dynamically adapt to human behavior while maintaining high levels of safety, efficiency, and grasp success in collaborative tasks.

Overall, this thesis focuses on enhancing the integrated motion planning and object-grasping capabilities of cobots to achieve safer and more efficient HRC. Our objective is to develop cobots with human-like intelligence that can collaborate safely, fluently, and efficiently with humans—not only in industrial settings but also in everyday life.

1.3 Scientific Contributions

This thesis makes contributions to the field of safe and efficient dynamic HRC in tightly shared workspaces, as well as to enhancing the dexterous grasping capabilities of collaborative robots (cobots). The primary contributions of this work are summarized as follows:

- We propose a comprehensive HRC pipeline that integrates high-level human body

multi-joint trajectory prediction and intended target estimation with low-level on-line collision-free trajectory generation to address the challenges of dynamic HRC tasks. The pipeline leverages both observed and predicted palm trajectories to enhance the accuracy of final intended target estimation during the early stages of human motion. The proposed HRC pipeline is evaluated through extensive physical experiments. The results demonstrate that the pipeline enables robots to generate goal-oriented, collision-free trajectories that effectively enhance both the efficiency and safety of HRC in tightly shared workspaces. This work is named as **Pred-HRC-Body** (detailed in chapter 3).

- To the best of our knowledge, this is the first study to investigate the integration of Brain-Computer Interaction (BCI) with online trajectory optimization for HRC tasks in a narrow, shared workspace. In this work, we employed a recently developed BCI variant capable of recognizing gaze direction relative to a single flicker stimulus to predict human intention. The steady-state visual evoked potential (SSVEP) signal-to-noise ratio (SNR) was utilized as an indicator of human vigilance, enabling real-time modulation of the robot’s behavior based on the user’s cognitive state. This study also demonstrates the integration of BCI outputs, such as gaze direction and vigilance levels, with human arm tracking data within the robot controller. The resulting system allows for dynamic and adaptive robot behavior in response to both explicit and implicit human mental states, improving the fluency and safety of collaboration. This work is referred to as **Pred-HRC-EEG** (detailed in chapter 4).
- We develop a pluggable two-stage method for dexterous grasp simulation, leveraging the Isaac Gym simulator [74]. This method simulates dexterous grasping with soft fingertips and generates multimodal sensing data. Over 500,000 multimodal grasping scenarios are synthesized to create a comprehensive dataset for training and evaluation.

To advance dexterous grasping capabilities, we propose a versatile network, **ADG-Net**, designed to learn principles for dexterous grasping and status prediction. The ADG-Net integrates an attention mechanism with a Graph Convolutional Neural Network (GCN) for efficient information fusion and multimodal learning. The network operates in two distinct modes:

- **Image-based Mode:** The ADG-Net processes RGB-D images of a grasping scene to detect feasible grasp parameters, including hand pose, joint angles, grasp quality, and grasp status.
- **Multimodal Mode:** In addition to RGB-D images, the ADG-Net incorporates real-time tactile force data from fingertips and joint angles to optimize grasp parameters when the dexterous hand makes contact with an object.

We introduce an adaptive dexterous grasping method based on the ADG-Net, functioning at a control frequency of 5 Hz. This method optimizes grasp parameters using multimodal sensing data, enhancing grasp performance. The proposed approach is validated through extensive benchmark experiments conducted

in both simulated and real-world environments. This work is referred as **ADG-Net** (detailed in chapter 5).

- We present an Actor-Critic differentiable optimization framework specifically designed for dynamic grasping tasks in human-robot shared workspaces. The core of this framework involves reformulating the constrained differentiable Model Predictive Control (MPC) problem as a convex differentiable nonlinear least-squares (DNLS) optimization problem. This reformulation not only ensures computational efficiency but also enables the effective utilization of GPU resources to solve the optimization problem in real-time. The proposed method has been validated through simulations and experiments on real-world robotic systems, showcasing its effectiveness, practicality, and adaptability in dynamic and collaborative environments. By combining reinforcement learning (RL) with the DNLS-based optimization solver, the framework achieves robust performance in dynamic grasping tasks, maintaining high efficiency and safety in shared workspaces. This work is named as **RL-DNLS** (detailed in chapter 6).

1.4 Thesis Structure

The structure of this thesis is summarized in Fig. 1.4. The remainder of the thesis is organized into six chapters, each addressing key aspects of motion planning and dexterous grasping for cobots, as outlined below:

- **Chapter 2 Related Work** This chapter provides a comprehensive review of the state-of-the-art in areas relevant to this thesis. Specifically, it examines prior research on human motion and intention prediction, trajectory optimization for HRC, decoupled grasping generation and motion planning, joint motion and grasping generation, as well as simulation and sim-to-real multimodal learning for dexterous grasping.
- **Chapter 3 Pred-HRC-Body** This chapter introduces the Pred-HRC-Body framework, which enhances the efficiency and safety of HRC in assembly tasks. The framework combines accurate human motion prediction with goal-oriented trajectory generation. Real-world experiments validate the proposed pipeline, demonstrating its effectiveness in practical applications. This work has been published in [69].
- **Chapter 4 Pred-HRC-EEG** This chapter presents the Pred-HRC-EEG framework, which incorporates EEG signals for predicting human intentions and vigilance. The predicted results are seamlessly integrated into an online trajectory optimization framework, enhancing the safety and efficiency of HRC. Real-world experiments confirm the framework’s practicality and reliability. This work has been published in [68].
- **Chapter 5 ADG-Net** This chapter introduces ADG-Net, a multimodal dexterous grasping framework. A large multimodal dataset for dexterous grasping is

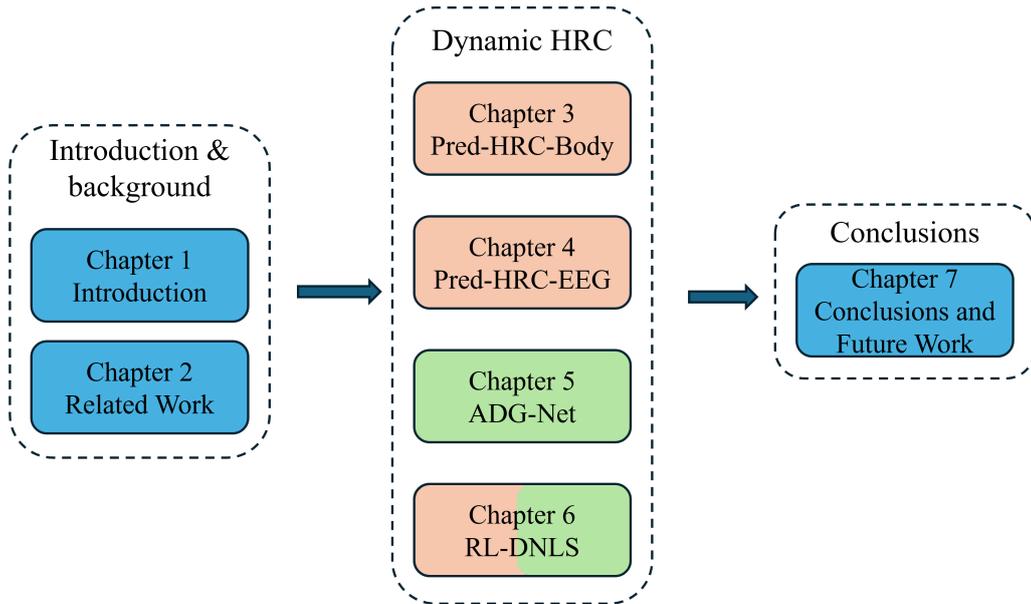


Figure 1.4: The structure of the thesis.

collected using a GPU-accelerated parallel simulator, including RGB-D images, joint angles, and tactile information, with fingertip deformation considered during data generation. ADG-Net employs supervised learning to train a dexterous grasp generation network efficiently. The effectiveness of the proposed framework is validated through both simulation and real-world experiments. This work has been published in [147].

- **Chapter 6 RL-DNLS** This chapter focuses on the integration of RL with differential optimization for safe and efficient dynamic grasping in tightly shared workspaces. The proposed RL-DNLS framework reformulates the constrained optimization problem as a differentiable nonlinear least-squares problem, enabling efficient GPU-based computation. Both simulation and real-world experiments validate its effectiveness. This work has been submitted for publication.
- **Chapter 7 Conclusions and Future Work** The final chapter summarizes the contributions and findings of the thesis and outlines potential directions for future research in motion planning and dexterous grasping in HRC.

Chapter 2

Related Work

Cobots are designed to operate in close proximity to humans, often without the need for extensive safety barriers. In this dynamic HRC environment, online motion planning and dexterous grasping are two fundamental capabilities required of cobots. Due to rapid human movements, online motion planning must incorporate human intention and motion prediction so that cobots can re-plan their trajectories early, ensuring both safety and efficiency in tightly shared workspaces. Meanwhile, dexterous grasping is essential for robustly handling the diverse range of objects encountered in daily life. To facilitate more fluent and efficient HRC, recent research has increasingly focused on solving online motion planning and grasping tasks jointly. Consequently, this chapter reviews the state-of-the-art in online motion planning (human intention and motion prediction, trajectory optimization), static dexterous grasping, and integrated motion-and-grasping planning, highlighting foundational studies and recent advancements that inform the approaches proposed in this thesis.

2.1 Online Motion Planning

In this section, robot motion planning is considered independently of object-grasping tasks. The robot's goal pose is manually scheduled based on human intentions, and the trajectory optimization process accounts for human motions. Accordingly, this section reviews the state-of-the-art in human intention and motion prediction, as well as trajectory optimization.

2.1.1 Human Motion Prediction and Intention Estimation

Human Motion Prediction

Human motion prediction plays a foundational role in HRC, as it enables robots to anticipate human actions and adapt their behavior accordingly. Different methods have been proposed for human motion prediction. Human joint-space trajectories are predicted based on Dynamical Movement Primitives (DMP) and then used to predict human joint torques for intention estimation during walking [102]. Another category of algorithms for AH motion prediction is based on Inverse Optimal Control (IOC), which tries to

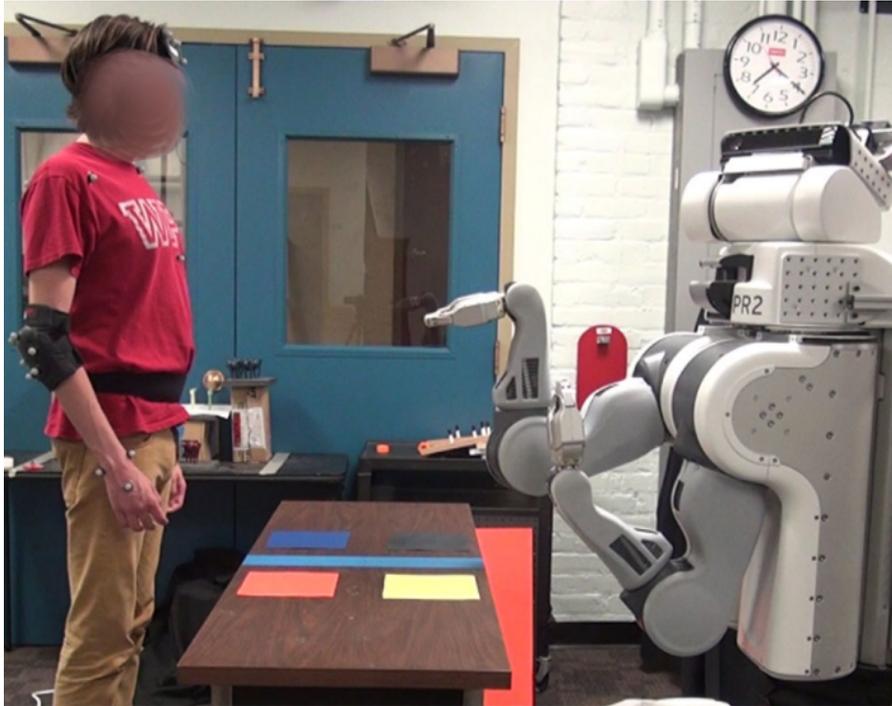


Figure 2.1: The subject and robot are standing opposite to each other. The subject’s task is to touch the yellow or black pads and the robot’s task is to touch either the blue or orange pads [67].

approximate a cost function explaining the observed behavior, e.g., [73]. However, with IOC the goal information needs to be known first, which is not possible for the task we are interested in. Other work predicts human motions with explicitly defined dynamic equations derived from physical theory, such as [5,52,83]. But it is hard to model human dynamics, and the model-based methods usually only work well for a very short time horizon.

There are also some pattern-based approaches. They can learn complex dynamic models from datasets based on all kinds of approximation methods (e.g., neural networks, Hidden Markov Models, Gaussian Mixture Models (GMMs)). Luo et al. used the GMMs to model the human AH trajectory in [67], and the experiment is shown in Fig. 2.1. This unsupervised method can generalize to new persons by dynamically updating or generating new models. Wang et al. proposed a position-velocity recurrent encoder-decoder neural network (PVRED) [132]. A velocity connection is added to the input of the long short-term memory (LSTM), and their results show that this method can achieve a better performance than previous results. This work builds on these approaches by revising and adapting GMMs to improve motion prediction accuracy in scenarios with closely positioned targets.

Intention Estimation based on Human Motion

Intention estimation complements trajectory prediction by providing insights into human goals and actions. Approaches leveraging human body motion have been widely studied. Arpino and Shah predicted the reaching target by time series classification

in [95]. They encoded each time step as a multivariate Gaussian distribution and calculated the class posterior probability with the observed trajectory. The result shows that they can achieve a rather accurate target prediction. A similar idea has been promoted in [67], where GMMs are used to approximate one class of trajectories. In [54], Landi et al. combined the minimum jerk model with an adaptive neural network to predict whether the human will react to the robot end-effector. The similarity between observed short-term movements and the learned user behavior was used to predict human reaching goal in a teleoperation task [133].

The Q-learning method was also used for this task. Cheng et al. [20] proposed that humans optimize a reward function during the pick-and-place task, related to the distance and velocity from the human hand to the target position. Assuming that the human motion follows a Boltzmann policy, they estimated the posterior probability distribution over all targets based on the observed trajectory. However, this method does not work so well when targets are located close to each other (e.g., 10 cm in our scenario), because there will be several similar probable target positions in this situation, especially during the initial motion stage.

Therefore, we will use the probabilistic model GMMs for target estimation instead of end-to-end deep learning methods. The benefit of GMMs is that they are easier to train and also provide us with probability information. As the target positions in our task are close to each other, the trajectories are very similar at the early stage. Unlike the work mentioned above, we make use of both the observed AH trajectory and the short-term prediction as the input of GMMs to improve the estimation accuracy at the beginning of the reaching motion.

Intention and Vigilance Estimation based on EEG

The integration of BCIs with HRC systems has gained significant attention in recent years, particularly for enhancing safety and efficiency in shared workspaces. EEG, a widely-used non-invasive neuroimaging modality, enables decoding of human intentions and cognitive states, providing a valuable channel for human-robot communication. This section reviews related work on EEG-based human intention prediction, highlighting foundational studies and recent advancements that inform the approaches proposed in this thesis.

EEG signals, such as movement-related cortical potentials (MRCPs) and readiness potentials, offer early markers of human movement intention. These slow cortical potentials, detectable several hundred milliseconds before movement onset, have been utilized for predicting the initiation and type of human movement. For instance, readiness potentials have been employed to predict which arm a human will move, enabling robots to adjust their speed and trajectories in response to human actions in shared workspaces [11, 28]. Additionally, MRCPs have been demonstrated to provide information about upcoming grasp actions, such as distinguishing between palmar and pinch grasps, enhancing robot adaptability in collaborative scenarios [141].

Motor imagery, involving the mental simulation of movements without physical execution, is another prominent EEG paradigm for intention decoding. Imagining movements of the hands, feet, or tongue generates distinct EEG patterns, which can be clas-

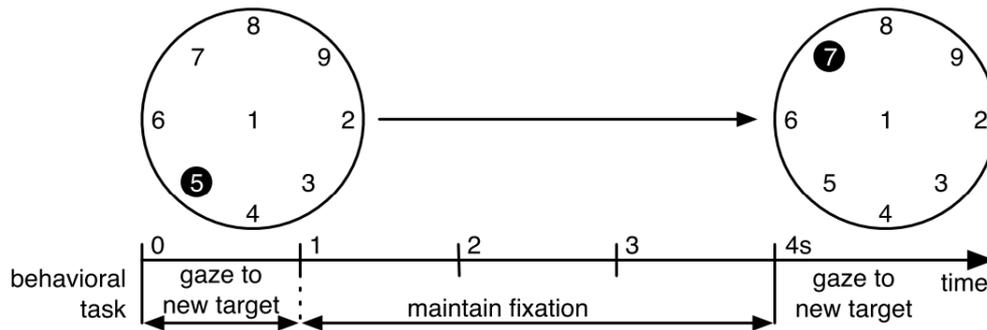


Figure 2.2: The stimulus (top) and timing (bottom) of the experiment are illustrated. A large white disc, resembling a clock face, flickers at the specified stimulation frequency, while targets are highlighted against a steady black background. In the example, the target sequence displayed is “5” followed by “7” [78].

sified with high accuracy using machine learning methods, including deep neural networks [41, 146]. This paradigm has been applied to various applications, such as robotic teleoperation and assistive technologies, where decoding motor imagery allows users to issue high-level commands to robots.

Steady-state visual evoked potentials (SSVEPs), elicited when individuals focus on flickering visual stimuli, have been widely employed in BCIs for intention prediction and target selection. Traditional SSVEP-based systems often require users to switch gaze between a graphical interface and the workspace, which can disrupt task performance [17, 18, 119]. Recent advancements have introduced spatially-coded SSVEPs, enabling gaze direction decoding relative to a single flickering stimulus [78] as shown in Fig. 2.2. This approach facilitates seamless intention prediction by projecting the BCI interface directly onto the workspace, allowing users to naturally integrate intention signals with their operational tasks. By leveraging the tendency of humans to gaze at intended action locations, these systems improve the intuitiveness and efficiency of BCIs in HRC applications [43, 53].

EEG-based vigilance monitoring has been extensively studied as a means to assess attention levels and detect fatigue during tasks. Variations in the amplitude and signal-to-noise ratio (SNR) of SSVEPs have been associated with changes in attention and cognitive load, making them effective indicators of vigilance [14, 118]. Integrating vigilance monitoring into HRC systems enables robots to dynamically adapt their behavior based on the operator’s cognitive state. For instance, modulating robot speed or task difficulty in response to operator vigilance can enhance both safety and task performance [45]. Most existing studies focus on developing standalone BCI components or evaluating BCI applications in simulated environments [11, 17, 18].

Building on these foundational studies, this thesis investigates a novel approach to integrating EEG-based human intention prediction with online trajectory optimization in HRC. The proposed system not only predicts human movement intentions but also monitors vigilance levels to modulate robot behavior adaptively. The experimental setup validates the system’s effectiveness through real-world evaluations, demonstrating significant improvements in both collaboration efficiency and operator safety.

2.1.2 Online Trajectory Optimization

Safe and efficient trajectory generation is also critical for HRC, particularly in dynamic environments with human co-workers. Only specific motion planning algorithms can deal with the dynamic obstacle avoidance problem, such as trajectory optimization [32] and sampling-based methods [38]. Considering the whole volume of obstacles across all prediction time steps for safe trajectory generation results in conservatively planned trajectories. Zheng et al. [154] propose a framework to deal with this problem. They reformulate the obstacle avoidance problem into two Quadratic Programming (QP) programs. This way, they can generate a collision-free trajectory very fast. However, in some scenarios, e.g., when the separating plane used in their approach is close to vertical, the generated trajectory is not safe anymore because of local minima and the linearized kinematics. In other work like [9], [115], they generated collision-free and custom-preferred waypoints in Cartesian space online, during which the dynamics limitation was not considered. They then control the robot end-effector to track these points.

In [112], a trajectory tracking problem in a static environment was solved. In this thesis, we model a predicted AH trajectory as several moving capsules and solve the trajectory optimization problem in a MPC style. A set of penalty terms are added into cost functions to efficiently generate a smooth and safe trajectory for the dynamic HRC task.

2.2 Static Dexterous Grasping

This section discusses the state-of-the-art in dexterous object-grasping. Here, object grasping is treated as an independent problem separate from motion planning. The focus is solely on grasping stability, while robot reachability and motion optimality are not considered.

2.2.1 Simulation of Dexterous Grasping

The simulation of dexterous grasping plays a crucial role in advancing research and applications for dexterous hands.

One of the core aspects of dexterous grasping simulation is the development of sophisticated algorithms that empower dexterous hands to perform various poses for human-like grasping [24]. Many grasp simulators typically synthesize dexterous grasp by defining a grasp quality metric to optimize hand kinematics and force-torque closures, such as GraspIt [80], SynGrasp [75] and UGG [65]. Recent advancements utilize differential optimizers to identify feasible dexterous grasps via differentiable force closures [16, 63, 135, 142]. Nevertheless, most works synthesize dexterous grasps for rigid-body hands, grippers and objects, ignoring their deformations due to the high computational complexity of dexterous grasp simulation [16, 65, 75, 80, 135, 142]. Some dexterous simulation frameworks [63, 135, 142] analyze contacts of dexterous hands using a set of limited contact candidates in front of fingers to accelerate simulation speed. These simplifications may result in synthetic grasps that lack the flexibility of a real human hand.



Figure 2.3: The diverse grasps on the objects from DexGraspNet [135].

In recent years, integrating tactile sensing technologies becomes important for enhancing robotic hand dexterity [151]. Many state-of-the-art (SOTA) tactile sensors, such as BioTac tactile sensors [108] and GelSight sensors [34], are mounted on soft base materials. Hence, simulation with soft-body sensors is necessary for tactile sensing. Narang et al. [88] developed a sim-to-real (Sim2Real) framework for the BioTac tactile sensor to learn representations of tactile sensing with various deformation, using the Isaac Gym and finite element methods (FEMs). Similarly, a novel simulation method was proposed for the tactile simulation of GelSight sensors [19]. However, traditional soft-body simulation methods based on FEMs face computational challenges, limiting their effectiveness for the large-scale dataset generation.

This thesis proposes a two-stage simulation method that combines rigid-body modeling with soft fingertips to generate diverse multimodal grasping scenarios efficiently. A dexterous hand is modeled with a rigid-body hand to quickly touch a target object, and then modeled with a rigid-body hand and soft fingertips to continue optimizing the grasp and collecting tactile forces from the soft fingertips. Owing to the combination of the two simulation stages, massive dexterous grasping scenarios with multimodal sensing data can be simulated effectively.

2.2.2 Sim2Real Learning of Dexterous Grasping

Sim2Real learning of dexterous grasping aims to bridge the gap between synthetic dexterous grasps and real-world applications. Briefly, this learning approach can be classified into two categories according to their foundational frameworks. The first category collects synthetic grasp examples in simulation and then utilizes various deep neural networks to learn grasp principles for detecting grasp parameters [89, 117, 153]. For these methods, collecting a dataset in simulation is a pivotal procedure. Therefore, it is necessary to restrict hand motions in the simulation to prevent neural networks from learning grasp poses that are impractical for a dexterous hand mounted in real-world settings. Unfortunately, this feature is lacking in some existing simulators [135, 142], which can lead to a Sim2Real gap. Generated diverse grasps in the simulator are visualized in Fig. 2.3.

Another research line leverages RL, and learning targets can be various tasks beyond dexterous grasping [56, 60, 76, 101]. For instance, Liang et al. [60] introduced a Sim2Real RL framework for ShadowHand that detects grasp position based on the PointNet [61, 97] and employs a RL algorithm to adapt grasp principles from simple two-finger jaw grippers to the ShadowHand. Lee et al. [56] developed a DexTouch for Sim2Real learning of dexterous grasping and manipulation with tactile feedback.

This thesis introduces a novel Sim2Real grasp learning with two-stage grasp simulation. A multimodal deep learning (DL) method, instead of an RL method, is employed for grasp learning due to both theoretical and technical aspects. First, RL methods require feasible learning policies. However, creating effective reward functions and learning policies for RL methods of dexterous hands presents challenges due to their intricate kinematic structures [60]. Moreover, dataset collection and retraining are crucial for establishing benchmarks for the dexterous grasping task. For RL methods, the robotic task scenarios have to be reconfigured whenever the learning policies or models are updated, making the dataset non-reusable.

2.2.3 Multimodal Sensing for Dexterous Grasping

Dexterous grasping with multimodal sensing is a critical strategy to enable robots to perceive and interact with their environments, and thus achieve human-like grasping [24, 29, 55, 57]. Such methods typically amalgamate diverse types of sensing data, including visual, tactile, and wrist force-torque sensing, along with joint states of a dexterous hand, to enhance grasp pose detection [57], status monitoring [37, 140] and optimization processes [145]. For instance, Zhang et al. [149] integrated the robotic vision with gripper wrist force-torque data through a GA-CNN reactive grasping method for a soft gripper to grasp moving objects. Additional research reveals that the integration of tactile sensing and gripping forces can markedly refine the accuracy of grasping state recognition for a multifingered hand [140].

This thesis introduces an adaptive dexterous grasping method utilizing the multimodal ADG-Net (Section 5.5). The proposed grasping method initially identifies feasible grasp parameters from an RGB-D image of a grasp scene and then optimizes these parameters based on the feedback of tactile forces and joint angles when the dexterous hand touches a target object.

2.3 Joint Motion and Grasping Planning

The object-grasping task can be classified into static and dynamic scenarios. In static scenarios, as discussed in Section 2.2, the task is typically divided into separate sub-tasks for grasp generation and motion planning. In contrast, in dynamic scenarios such as dynamic HRC, grasp generation and motion planning must be addressed jointly to accommodate the continuously changing environment. This section discusses the state-of-the-art methods for jointly solving motion and grasp planning tasks.

2.3.1 Decoupled Grasp Generation and Motion Planning

Traditional grasping frameworks often decouple grasp generation, selection, and motion planning. Data-driven approaches are widely used to tackle the grasp generation problem. In these studies, the physics simulators, like GraspIt! [81], [90] and Flex [70], are utilized to label large datasets of grasp candidates, which are then used to train neural networks for grasp generation or evaluation using depth images or raw point clouds as

input. Finally, some motion planning framework like MoveIt! [26] will be used to check for a collision-free path by looping the generated grasps in descending order of grasp score [62, 86, 124]. This kind of method is poorly adaptable to dynamic environments. If the object pose or the environment obstacles change, the entire three processes may need to be redone.

Some methods for grasp candidate reachability analysis have been proposed to speed up the process of finding feasible grasps for the robots. CollisionNet [87] is used to classify if the gripper at the grasp poses collides with the environment based on the point cloud of the end-effector, object, and obstacles. A collision-aware Reachability Predictor [64] is trained to predict the collision-free probability of the grasp poses. While these methods can accelerate the process of collision-free grasp selection, they fail to consider the grasp optimality in motion planning. For instance, with the selected collision-free grasp candidate, it remains uncertain whether the generated trajectory is both feasible and shorter.

2.3.2 Joint Grasp and Motion Planning

Integrating motion planning and grasp planning is crucial, as these processes are inherently interdependent in object-grasping tasks. Their joint optimization can significantly enhance the success rate and efficiency of such tasks.

Simple heuristic methods have been used to select a grasp pose for motion planning from the multiple grasp candidates provided by existing grasp synthesis methods. The grasp pose closest to the robot arm's end-effector is used as the target pose, followed by trajectory optimization using MPC [143, 144]. These algorithms perform well in human-robot handover scenarios. However, in more complex environments with obstacles, such as humans operating in tightly shared workspaces with robots, the grasp poses selected by simple heuristic methods are often suboptimal or even infeasible. Using the nearest grasp pose as the goal for motion planning can result in the trajectory planner failing to generate a feasible path for the robot to reach the selected grasp pose.

Joint grasp and motion planning algorithms have been proposed recently. For example, CHOMP [106] is employed to compute collision-free trajectories for a set of grasp candidates while simultaneously updating the grasp candidate distribution during trajectory optimization. The cost function weights in the trajectory optimization problem significantly impact the grasping task and must be carefully tuned to achieve optimal performance. In [42], a time-optimal trajectory is computed for each grasp candidate, and the grasp pose that minimizes motion time while avoiding obstacles is selected as the optimal candidate. However, the algorithm relies on sequential quadratic programming (SQP) to solve the optimization problem for each candidate, resulting in a significant computational load that limits its applicability in dynamic scenarios.

Different from outputting a discrete grasp candidate set, the continuous manifold of valid grasp poses is modeled as the level set of a neural implicit function [138]. The neural implicit function takes the 6D query pose and object point cloud as input and estimates the unsigned distance to the nearest valid grasp on the manifold. It is used as the cost function for the trajectory waypoints during gradient-based optimization. This method still has an issue: it is challenging to find appropriate weights for different cost

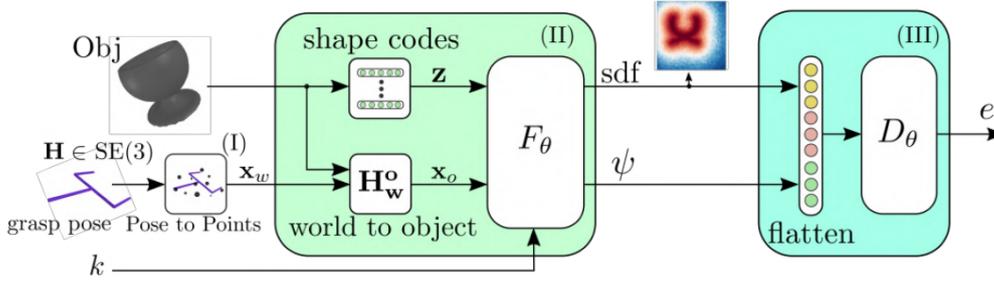


Figure 2.4: SE(3)-DiF’s architecture for learning 6D grasp pose distributions [129].

functions, ensuring that the trajectory planning algorithm can both grasp the object and avoid collisions. Similarly, a smooth SE(3) cost function is learned by a differentiable diffusion model [129] and integrated into trajectory optimization algorithms as a cost function. The architecture of the proposed method is shown in Fig. 2.4 The same issue with weight adjustment still persists.

Numerous studies have already employed RL for object-grasping. The trained policy is used directly for joint grasp and motion planning. RGB images are used as the observation in [46, 77, 105]. More recently, Point clouds have been preferred over images for RL-based object-grasping because they provide rich, direct 3D information that is critical for accurately planning and executing grasps in a three-dimensional space. PointNet [98] and PointNet++ [99] are used to extract object point clouds features in [100] and [134], respectively. The trained policy is capable of grasping both static and dynamic objects, but it does not account for collision issues. A RL-based framework is proposed for human-to-robot handover tasks in [23]. Human hand safety is considered by assigning negative rewards when collisions occur. The negative reward for collision penalties can reduce the frequency of collisions, but it does not guarantee collision-free behavior as shown in [23].

Differentiable optimization layer [30] proposes an efficient way to add a learnable optimization layer in neural networks to include structural priors. This approach has been applied to a wide range of robotic and computer vision learning tasks. Building on this concept, Differentiable MPC [3] has been developed and evaluated on simple control tasks, such as pendulum and cartpole systems. By leveraging the differentiable property of the optimization layer, it can also be integrated into RL, though it slows down the training process significantly [110]. A GPU acceleration optimization layer for DNLS problems is proposed in [96], which could speed up the training process a lot. In this thesis, we use RL to provide the goal pose and tune the cost function weights of the differentiable MPC controller. So that we could get a better performance in dynamic grasping tasks within highly dynamic environments. To further accelerate training, the differentiable MPC optimization problem is reformulated as a DNLS optimization problem and solved using a DNLS optimizer proposed in [96].

Chapter 3

Predictive Planning for HRC Using Human Motion

HRC plays a critical role in both manufacturing and domestic robotics, yet achieving both high efficiency and safety remains a formidable challenge. This chapter introduces an HRC framework that produces collision-free, efficient robot trajectories by leveraging predictions of human AH movements. Specifically, we develop a recurrent neural network (RNN) trained on the initial segments of observed AH trajectories to forecast future motions. To enhance early target estimation, our approach fuses the observed hand palm trajectory with its predicted counterpart using GMMs, thereby identifying the intended motion target. Additionally, we propose an optimization-based trajectory generation algorithm that prioritizes human safety during collaboration. The effectiveness of our system is validated in a shared-workspace scenario involving human pick-and-place tasks, demonstrating that our pipeline can accurately and promptly predict human AH trajectories and estimate the intended motion target, ensuring both safe and efficient collaborative operations.

3.1 Introduction

Robots excel in speed and power, whereas humans possess the intelligence and dexterity required for complex manipulation tasks that can be challenging for robots. As a result, HRC is increasingly adopted to enhance work efficiency and flexibility. However, balancing human safety with robotic efficiency remains a significant challenge. In this work, we consider the scenario depicted in Fig. 3.1, where a human and a robot share a narrow workspace. Although physical interactions such as compliance control (as discussed in [33]) are not considered here, the robot is expected to operate collaboratively with the human. To improve the efficiency of joint assembly tasks and ensure safety in the shared workspace, the robot must be capable of quickly predicting the human AH trajectory and inferring the human’s target position within a short time horizon

We represent human motion using four key joints—shoulder, elbow, wrist, and palm. While previous studies in similar scenarios (e.g., [20, 136, 152]) have attempted to predict human trajectories, they typically focus on just one or two joints, such as the wrist

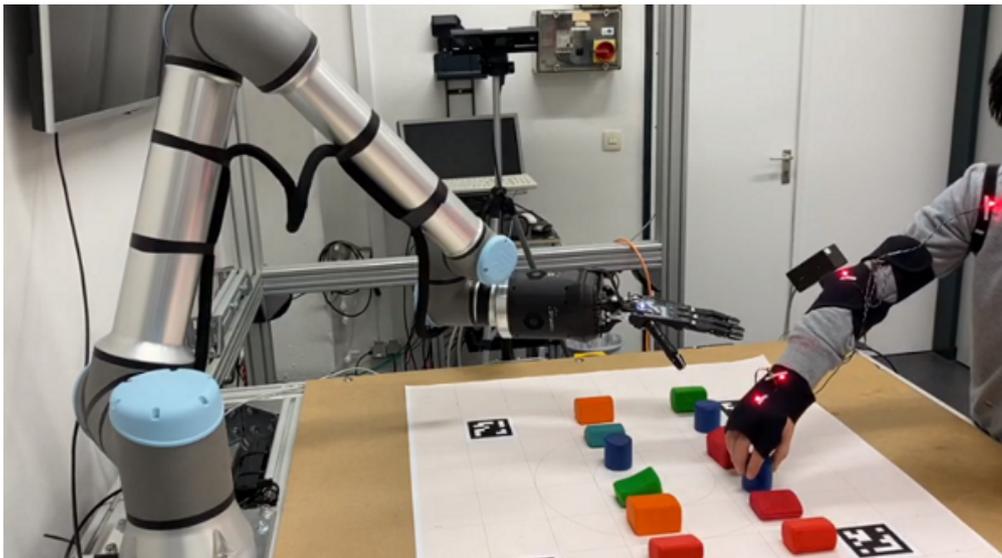


Figure 3.1: A side view of the human–robot pick-and-place platform used in our experiments [69]. In this shared workspace, the human operator places objects at one of twelve designated target locations. Meanwhile, the robot monitors the current AH trajectory to predict both the short-term human motion and the intended reaching target. Using these predictions, the robot generates collision-free, goal-oriented trajectories in real time to collaborate with the human during the assembly task..

and elbow. This limited approach may not suffice for enabling the robot to effectively avoid the human AH, particularly within our confined workspace. In [21], an adaptive method is proposed for predicting the human hand trajectory; however, its applicability to our task is limited due to the high dimensionality of the problem, which complicates online weight adjustment. Overall, accurately modeling the dynamic nature of human AH motion is challenging, especially given individual differences. Consequently, recent state-of-the-art methods (e.g., [36, 132]) have turned to data-driven models. Inspired by [132], we adopt a position-velocity encoder-decoder neural network for AH trajectory prediction.

In principle, a neural network could be trained to simultaneously predict short-term AH trajectories and estimate the intended final target position. However, such a multi-task network would be challenging to train due to the increased number of parameters and the difficulty of incorporating semantic information—such as a set of known target positions—into human intention prediction. Instead, probabilistic methods have been preferred for target inference or motion regression [66, 92] because they generalize well to new scenarios. When the number of possible target positions is limited (in our work, there are 12 targets, as shown in Fig. 3.2), these methods impose a low computational load, making them efficient enough for fast motion prediction. Moreover, probabilistic methods can be trained quickly in an unsupervised manner. It is worth noting that previous work using these methods focused solely on estimating human intentions based on the observed trajectory, which does not fully address the requirements of our task.

In our scenario, the targets are close to each other (separated by 10 cm), and the

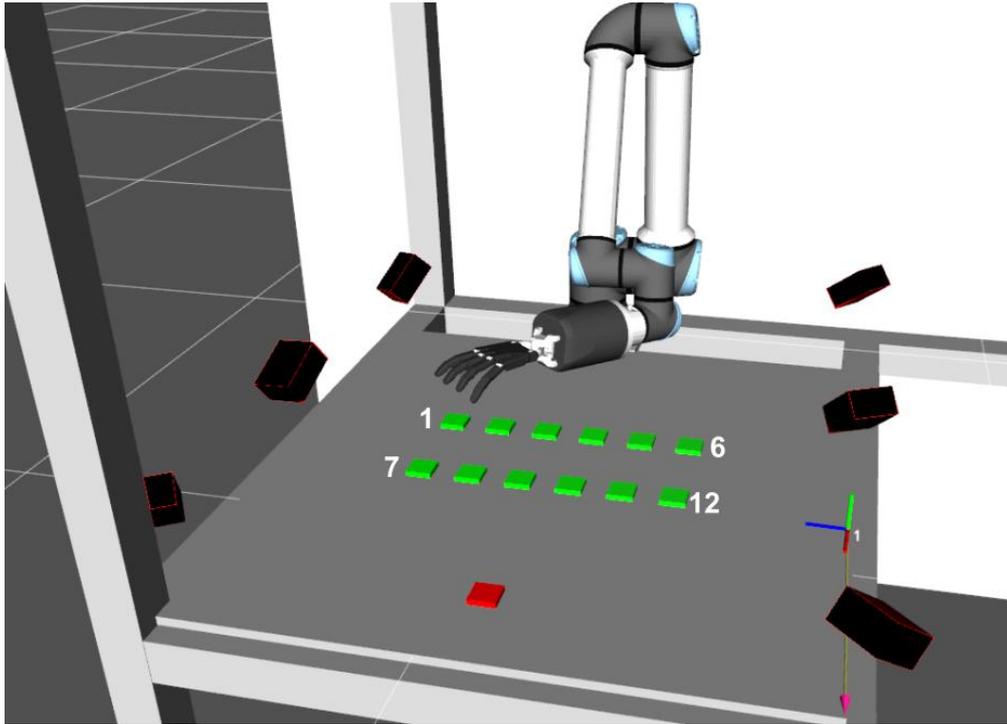


Figure 3.2: Placement of the target and initial positions: The green markers represent the target positions (numbered 1–12), while the red marker at the front denotes the initial (and resting) position of the human hand. The black boxes indicate the locations of the motion-capture system’s cameras used to track human motion. [69]

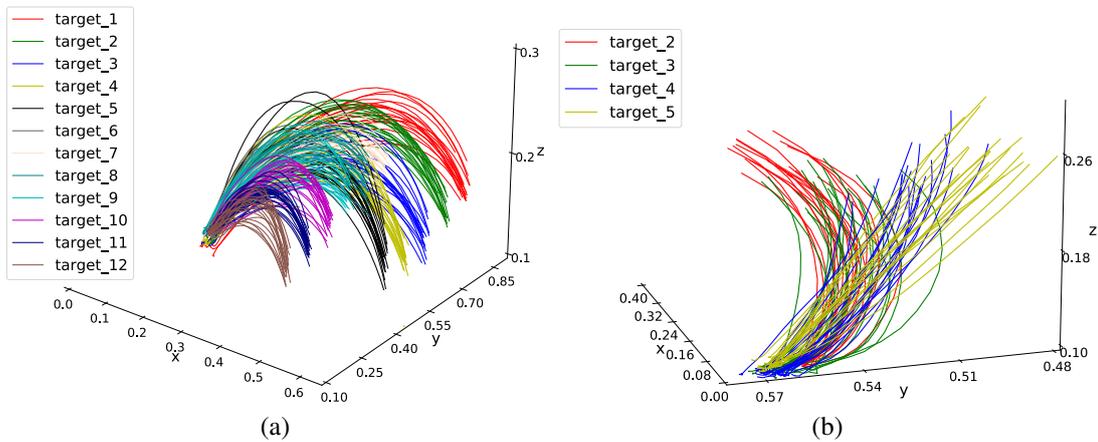


Figure 3.3: (a) Recorded human reaching trajectories of the palm from the resting position to the twelve target positions. (b) A close-up view of the first 40% of several reaching trajectories for target positions 2–5, illustrating the initial overlap among the trajectories. [69]

initial portions of the trajectories (approximately 50%) are very similar, as shown in Figures 3.3a and 3.3b. To improve target estimation accuracy during the early phase of reaching, we propose using both the observed and predicted palm trajectories in conjunction with GMMs.

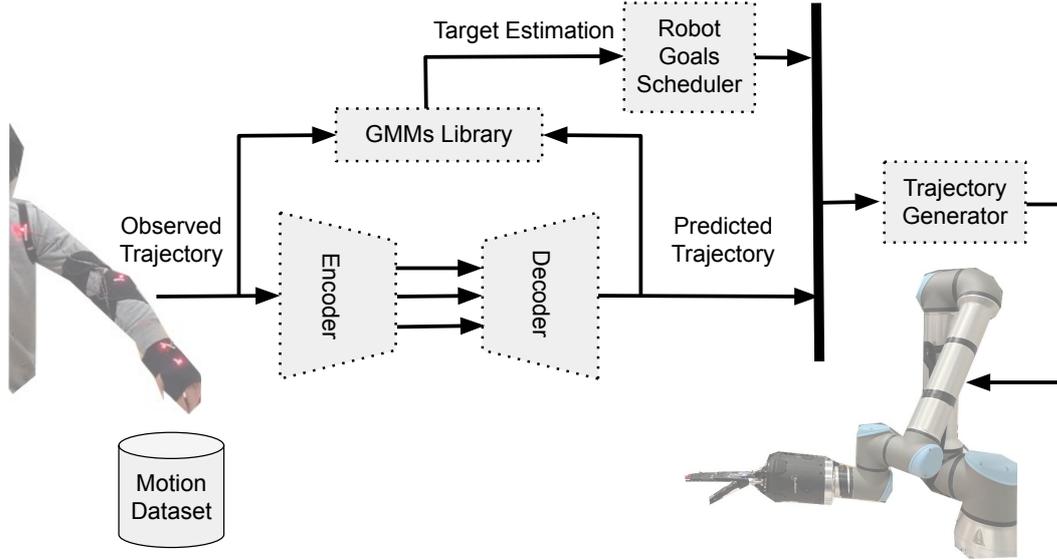


Figure 3.4: The overall design of our HRC framework [69]. It is organized into three primary modules: human trajectory prediction, target inference, and real-time trajectory generation. First, the system employs an encoder-decoder network that analyzes the observed motion of the human arm and predicts its future movement segments. This forecast, combined with the original trajectory data, is then fed into a Gaussian Mixture Model (GMM) component, which deduces the intended destination of the human hand. Both the neural network and the GMM are developed and refined using a custom-collected dataset of human arm movements. Following target estimation, a goal scheduling unit updates the robot’s objectives to align with the inferred human intent. Finally, the online trajectory generator computes collision-free paths to ensure safe and efficient collaboration between the human and the robot.

Previous studies [10, 154] generated safe robot trajectories online by solving two optimization problems, assuming that the AH trajectory prediction was already available. In contrast, by leveraging the predicted AH trajectory from our motion prediction module, we can efficiently generate a safe robot trajectory by solving a single optimization problem with fewer objective functions. Furthermore, to significantly reduce the number of geometric constraints in the trajectory optimization process, we model the AH and robot links using capsules instead of the numerous spheres employed in earlier work.

3.2 Methodology

As described above, our HRC pipeline is organized into three key modules—trajectory prediction, final target estimation, and online trajectory generation—as depicted in Fig. 3.4.

3.2.1 Human Trajectory Prediction and Target Estimation

Predicting the human arm trajectory is a critical first step in our system. This prediction not only helps infer the intended hand position but also enables the controller to generate

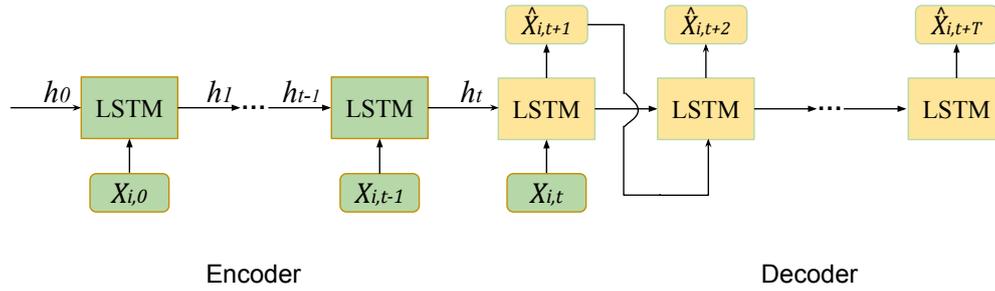


Figure 3.5: The encoder-decoder neural network architecture is composed of two primary modules: an encoder and a decoder. Both modules are implemented using LSTM cells to effectively capture long-term dependencies in the sequential data. [69]

safe, collision-free trajectories.

For trajectory forecasting, we employ an encoder-decoder architecture similar to the Seq2Seq model described in [132]. Our model leverages LSTM networks [40] to capture long-term dependencies in sequential data. Figure 3.5 illustrates the architecture of our encoder-decoder network.

Although the Gaussian Mixture Models (GMMs) introduced later in this chapter could be used for regression tasks [67], our experiments indicate that the trajectories predicted by the GMMs do not closely resemble the actual trajectories. Consequently, we continue to rely on the Seq2Seq model for accurate motion prediction.

The Seq2Seq model is defined as

$$\hat{X}_{i,t+1:t+T} = f(X_{i,0:t}),$$

and is trained on a dataset $\mathcal{D} = \{X_{i,j}\}_{i=1}^N$, where N represents the number of demonstrated trajectories, t is the length of the observed segment, and T is the length of the segment to be predicted. In this context, $X_{i,t}$ denotes the Cartesian positions of four markers placed on the human shoulder, elbow, wrist, and palm. Each trajectory is segmented as $X_{i,0:t+T}$, where the input to the model is the observed segment $X_{i,0:t}$ and the corresponding target labels are given by $X_{i,t+1:t+T}$. Our objective is to train the model so that the predicted trajectory $\hat{X}_{i,t+1:t+T}$ closely approximates the true trajectory $X_{i,t+1:t+T}$.

To achieve this, we define a loss function based on a weighted prediction error over the four markers:

$$\text{Loss} = \sum_{X_{i,0:t+T} \in \mathcal{D}} \sum_{j=1}^4 w_j \left\| \hat{X}_{i,t+1:t+T}^{(j)} - X_{i,t+1:t+T}^{(j)} \right\| \quad (1)$$

where w_j is the weight corresponding to the j -th marker.

Using the dataset \mathcal{D} , we construct a GMMs library $G = \{g_k\}_{k=1}^M$ using the standard unsupervised Expectation Maximization (EM) algorithm, where M denotes the number of potential target objects. For target position estimation, we focus solely on the trajectory of the human palm since it encodes the most pertinent information for trajectory classification [67]. The observed trajectory, denoted as X_{ob} , is represented as an $m \times n$

matrix, where m is the number of waypoints and n is the number of dimensions per waypoint.

To model each component $g_i \in G$, we approximate it using K multivariate Gaussians, $\{gc_k\}_{k=1}^K$. Consequently, the probability that a single trajectory point $X_{i,t}$ (observed at time t in demonstration i) belongs to g_i is given by

$$p(X_{i,t} | g_i) = \sum_{k=1}^K p(gc_k | g_i) p(X_{i,t} | gc_k, g_i) \quad (2)$$

where the conditional probability $p(X_{i,t} | gc_k, g_i)$ is defined by the Gaussian distribution:

$$p(X_{i,t} | gc_k, g_i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} \exp\left(-\frac{1}{2}(X_{i,t} - \mu_k)^T \Sigma_k^{-1} (X_{i,t} - \mu_k)\right) \quad (3)$$

Due to the high similarity of the trajectories during the initial phase of the reaching motion, we utilize both the observed trajectory $X_{i,0:t}$ and the predicted trajectory $\hat{X}_{i,t+1:t+T}$ for target estimation. The joint probability of these trajectories, given a specific target g_i , is calculated as

$$p(X_{i,0:t}, \hat{X}_{i,t+1:t+T} | g_i) = \prod_{s=0}^{t+T} p(X_{i,s} | g_i) \quad (4)$$

Applying Bayes' rule, the log-likelihood of g_i given both the observed and predicted trajectories is expressed as

$$p(g_i | X_{i,0:t}, \hat{X}_{i,t+1:t+T}) = \sum_{s=0}^{t+T} \log p(X_{i,s} | g_i) + \log p(g_i) \quad (5)$$

We then select the g_i with the highest posterior probability as the estimated target position.

3.2.2 Online Trajectory Generation

Our online trajectory generation method, which is based on optimization, enables the robot to perform manipulation tasks while simultaneously avoiding collisions with human arm-hand motions, respecting workspace boundaries, adhering to joint position limits, and satisfying dynamic constraints.

To achieve this, we define a limited quadratic position loss l_P based on the distance between the end-effector position P_E and the goal position P_G . This loss is capped at a maximum value m to ensure that it does not override other critical objectives such as collision avoidance. Additionally, we minimize a quadratic orientation loss l_R that measures the discrepancy between the end-effector orientation R_E and the goal orientation R_G . These loss functions are given by:

$$l_P = \min(m, \|P_E - P_G\|)^2 \quad (6)$$

$$l_R = \|R_E - R_G\|^2 \quad (7)$$

The generated robot trajectory, represented by the joint angles $\{p_i\}_{i=0}^{T_R}$, must adhere to constraints on position, velocity, and acceleration. Specifically, each joint angle p_i is required to lie within its corresponding position limits $(p_{i,\min}, p_{i,\max})$; the joint velocities p'_i must fall between $(v_{i,\min}, v_{i,\max})$; and the joint accelerations p''_i must remain within $(a_{i,\min}, a_{i,\max})$. Here, T_R denotes the length of the robot trajectory generated in each optimization loop. These constraints are formally expressed as:

$$p_{i,\min} < p_i < p_{i,\max} \quad (8)$$

$$v_{i,\min} < p'_i < v_{i,\max} \quad (9)$$

$$a_{i,\min} < p''_i < a_{i,\max} \quad (10)$$

During real experiments, inherent time delays in computations and communications among modules can lead to abrupt changes in the robot's motion when trajectories are updated. To mitigate these motion jumps between consecutive optimization loops, we enforce a constraint that requires the first two steps of any newly generated trajectory to match the corresponding steps in the previous trajectory.

We also add a velocity and acceleration regularizer r_i with weights b, c to prefer smooth motions.

$$r_i = b p_i'^2 + c p_i''^2 \quad (11)$$

Finally, to prevent collisions between the robot and humans—as well as with fixed workspace boundaries—we model human limbs and robot links as capsule-shaped collision objects (i.e., cylinders with hemispherical ends), as shown in Fig. 3.6a. This approach significantly reduces the number of collision geometries compared to the conventional method of using multiple collision spheres.

Specifically, we generate capsules with radii $(r_{H,j})_{j=0}^{T_R}$ between all consecutive human joints with predicted positions $(P_{H,j})_{j=0}^{T_R}$. Similarly, capsules with radii $(r_{R,k})_{k=0}^{T_R}$ are created between all connected robot joints with positions $(P_{R,k})_{k=0}^{T_R}$. We then calculate the pairwise closest distances between these human limb capsules and robot link capsules, as depicted in Fig. 3.6b. The directions of the shortest distance vectors are used as the normals for the separating planes $N_{j,k}$, as illustrated in Fig. 3.6c.

For human–robot collision avoidance, we incorporate a set of penalty terms $q_{j,k}$ to enforce a desired minimum distance d between the human and the robot. We opt for penalty terms rather than hard constraints because a human may approach the robot at a speed exceeding the robot's maximum allowable velocity. In such situations, hard constraints could render the optimization problem infeasible, potentially causing the robot to halt and inadvertently provoking a collision. By contrast, soft penalties allow the robot to continue moving away from the human as rapidly as possible. Furthermore, because the position loss l_p is capped, collision avoidance remains the primary objective over simply reaching the target.

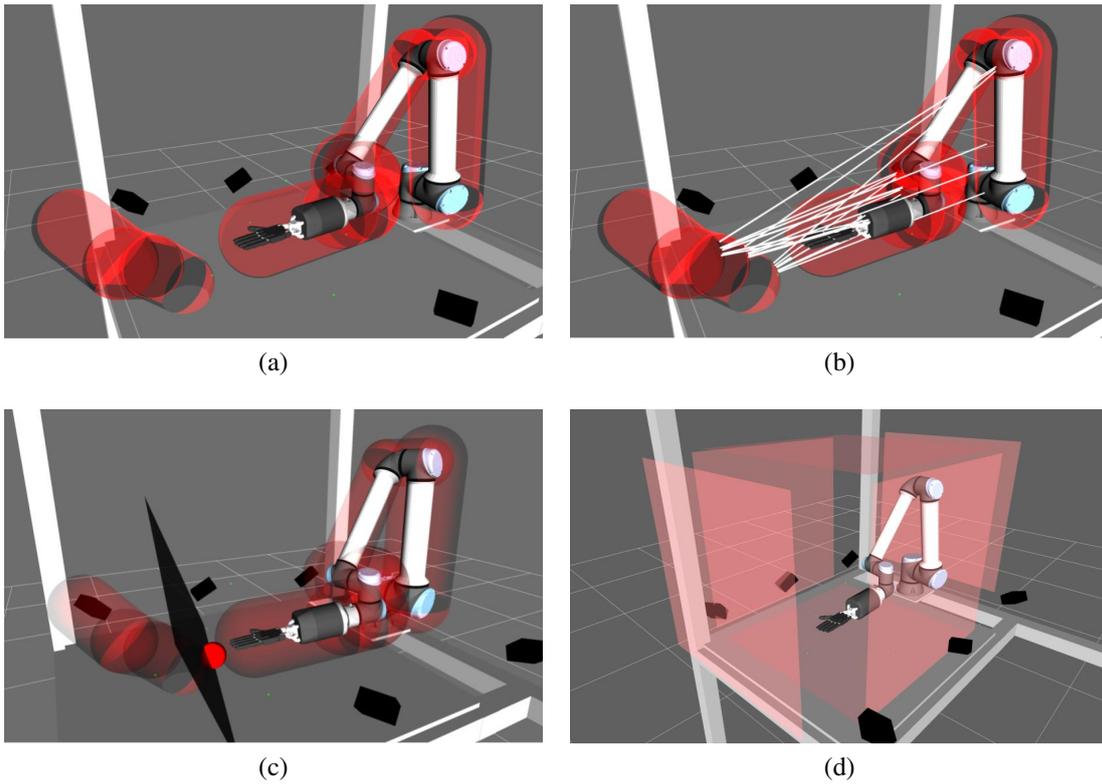


Figure 3.6: Separation Plane Generation and Workspace Constraints for Collision Avoidance. **(a)** We employ capsules as efficient collision geometries, constructing several capsules to encompass both the robot and the human arm. **(b)** The white lines illustrate the pairwise closest distances between each robot capsule and its corresponding human capsule. **(c)** Separation planes are then computed between every human capsule and robot capsule; these planes are integrated into our trajectory optimization process to ensure a safe separation whenever the distance between the human and robot falls below a specified threshold. **(d)** In addition, static workspace boundaries—defined by six planes (four vertical and two horizontal)—constrain the robot’s motion within a designated volume. [69]

In many scenarios, a large open space is available for the robot to retreat into safely. When avoiding human motions, it may be advantageous for the robot to move toward this area. To facilitate this, we introduce an optional bias B . In our experiments, the robot can safely move upward into the open space above the table, so we set the bias B to $[0, 0, 0.5]$.

$$N_{B,j,l} = \frac{N_{j,k} + B}{\|N_{j,k} + B\|} \quad (12)$$

$$q_{j,k} = \min(0, N_{j,k}(P_{R,k} - P_{H,j}) - r_{H,j} - r_{R,k} - d)^2 \quad (13)$$

Finally, the workspace boundaries are enforced by a set of planes with normals $(N_{B,m})_{m=1}^6$ and offsets $(o_{B,m})_{m=1}^6$, as shown in Fig. 3.6d. We add one inequality constraint for each plane and robot sphere.

$$P_{R,k} N_{B,m} < o_{B,m} \quad (14)$$

At each time step, we optimize a trajectory over a horizon of 10 future time steps with a step size of 0.1 seconds. This trajectory is re-optimized at a rate of 10 Hz, incorporating the most recent human motion predictions. Our implementation is based on sequential quadratic programming, which we solve using a primal-dual interior-point method as described in [112].

3.3 Experiments

In this section, we evaluate the proposed method using real human motion data and a robotic system in a desktop assembly task scenario, as illustrated in Fig. 3.1. The experimental results demonstrate that our approach generates collision-free trajectories, thereby enabling the robot to collaborate with the human more efficiently and safely.

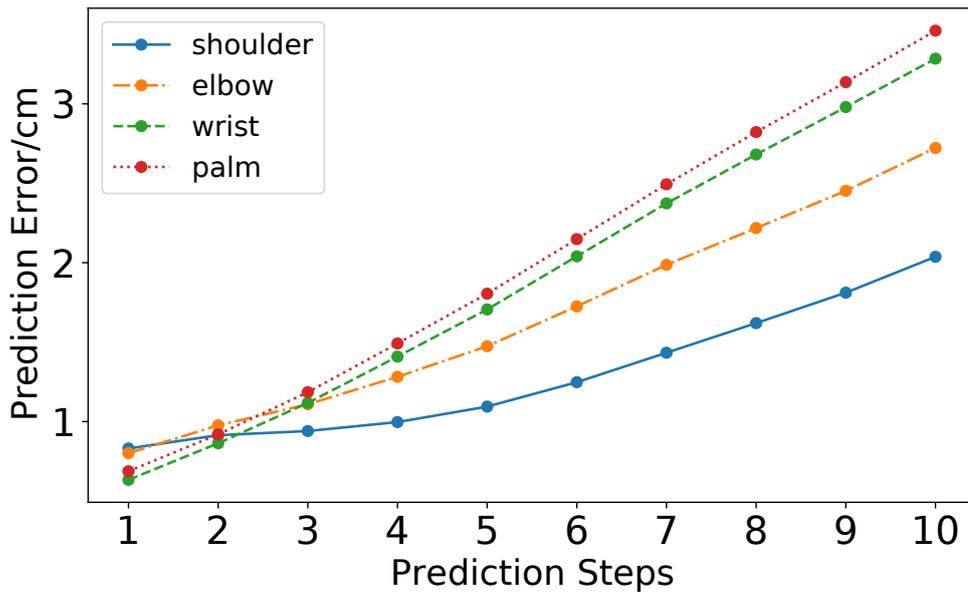
3.3.1 Trajectory Prediction and Target Position Estimation

In our study, we acquired a dataset of pick-and-place trajectories from five healthy participants (four males and one female) of varying heights. Each participant had four LED markers attached to their shoulder, elbow, wrist, and palm (see Fig. 3.1). During the task, their AH motions were captured using a PhaseSpace Impulse X2 motion-capture system. The data were initially recorded at 270 Hz and subsequently down-sampled to 27 Hz to match the input requirements of our Seq2Seq neural network. Each subject performed 240 trajectories, with 20% of the data from each subject reserved for testing.

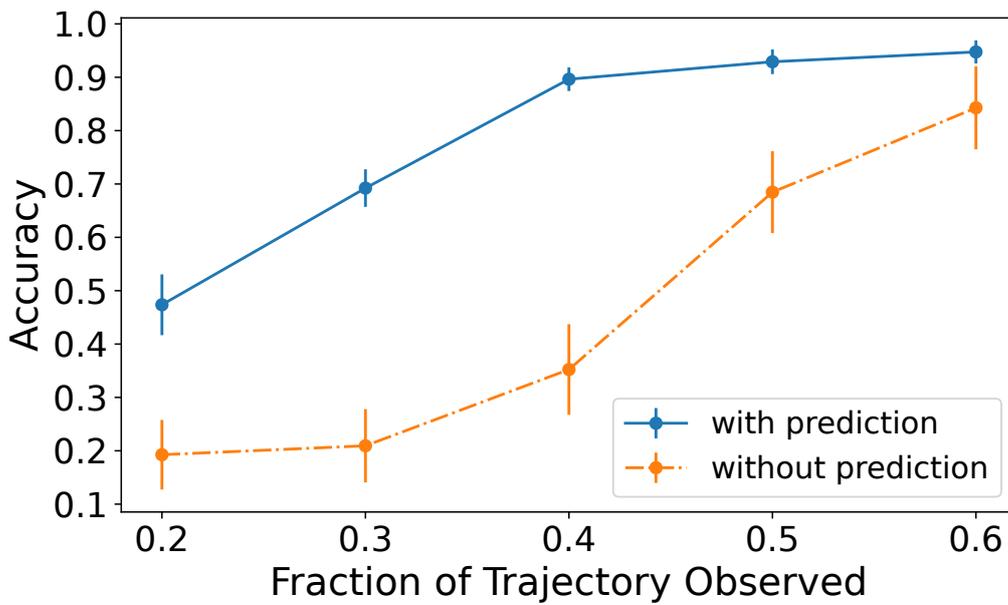
Figure 3.2 illustrates the layout of the twelve target positions. These targets are arranged in two rows with six targets per row: within each row, targets are spaced 10 cm apart, and the two rows are separated by 20 cm. The targets are numbered from the human operator’s perspective in a top-down, left-to-right order—targets 1 to 6 in the first row and targets 7 to 12 in the second row. Figure 3.7a shows the test results for trajectory prediction. The results indicate that the palm and wrist exhibit the largest prediction errors, likely due to the higher variability in their motions. Overall, the prediction errors for all joints remain below 2 cm over 5 time steps.

For predicting the endpoint of a human reaching motion, we train twelve GMMs using Python [130] on the dataset $\mathcal{D} = \{X_{i,0:T}\}_{i=1}^N$, allocating 80% for training and 20% for testing. As noted earlier, the initial segments (approximately 40%) of the recorded human trajectories are very similar and exhibit significant overlap, as shown in Fig. 3.3b. This overlap makes early-stage trajectory classification particularly challenging, as illustrated in Fig. 3.7b. When relying solely on the observed motion data, the GMM-based classification (depicted by the orange curve) appears largely random at first, but improves once around 40% of the trajectory has been observed. However, even after 60% of the motion is known, some misclassifications still occur.

However, our LSTM network has already learned to distinguish among different reaching motions and can accurately forecast the human palm’s position for the upcoming time steps. Consequently, by providing the GMMs with a combination of predicted hand positions along with the observed data, we can substantially enhance the accuracy of reaching target classification during an ongoing motion.



(a)



(b)

Figure 3.7: Analysis of Trajectory Prediction and Target Estimation [69]. **(a)** Average prediction errors across different time horizons for the four monitored human arm and hand joints. **(b)** Comparison of early-stage target estimation accuracy, with and without incorporating short-term trajectory predictions. The findings clearly demonstrate that integrating the predicted trajectory markedly enhances early target estimation accuracy.

During training, the Seq2Seq neural network’s dataset was divided into equally sized trajectory segments, each lasting 0.7 s. The first 0.35 s of each segment (corresponding to 10 steps) served as the network’s input, while the subsequent 0.35 s (also 10 steps)

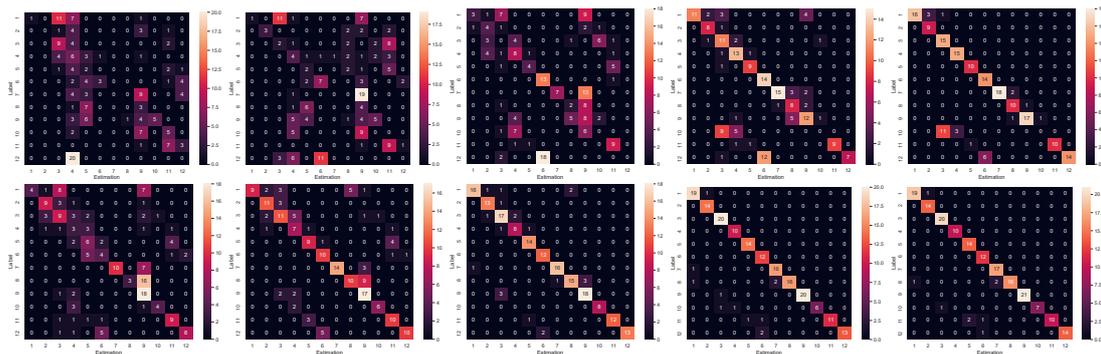


Figure 3.8: The confusion matrices in the upper row depict the intended position estimation results using solely the observed trajectories, while those in the lower row illustrate the results when both observed and predicted trajectories are employed. Additionally, each column corresponds to a different observation level—20%, 30%, 40%, 50%, and 60% of the reaching trajectory. [69]

was used as the target output. For context, one complete pick-and-place cycle consists of approximately 32 steps.

In our initial experiments, we selected an LSTM layer with a 128-dimensional hidden state. The Seq2Seq network was implemented in PyTorch [93] and trained using a batch size of 128, with a teacher forcing ratio of 0.6. The training started with a learning rate of 0.005, which decayed exponentially at a rate of 0.01. To further accelerate training, we applied batch normalization. Finally, we determined the loss function weights through grid search, resulting in hyperparameter values of 0.08, 0.16, 0.32, and 0.44.

Figure 3.7b (blue curve) shows how classification accuracy improves as a larger portion of the trajectory is observed. To further analyze the algorithm’s performance, we present the corresponding confusion matrices in Fig. 3.8. These diagrams compare the classification results obtained without and with LSTM predictions (upper and lower rows, respectively) at different observation levels—20%, 30%, 40%, 50%, and 60% of the human reaching trajectory being fed into the GMMs. In the upper row, the classification is initially nearly random and only shows significant improvement after 50% and 60% of the human motion is observed. Notably, misclassifications between targets 3 and 10, as well as between targets 6 and 12, persist; these errors can be attributed to the experimental setup, where the human hand passes over targets 3 and 6 to reach targets 10 and 12. With trajectory prediction enabled, however, the target estimates become considerably more accurate and robust once at least 30% of the human motion has been observed.

3.3.2 Online Trajectory Generation

The proposed online trajectory generation method is evaluated on a 6-DoF UR10e manipulator arm outfitted with a Shadow C6 hand, as depicted in Fig. 3.1. To ensure human safety, the arm’s maximum joint velocity is capped at 0.02 rad s^{-1} and its acceleration is limited to 1 rad s^{-2} (in accordance with the previously defined constraints). For collision avoidance, both the robot and human elements are modeled using capsule geometries

with a fixed radius of 10cm, and the collision margin d between capsules is set to zero.

The trajectory optimizer is implemented in C++ utilizing the Eigen library for linear algebra operations. All modules communicate via the ROS platform [103]. The robot model is loaded through MoveIt [25], and real-time commands are sent using Roscontrol [22] together with the `ur_modern_driver` [4]. In our system, predictions from the LSTM and GMM modules are computed at approximately 20 Hz, while the trajectory optimizer runs at 10 Hz.

The experiment is divided into two distinct phases: a reaching phase and a staying phase. During the reaching phase, the human begins by picking up a screw bolt from the initial position and placing it at target 3. As soon as the human departs from the starting location, the robot commences its movement from target 7 toward target 6. Once the bolt is placed, the human remains at target 3 for 5 seconds (the staying phase), while the robot continues to execute its task. Finally, both the human and the robot return to their respective starting positions.

To compare the performance of two control strategies—a reactive controller that considers only the current human arm-hand (AH) positions during trajectory generation, and a predictive controller that incorporates both current and forecasted AH positions—we repeated the experiment multiple times, as illustrated in Fig. 3.9 and Fig. 3.10. In this scenario, the predictive controller leverages five predicted palm positions to enhance target estimation at the early stage of the reaching motion.

Figure 3.9 displays the outcomes for the reactive controller. The first three columns show the human reaching phase, where the robot begins moving toward its target without any motion prediction, inadvertently steering its trajectory closer to the human arm. When the distance between the human arm and the robot falls below a predetermined threshold (as shown in the third column), the robot adjusts its motion to avoid a collision before proceeding to target 6. Although the resulting path is collision-free, it is far from optimal.

In contrast, Fig. 3.10 demonstrates the performance of the predictive controller. Here, the robot anticipates the human’s entry into the shared workspace much earlier, allowing it to plan a trajectory toward target 6 that avoids approaching the human arm too closely. Notably, the minimum separation between the human arm and the robot is greater than that observed with the reactive controller, indicating that the predictive approach offers improved safety.

Figure 3.11 offers a side-by-side comparison of the robot trajectories produced by both controllers over three successive phases—reaching, staying, and returning—in a single diagram. During the reaching phase, the reactive controller takes a pronounced detour, whereas the predictive controller, despite some initial variability, quickly converges to a smoother path as the target prediction stabilizes. Both controllers achieve nearly ideal, straight-line trajectories during the return phase.

We also evaluated several key performance metrics, including the minimum distance between the human palm and the robot’s index fingertip, the overall length of the robot trajectories, and the execution time (see TABLE 3.1). As expected, the predictive controller generates shorter trajectories and maintains a larger minimum separation between the human and the robot compared to the reactive controller. These results indicate that the predictive controller enhances both the efficiency and safety of the system.

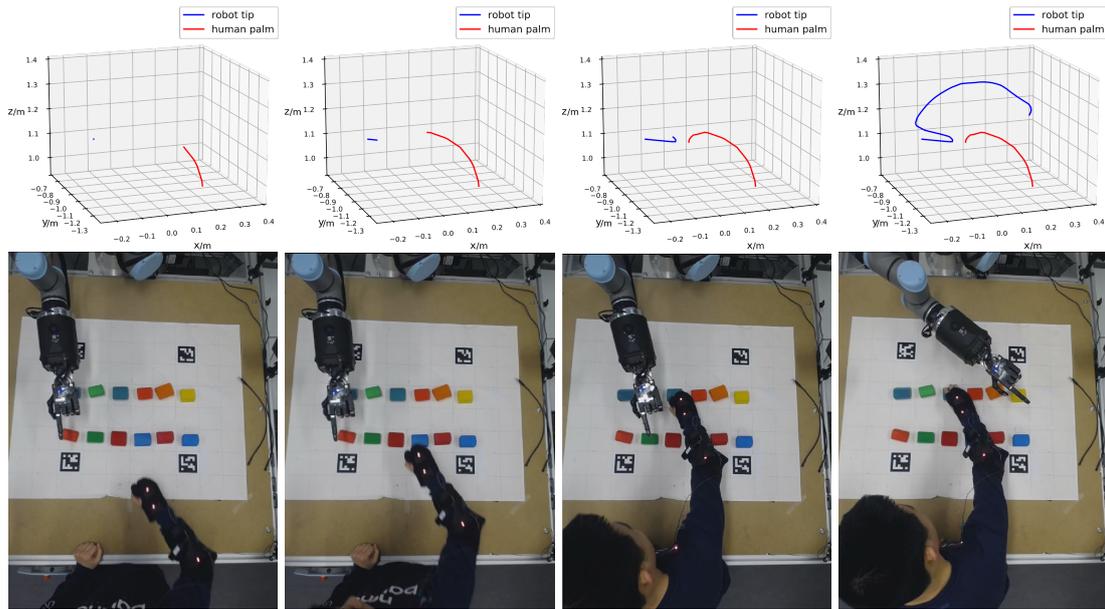


Figure 3.9: Experiments with a reactive controller (no prediction). The upper row of figures shows the Cartesian trajectories of both the human and the robot: the blue line depicts the robot’s path—specifically, the tip of the first finger of the Shadow hand moving from target 7 to target 6—while the red line represents the human palm joint trajectory reaching toward target 3. The lower row of photos captures the human arm’s movement during the reaching phase, with the final image showing the human arm holding position at the target for 5 seconds. [69]

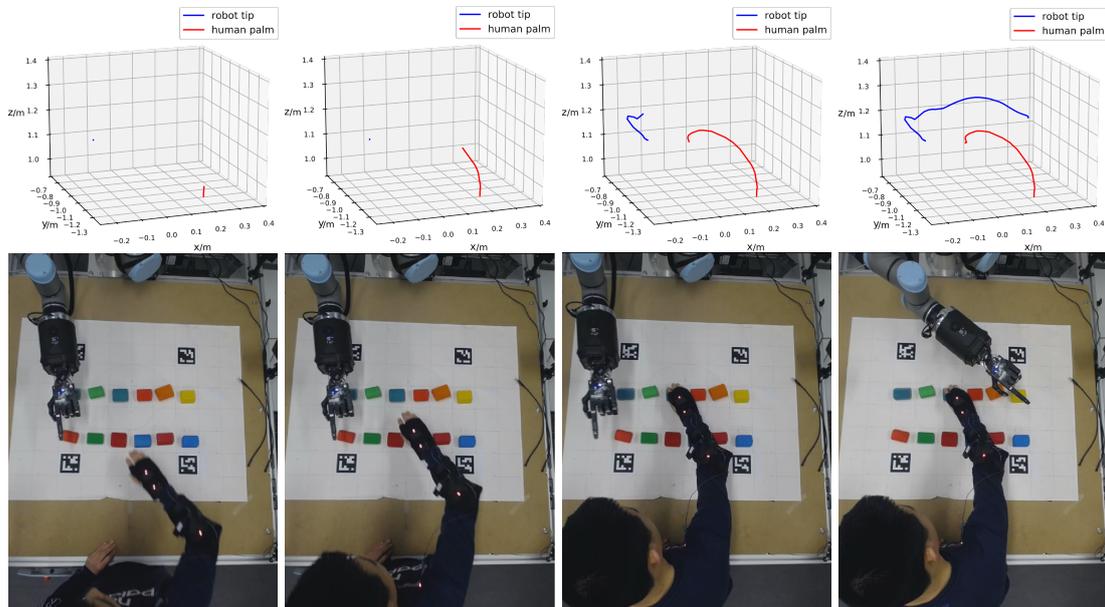


Figure 3.10: Experiments with our predictive controller are conducted under the same conditions as those shown in Fig. 3.9: the human motions, initial and target poses, and the robot’s dynamic constraints remain unchanged. [69]

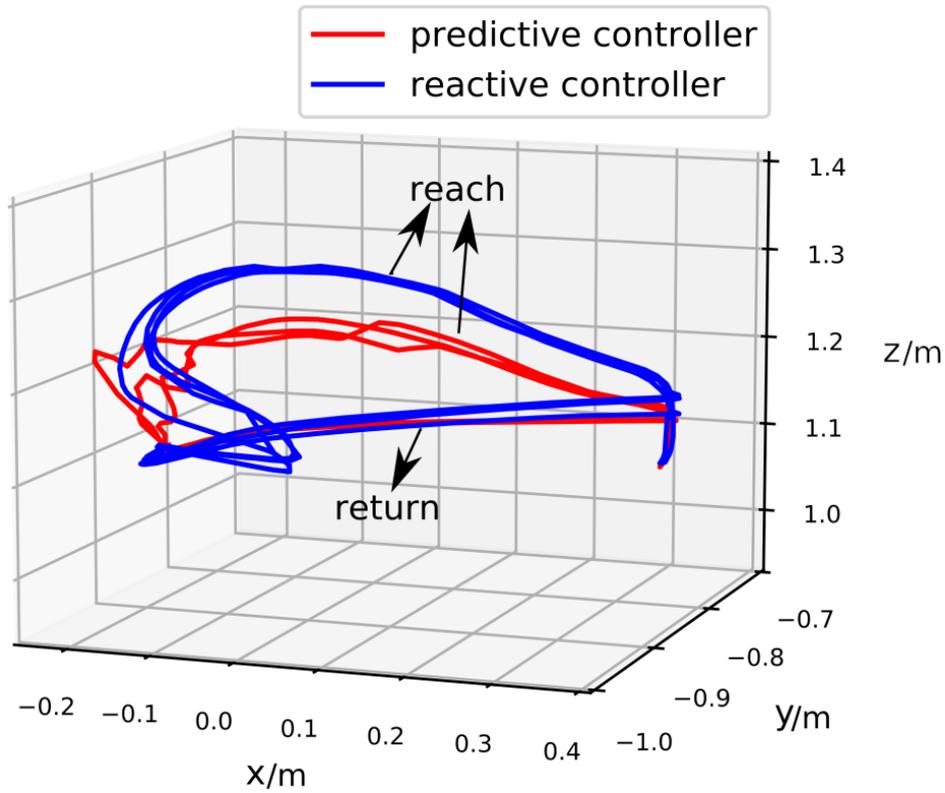


Figure 3.11: The trajectories of the robot’s Shadowhand first fingertip, recorded during experiments with both reactive and predictive trajectory controllers, are presented. [69]

3.3.3 HRC Efficiency

To assess the overall efficiency of our algorithm, we conducted experiments in which the participant performed the collaborative task under various scenarios. In these experiments, the participant retrieved an object from an initial position and placed it at any of the 12 target positions in a random order (e.g., $7 \rightarrow 12 \rightarrow 5 \rightarrow 11 \rightarrow 3 \rightarrow 9 \rightarrow 2 \rightarrow 8 \rightarrow 10 \rightarrow 4 \rightarrow 1$

Table 3.1: Comparison Results Between Predictive and Reactive Controller [69]

Trajectory Controller	Trial	Minimum Distance (cm)	Trajectory Length (cm)	Execution Time(s)
Predictive Controller (ours)	1	25.40	75.07	3.88
	2	26.58	86.73	3.89
	3	25.33	99.25	3.89
	Average	25.77	87.02	3.89
Reactive Controller (baseline)	1	18.31	138.72	4.64
	2	14.67	141.24	4.71
	3	16.88	144.15	4.95
	Average	16.62	141.37	4.77

→6→7). For example, one task flow involved the participant placing a screw bolt at target 3 and then waiting for 5 seconds. During this waiting period, the robot had to change its target from target 3 to target 5 and re-plan its trajectory—based on the predicted human AH trajectory shown in Fig. 3.12a—to avoid disrupting the human’s work.

The relationship between the human’s intended target and the robot’s available target positions is summarized in TABLE 3.2. The design of this workflow aims to enable the human and the robot to work concurrently without interference. Coordination is managed by the robot goal scheduler module, as illustrated in Fig. 3.4. For instance, once the robot has reached target 6, it continues to move through the targets in the order 7→8→9→10→11→12→1 if no human movement is detected in the workspace; otherwise, the robot’s target is updated online according to the schedule in TABLE 3.2.

We performed this experiment under two different conditions as part of an ablation study to further investigate the system’s efficiency.

In Situation 1, human target positions were predicted without incorporating any arm-hand (AH) motion prediction (denoted as NP), and the robot’s trajectory was replanned using the predictive controller. In Situation 2, human intentions were estimated by incorporating predicted AH trajectories (denoted as WP), and the robot’s trajectory was similarly replanned using the predictive controller. Throughout the experiment, we recorded the number of assembled products produced by the robot. As illustrated in Fig. 3.12b, our algorithmic strategy yielded a reliable improvement in HRC efficiency. Although the overall improvement was modest, this can be attributed to the fact that many human-robot target combinations were inherently conflict-free (for example, when the human’s intended target was 7 while the robot was moving from target 3 to target 4). Excluding these conflict-free cases would likely result in a more significant efficiency improvement

Table 3.2: Experiment workflow

Human Intended Target	Robot available Targets
1	3, 4, 5, 6, 10, 11, 12
2	4, 5, 6, 7, 11, 12
3	1, 5, 6, 7, 12
4	1, 2, 6, 7, 8, 12
5	1, 2, 3, 7, 8, 9
6	1, 2, 3, 4, 7, 8, 9, 10
7	1, 2, 3, 4, 5, 6, 9, 10, 11, 12
8	1, 2, 3, 4, 5, 6, 10, 11, 12
9	1, 2, 3, 4, 5, 6, 7, 11, 12
10	1, 2, 3, 4, 5, 6, 7, 8, 12
11	1, 2, 3, 4, 5, 6, 7, 8, 9
12	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

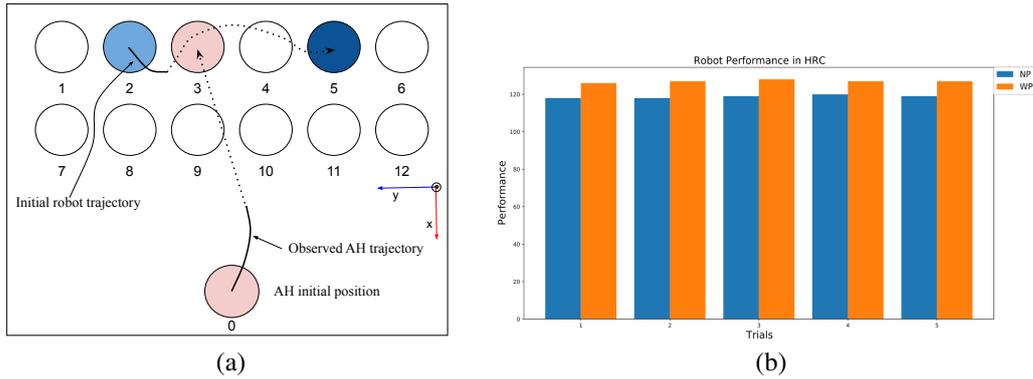


Figure 3.12: (a) Initially, the robot is set to proceed to target 3. However, once the system’s human intention and arm-hand (AH) trajectory prediction—based on the observed data—confirms that the human is indeed aiming for target 3, the robot re-plans its trajectory in real time (in this instance, redirecting to target 5 instead of target 3). This adjustment ensures that the robot maintains a safe distance and does not disrupt the human’s task. (b) This panel displays the number of products assembled by the robot under two conditions. NP: Human intention classification performed without incorporating predicted arm trajectories; WP: Human intention classification enhanced with predicted arm trajectories. [69]

between WP and NP.

3.4 Summary

We introduce a comprehensive pipeline aimed at enhancing both the efficiency and safety of HRC assembly tasks. Our approach begins with training a Seq2Seq neural network to accurately predict the human AH trajectory. Unlike conventional methods, we feed both the observed and predicted trajectory segments into GMMs for target estimation. Our experiments show that this strategy produces a significantly more accurate posterior probability distribution over potential target positions, even when the initial portions of the trajectories are very similar.

Next, we combine the predicted trajectory with the estimated target position to generate a goal-oriented, collision-free trajectory using a novel trajectory generation method. Evaluations on our real robot system confirm that the overall pipeline markedly improves the safety and efficiency of the HRC task.

Chapter 4

Predictive Planning for HRC Using EEG

This chapter investigates the use of predictive planning augmented by BCI technology to improve the HRC performance. By leveraging a novel BCI paradigm that detects a human's overt attention, the robot is able to anticipate upcoming actions and proactively adjust its behavior. This approach contrasts with traditional methods that rely solely on reactive responses to human postures, by incorporating early intention prediction to enhance task coordination. Furthermore, the system utilizes the signal-to-noise ratio (SNR) of brain responses to dynamically modulate the robot's velocity in accordance with the human's level of vigilance. The proposed system is validated through physical robot experiments, which demonstrate its capacity to improve collaboration efficiency while maintaining safe working distances.

4.1 Introduction

While the previous chapter provided an overview of human–robot collaboration in industrial production settings and discussed the inherent challenges of balancing performance with safety, this chapter delves into a specialized approach for enhancing HRC through predictive planning using BCI technology. Here, our focus is on leveraging early intention prediction to enable robots to anticipate human actions and adjust their behavior based on human vigilance states.

Multidisciplinary research can open up new communication channels between humans and robots, potentially enhancing HRC performance [51]. In line with this idea, we explore the integration of BCIs with online trajectory optimization to achieve safer and more efficient HRC. BCIs have been widely applied in robot control, ranging from predicting human intentions for collaborative tasks to providing high-level commands in teleoperation. For example, the readiness potential—an EEG signal deflection occurring about 500 ms before movement onset—can indicate which arm a person will move, allowing the robot to adjust its speed based on human presence in the workspace [11, 28]. Moreover, movement-related cortical potentials recorded around the time of movement onset carry valuable information about the upcoming grasp type (such as palmar or

pinch) [141]. Motor imagery is another effective BCI paradigm, where imagined movements of the hands, feet, or tongue produce distinct EEG patterns that deep neural networks can classify with impressive accuracy [41, 146]. Additionally, BCIs leveraging the P300 component have been successfully used to control assistive robot arms and mobile robots, demonstrating the potential to encode a broader range of intentions through interfaces like BCI spelling applications [119].

While previous studies have explored BCI applications in HRC, our approach introduces two key innovations that differentiate it from existing work. First, rather than relying on traditional P300 or SSVEP paradigms—which often require the user to switch gaze between a dedicated interface and the workspace—we employ a spatially-coded SSVEP BCI that detects gaze direction relative to a single flicker stimulus. This design allows the BCI interface to be directly projected onto the workspace, enabling seamless and intuitive intention prediction as the operator naturally focuses on the target area.

Second, in addition to capturing overt attention, our system continuously monitors the operator’s vigilance by analyzing the signal-to-noise ratio (SNR) of the SSVEP response. By treating vigilance as a dynamic parameter that can fluctuate over sub-minute intervals, we use the SNR as an index to adapt the robot’s velocity in real time. This dual strategy of early intention prediction and dynamic speed modulation is then integrated with an online trajectory optimizer and real-time human arm tracking, ensuring that the robot adjusts its workspace target and operating behavior in response to both predicted human actions and current attention levels.

We validated this integrated system on seventeen participants under three conditions—using only arm tracking, using BCI-based target prediction prior to movement onset, and combining BCI target prediction with vigilance-based velocity modulation. Our results demonstrate that this novel integration of spatially-coded SSVEP BCI and vigilance monitoring can significantly enhance both the efficiency and safety of HRC.

4.2 Methodology

4.2.1 Setup and task description

This study focuses on non-physical collaboration scenarios, where a robot and a human work in the same workspace but at distinct locations, ensuring they never occupy the same work area simultaneously. Examples of such tasks include component allotment [58, 92, 95, 128], collaborative assembly [20, 152], and daily food preparation [128]. In our work, we investigate a collaborative screw assembly scenario involving five products that each require screwing at different locations. In this setup, the human places screws into the products, while the robot retrieves screws from a different product to avoid interference. For simplicity, we simulate this scenario using stand-in touch gestures and define five target locations—indicated by colored blocks—positioned between the human and the robot.

Experiments were conducted using a UR10e robot arm equipped with a Shadow C6 dexterous left hand. The architecture of the proposed HRC framework is illustrated in Fig. 4.1.

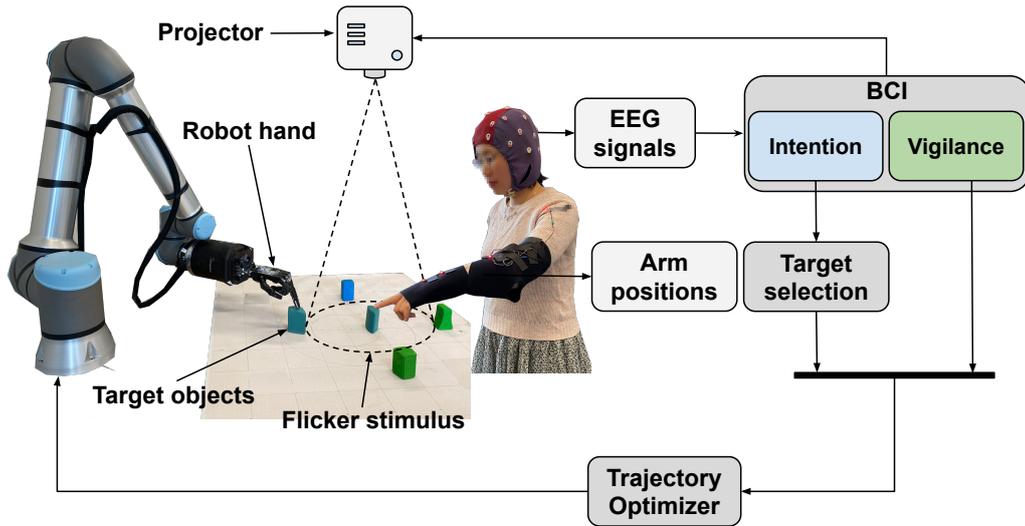


Figure 4.1: The proposed HRC framework. [68]

In the experiment, both the human and the robot used their index fingertips to touch designated blocks. At each target, they hovered their hand above the block for 1.5 seconds before performing a simple finger motion to make contact with the block's top. Meanwhile, the robot had to avoid contact with the human's arm by dynamically adjusting its reach target, and its arm velocity was modulated based on the human's vigilance level.

4.2.2 Participants

Seventeen participants took part in the study, five of whom were female. Their ages ranged from 17 to 40 years (mean age: 27.82 years). All participants had normal vision and were free from any neurological or ophthalmological disorders. Except for two individuals, all reported being right-handed; the two left-handed participants indicated that their right hand would perform equally well in the task. Additionally, eight participants were recruited from outside the associated institutions and received financial compensation for their participation.

Before the experiment commenced, the study's objectives and procedures were explained in detail to all participants. Safety measures were thoroughly reviewed, including the velocity and force limits of the robot arm, as well as the operation of a foot pedal and an emergency stop button controlled by the experimenter. Participants were informed of their right to withdraw from the study at any time without needing to provide a reason. A consent form was also provided and explained in detail. The study was approved by the ethics committee of the Medical Association of the City of Hamburg (for the BCI component) and by the ethics commission of the Department of Informatics at Universität Hamburg (for the robot experiment).

4.2.3 Experimental conditions

The study comprised three experimental conditions, each preceded by an initial training session for BCI calibration. The order of these conditions was randomized and counter-balanced across participants.

During the training session, the robot hand followed a fixed sequence to approach and tip each target, as previously described. Participants were instructed to stand still in front of the table and to direct their gaze toward the target positions indicated by the BCI cues. The EEG signals recorded during this session were then used to train the BCI classifier.

In the arm-tracking (AT) condition, the online robot controller relied exclusively on arm tracking data to ensure safe operation. At each time step, the target location closest to the participant's palm was designated as the current reach goal, while the robot's velocity limits remained constant. This condition served as the baseline for evaluating the two BCI-enabled conditions.

In the BCI condition, target information was provided to the robot controller based on the participant's gaze direction before the reach movement was initiated. The instructions for the operator and the robot control method remained the same as in the AT condition.

The third condition, termed BCI+VCV, combined the BCI-based target prediction with a dynamic adjustment of the robot's velocity based on real-time estimates of the participant's vigilance (vigilance-controlled velocity (VCV)). The underlying hypothesis was that a more vigilant participant could safely interact with a faster-moving robot. Therefore, the estimated vigilance level was used to modify the robot's joint velocity limits within a predefined margin.

Between sessions, participants were encouraged to enhance BCI classification accuracy by concentrating their visual attention on the target locations and minimizing distracting thoughts, thereby improving performance in subsequent sessions.

4.2.4 Brain-computer interface

In our study, we employed a novel spatially coded SSVEP-based BCI to determine the operator's gaze direction. Unlike traditional frequency-coded SSVEP systems that rely on multiple flicker stimuli—each with distinct frequencies or phases—to define different targets, this method uses only a single flicker stimulus and infers the target based on its position within the operator's visual field. Preliminary results indicate that this approach not only simplifies the stimulation setup but may also reduce visual strain. (A detailed description of this paradigm and its performance analysis can be found in [78]; here, we describe its adaptation for HRC.)

The underlying principle is that the brain's response to the flickering stimulus can be used to decode target locations. When the white disc flickers at 15 Hz, it induces a SSVEP in the occipital cortex, creating a standing wave pattern. The spatial distribution of this SSVEP across the scalp provides information about where the operator is looking. Because EEG topographies can vary significantly between individuals, we conduct a dedicated training session for each subject to collect personalized data for

classifier training. EEG signals were recorded using 32 electrodes arranged according to the 10-20 system and connected to an ActiveTwo amplifier (BioSemi, Amsterdam, The Netherlands). This system's active electrodes and common-mode rejection capabilities help minimize external electromagnetic interference, making it suitable for use in typical noisy environments. Aside from applying a band-pass filter, no additional artifact-removal techniques were used.

The flicker stimulus was implemented as a white disc, 40 cm in diameter, projected onto the workspace via a Toshiba TDP P9 overhead projector. Five target locations were defined on the disc—one at the center and one each at the eastern, southern, western, and northern peripheries—labeled 1 through 5 (see Fig. 4.1).

Each trial began with a cue: a small red disc appeared near one of the target blocks for 1 second, instructing the participant to fixate on that target. If the robot's hand obscured the cued area or if the cue was missed, participants were told to gaze at the target they believed had been indicated. After the cue, the disc flickered for 2 seconds while EEG data were recorded. Once the flicker period ended, the participant reached for the cued target and tapped the top of the wooden block, completing the action within an allotted 2 seconds (including the return to a resting position). This process was repeated with a randomly selected target for each trial, with ten trials per target collected during training. Parameters such as the stimulus size, flicker frequency, number of targets, and stimulation period were fine-tuned based on pilot experiments to optimize the BCI's classification accuracy.

For each trial, canonical correlation analysis (CCA) was performed between the EEG data and sine/cosine reference signals at the flicker frequency. The resulting correlation coefficients served as features for a linear discriminant analysis (LDA) classifier, which was trained on the data from the training session. During the experimental phase, EEG data were classified online after each trial, and the classifier output—representing the target number that the participant was gazing at—was transmitted to the robot controller.

4.2.5 Robot trajectory generation

Target selection

The robot's target location was selected online based on the predicted reach goal of the human. If the operator began moving during the robot's reaching phase, the current target was discarded and a smooth trajectory was re-planned, as described below. To simplify target selection, we manually encoded the geometric dependencies between targets using a set of conditions illustrated in Fig. 4.2. These rules ensure that the robot does not attempt to reach a target that is physically blocked by the human arm during the reaching motion. In practice, the robot still proceeds through the targets sequentially but skips those currently occupied according to the human motion prediction. Moreover, even if an invalid target is chosen due to an erroneous prediction, the trajectory optimization—which relies on real-time arm tracking data—prevents collisions by keeping the robot in a local minimum until the target is either unblocked or updated.

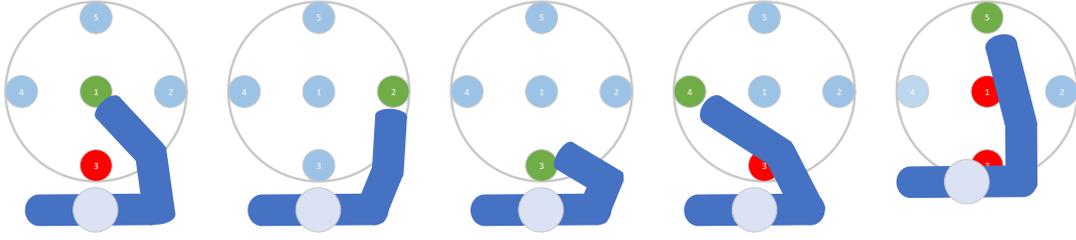


Figure 4.2: Relationship between valid robot target locations (light blue) when the human’s intended target is indicated in green. Targets marked in red are obstructed by the human arm. [68]

Trajectory optimization algorithm

Based on the goal positions and human arm pose data, our trajectory optimization algorithm—implemented in a MPC framework—generates collision-free, goal-directed trajectories. In MPC, control variables are adjusted to minimize a predefined objective function while satisfying a set of constraints [84]. In our formulation, the term l_1 represents the pose loss of the robot’s first fingertip (FFT) and is used to drive its pose (x_k, rot_k) toward the desired pose (x_{des}, rot_{des}) in Cartesian space. The desired orientation rot_{des} is specifically chosen to prevent the robot from blocking other target objects on the table from the operator’s viewpoint. Additionally, a constant c is incorporated to promote smoother and more consistent trajectories, and N denotes the total number of waypoints in the robot’s trajectory.

$$l_1 = \sum_{k=1}^N \min(c, ||x_k - x_{des}||) + ||rot_k - rot_{des}|| \quad (4.1)$$

The generated trajectories should satisfy the robot’s kino-dynamic constraints.

$$\underline{q} \leq q_k \leq \bar{q} \quad (4.2)$$

$$\underline{\dot{q}} \leq \dot{q}_k \leq \bar{\dot{q}} \quad (4.3)$$

$$\underline{\ddot{q}} \leq \ddot{q}_k \leq \bar{\ddot{q}} \quad (4.4)$$

In the BCI+VCV condition, we modulated the robot’s velocity limits according to the operator’s vigilance level using the following formula:

$$-\underline{\dot{q}} = \bar{\dot{q}} = (1 + 0.3 \times \text{vigilance}) \times V_{\max},$$

where the vigilance values are clipped to the interval $[-1, 1]$. This results in speed limits ranging from $0.7 \times V_{\max}$ to $1.3 \times V_{\max}$. In contrast, for both the BCI and AT conditions, the velocity limit was fixed at $0.7 \times V_{\max}$. Consequently, the robot arm generally operated at a higher speed in the BCI+VCV condition than in the other two conditions.

To ensure smooth motion, both the velocity and acceleration were regulated, with c_1 and c_2 serving as weights in the overall loss function.

$$l_2 = \sum_{k=1}^N c_1 \dot{q}_k^2 + c_2 \ddot{q}_k^2 \quad (4.5)$$

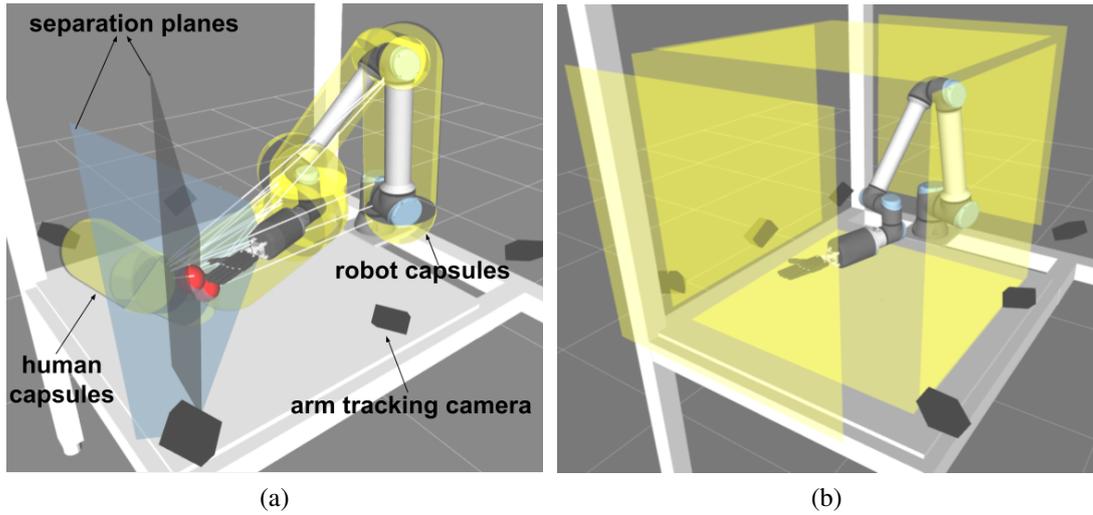


Figure 4.3: Geometric Model of the Work Cell. In this view, the human stands to the left of the table, with only the right arm reaching into the shared workspace. In (a), yellow capsules represent the geometric primitives assigned to various parts of the robot and the human arm (including the palm, wrist, lower arm, and upper arm). Trajectory planning and collision avoidance rely on efficiently computed pairwise distances between these capsules. Additionally, two blue separating planes are shown; their origins, marked in red, are positioned at the midpoints between the closest capsule pairs (specifically, between the human hand and palm versus the robot palm). In (b), the static boundaries of the robot workspace are defined by a cuboid formed by six planes, ensuring that no part of the robot extends outside the designated area. [68]

To prevent collisions between the robot’s and the operator’s arms, we represent both arms using capsule-shaped collision objects. For the operator’s arm, capsules with radii $\{r_{a,i}\}_{i=1}^4$ are constructed between each pair of connected joints, while for the robot arm, capsules with radii $\{r_{r,j}\}_{j=1}^6$ are generated between all adjacent joints. Next, we calculate the pairwise closest points between each operator capsule and each robot capsule. The line segment connecting these closest points defines the normal vector $N_{i,j}$ for the separation plane between capsule i and capsule j (see Fig. 4.3). The Cartesian positions of the operator’s arm joints and the robot’s joints are denoted by $\{P_{a,i}\}_{i=1}^4$ and $\{P_{r,j}\}_{j=1}^6$, respectively. Soft constraints based on these geometric relationships are then applied to avoid collisions between the robot’s and the operator’s arms.

$$l_3 = \sum_{i=1}^4 \sum_{j=1}^6 \min(0, (N_{i,j}^T (P_{a,i} - P_{r,j}) - r_{a,i} - r_{r,j} - d)^2) \quad (4.6)$$

Table 4.1: Trajectory optimizer parameters.

Parameter	Radii of Capsule	v_{max}	a_{max}	N	c_1, c_2	d	c
Value	0.1 m	0.02 rad/s	1 rad/s ²	3	1	0 m	0.2 m

Six boundary planes, denoted as $(N_{b,m})_{m=1}^6$, were employed as hard constraints to

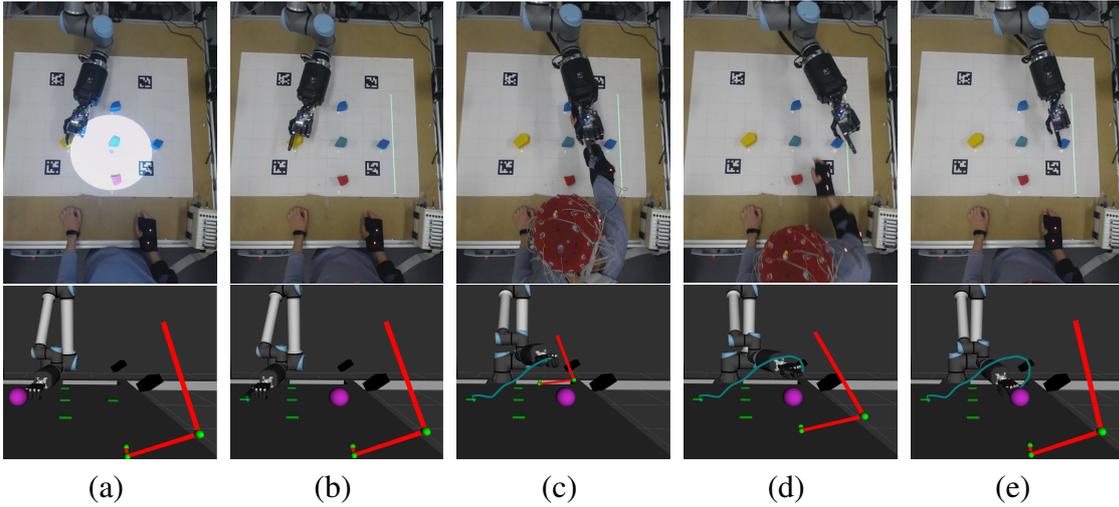


Figure 4.4: Example scenario for the robot arm trajectory generation based on the operator’s movement prediction and arm tracking. The purple sphere is the goal position of the robot’s first finger tip. Red cylinders visualize the current position of the operator’s arm. The green curve represents the trajectory of the first finger tip. **(a)** The operator gazed at target object no. 5, and the robot touched object 4. EEG signals were collected in this state (white circle shows flicker stimulus). **(b)** The BCI predicted the operator will target object 5, and the robot controller adjusted its next goal from target object 5 to target object 2 after touching target object 4. **(c-e)** The operator touched target object 5, then returned to the rest position. A collision-free robot trajectory was generated by the proposed trajectory optimization method. [68]

confine the robot’s motion to the designated workspace, as illustrated in Fig. 4.3.

$$P_{r,k}^T N_{b,m} \leq 0, \forall k \in [1, N], m \in [1, 6]$$

During the experiments, the trajectory optimizer was configured with the parameters listed in Table 4.1. The robot’s trajectory was re-optimized at 10 Hz using the most recent operator intention predictions and vigilance estimates. A primal-dual interior-point method, as described in [113], was employed to solve the optimization problem. Figure 4.4 illustrates a sample trajectory generated during the experiment.

Minimum distance analysis

In the experiment, data capturing the operator’s arm movements, the robot’s joint trajectories, and the cue signals were recorded synchronously. The recorded data were then segmented into individual trials, and for each trial, the minimum distance (MD) between the human arm and the robot limbs was calculated following the method described in the trajectory optimization section. These MD values were subsequently used to assess the safety of the human–robot collaboration.

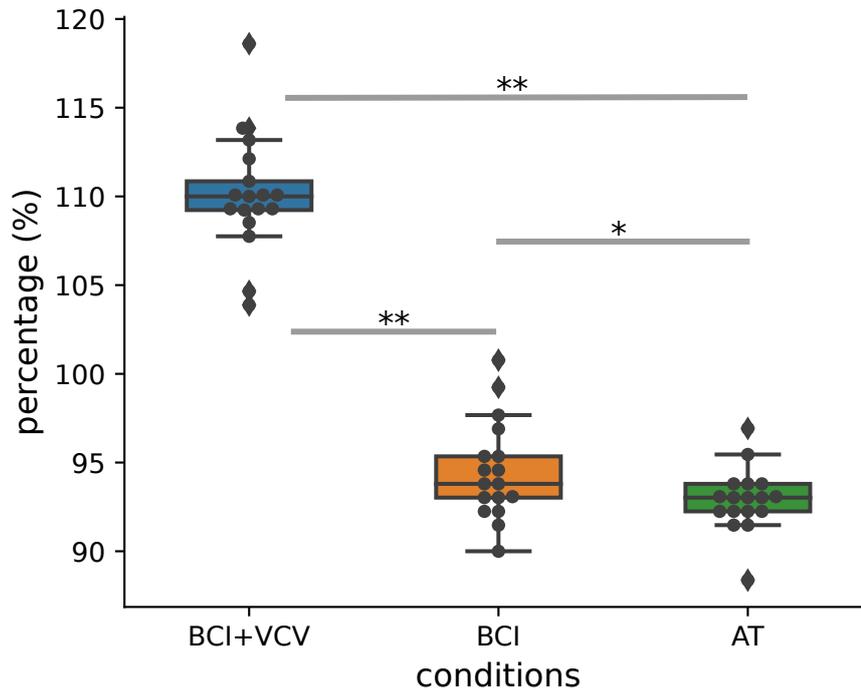


Figure 4.5: Robot performance relative to the training session. The horizontal line within each box indicates the median performance, while the upper and lower boundaries represent the upper and lower quartiles, respectively. The whiskers extend to the most extreme data points within 1.5 times the interquartile range from the quartiles. Statistical significance is indicated by $*p \leq 0.05$ and $**p \leq 0.001$. [68]

4.3 Experiments

4.3.1 HRC performance

The primary measure of robot performance was the number of targets touched during a session. We used the performance from the training session—where the operator did not interfere with the robot—as our baseline, and compared it with the performances observed in the BCI+VCV, BCI, and AT sessions. Our analysis revealed mean performance levels of approximately 110.28%, 95.04%, and 92.76% for these conditions, respectively (see Fig. 4.5). Paired t-tests confirmed significant differences between the conditions (BCI+VCV vs. BCI: $p = 1.67 \times 10^{-13}$; BCI+VCV vs. AT: $p = 1.95 \times 10^{-11}$; BCI vs. AT: $p = 0.039$), with normality verified via Lilliefors’ test [27]. These results indicate that while accounting for the operator’s actions during robot planning tends to lower performance, this adverse effect can be offset by modulating the robot’s velocity according to the operator’s attentional state.

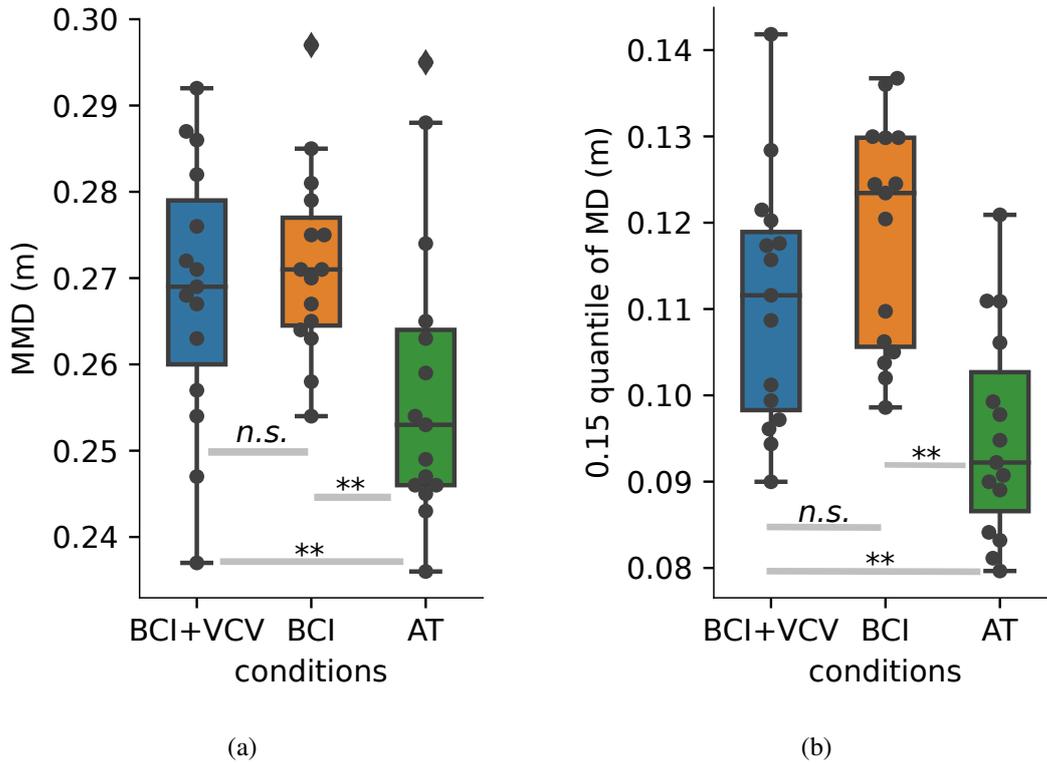


Figure 4.6: (a) Distribution of the MMD for each participant across sessions, along with the results of t-tests for statistical significance. (b) Distribution of the 0.15 quantile of the MD for each participant across sessions, accompanied by t-test results. * indicates $p < 0.05$, ** indicates $p < 0.01$, and "n.s." denotes non-significant differences where $p > 0.05$. [68]

4.3.2 Safety distance analysis

In HRC scenarios, robot performance is not the only critical metric; the safety of the interaction—assessed by maintaining a minimum distance between the robot’s and the operator’s arms—must also be ensured.

Figure 4.6a shows the distribution of the mean minimum distances (MMDs) across participants. The results indicate that the BCI condition yielded the largest safety distances, while the BCI+VCV condition, despite having a higher robot arm velocity, produced safety distances that were only marginally lower ($p > 0.49$). In contrast, the AT condition resulted in substantially smaller MMDs (both comparisons yielding $p < 0.01$).

To further evaluate whether the BCI+VCV or BCI methods improve HRC safety, we computed the 0.15 quantile of the MMD for each participant in each session. As depicted in Fig. 4.6b, the 0.15 quantile in the BCI condition was slightly higher than that in the BCI+VCV condition ($p = 0.058$) and markedly higher than in the AT condition ($p < 0.01$).

The trajectory optimization algorithm was designed to maintain a minimum distance of 20 cm between the operator’s and the robot’s arms. Nevertheless, individual samples (see Fig. 4.6) reveal that this threshold was sometimes breached. To determine whether

these sub-threshold distances were due to rapid operator movements or limitations in the trajectory optimization algorithm, we randomly selected five participants for a detailed analysis. We focused on time steps where the minimum distance d_t was less than 20 cm, and computed d_t , along with the velocities (v_{Pt} for the operator and v_{Rt} for the robot) and accelerations (a_{Pt} and a_{Rt}) at the closest points between the operator and the robot at each time step t .

All time steps were categorized into three groups: 1. **Moving Away (MA)**: $d_{t+1} > d_t$, indicating the robot's arm is moving away from the operator's arm. 2. **Braking (BR)**: $d_{t+1} < d_t$ but with the angles between v_{Pt} and v_{Rt} , as well as between v_{Pt} and a_{Rt} , greater than 90° . 3. **Failed Avoidance (FA)**: Time steps that do not satisfy either the MA or BR criteria.

In this experiment, the robot was programmed to disregard the operator's movements while it was touching a block. However, since the operator's target could coincide with the block the robot was already interacting with, several FA cases occurred where the robot's velocity dropped to zero.

The distribution of these collision scenarios, as shown in Fig. 4.7, suggests that almost all FA instances were triggered by rapid movements of the operator. In these cases, the robot's dynamic limitations prevented it from moving quickly enough to maintain the desired safety distance.

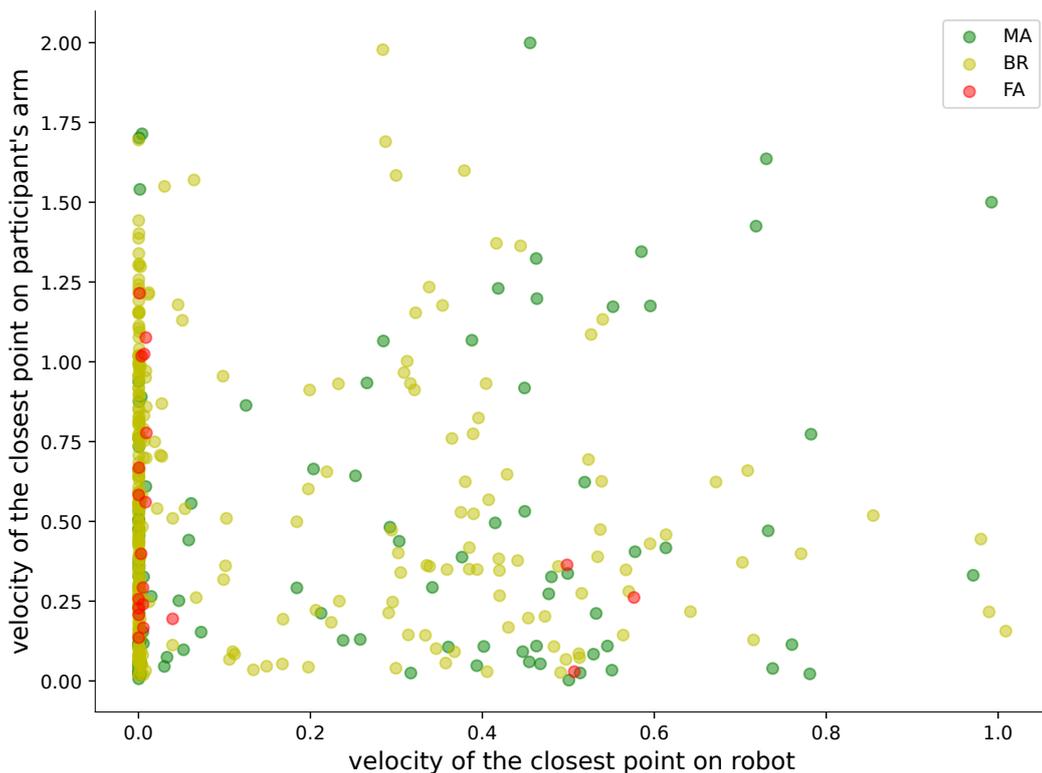


Figure 4.7: Velocities of the pairwise closest points between the operator's and the robot's arm. **MA**: robot was moving away; **BR**: robot was braking; **FA**: robot failed to avoid operator arm. [68]

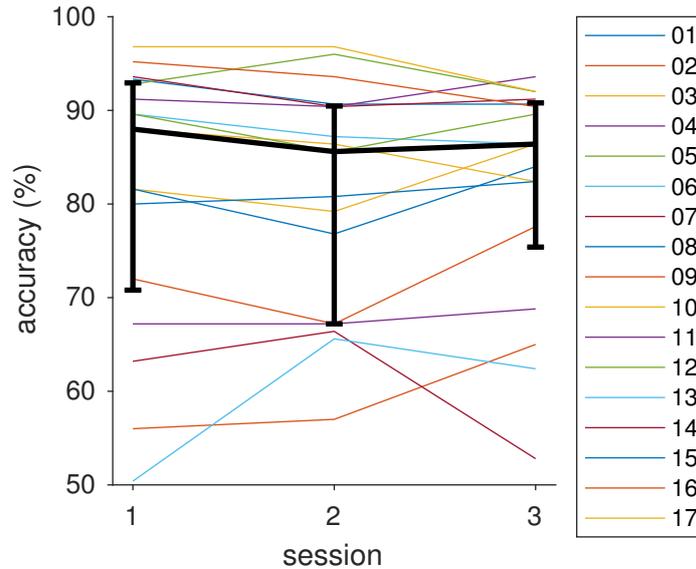


Figure 4.8: BCI classification accuracy of the participants in the three sessions. Thick black curve is the median, error bars show the 25% and 75% quantiles. [68]

4.3.3 Target prediction accuracy and vigilance

In the first session following training, the reach target prediction achieved a median accuracy of 88% across all participants. This performance remained stable in the subsequent two sessions, with median accuracies of 85.6% and 86.4%, respectively (see Fig. 4.8). Pairwise Kruskal-Wallis tests confirm that there were no significant differences between sessions (all $p > 0.93$). Although the majority of participants achieved a prediction accuracy of 80% or higher, some individuals consistently exhibited accuracies as low as 50%. This persistent underperformance across all conditions suggests that these differences are likely due to inherent individual traits rather than factors such as lapses in concentration, novelty effects, or training deficiencies.

Figure 4.9 shows an example time course of the operator’s vigilance over the three application sessions. As expected, vigilance levels varied between trials. However, distinct episodes of both higher and lower vigilance can be observed. In the example provided, vigilance gradually decreased toward the end of each session, with overall lower values in session 3. Additionally, session 2 exhibited periods of high vigilance at both the beginning and towards the end, separated by a pronounced drop in between.

Averaged across trials and participants, the vigilance values indicate that, similar to the BCI performance, participants maintained consistent attention throughout the experiment. Pairwise Kruskal-Wallis tests further confirm that there were no systematic changes in attention levels between sessions (all $p > 0.79$).

We investigated whether vigilance affects classification accuracy by analyzing the distribution of vigilance values for trials with correct and incorrect classifications. As depicted in Fig. 4.10b, the two distributions largely overlap; however, the median vigilance in trials with classification errors was slightly lower than in trials with correct classifications (-0.1165 vs. -0.1932 , $p = 0.043$, Kruskal-Wallis test). It is important

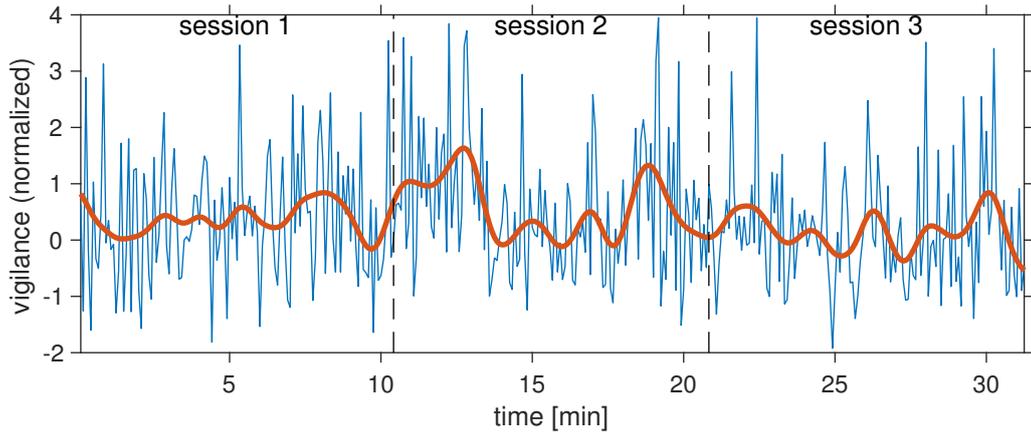


Figure 4.9: Vigilance time course for participant 08. The blue curve displays the raw vigilance values, quantified as the SSVEP signal-to-noise ratio (SNR) for each trial, while the red curve shows the corresponding low-pass filtered vigilance data (with a 0.01 Hz cutoff) to highlight slow fluctuations over time. [68]

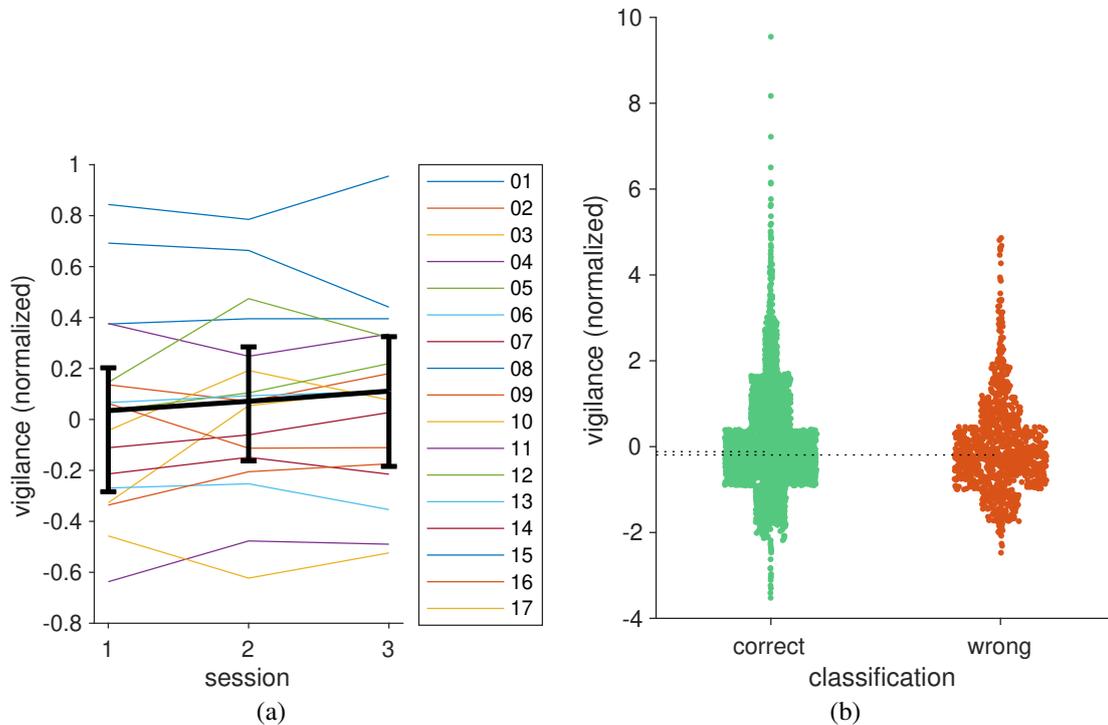


Figure 4.10: (a) Median vigilance levels for the three sessions, with error bars representing the 25th and 75th percentiles. (b) Distribution of vigilance values for trials with correct versus incorrect classifications. The width of each scatter block corresponds to the number of samples within that interval, and the dotted lines mark the medians of the two distributions. [68]

to note that factors other than vigilance—such as the robot arm casting a shadow on the flicker stimulus, obscuring target locations, or participants not fixating on the cued target—may also contribute to classification errors. These factors might explain the rel-

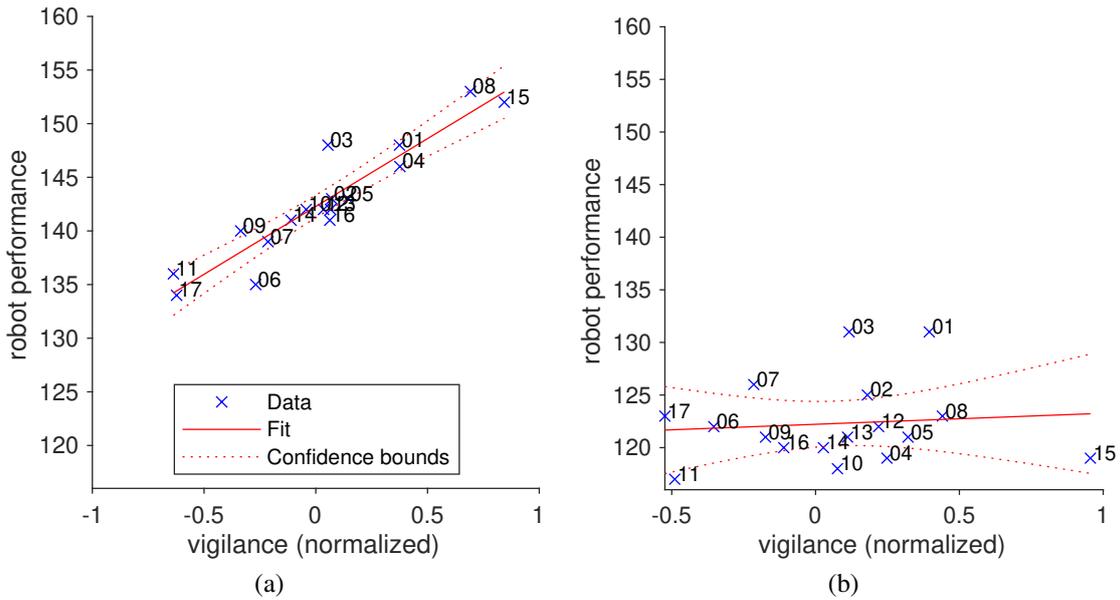


Figure 4.11: The relationship between human vigilance and robot performance was modeled using a linear fit. **(a)** Under the BCI+VCV condition, the robot’s velocity is modulated by the operator’s vigilance. **(b)** Robot velocity is not modulated by vigilance (BCI condition). Numbers at data points indicate participant number. [68]

atively high vigilance values observed in some error trials, potentially leading to an underestimation of the true median difference. Overall, although low vigilance seems to impair classification accuracy, the system remains fairly robust against occasional lapses in attention.

To verify that vigilance was the key factor driving robot performance in the BCI+VCV condition, we examined the relationship between each participant’s average normalized vigilance and their corresponding robot performance during that session. As shown in Fig. 4.11, a linear model provided an excellent fit for this relationship ($p = 5.74 \times 10^{-8}$). In contrast, no similar correlation was observed between BCI classification accuracy and robot performance. This finding supports the notion that target prediction and vigilance function as independent information channels provided by the BCI.

4.4 Discussion

In modern industrial environments, HRC is increasingly recognized as a vital approach to harness the complementary strengths of robots and humans [15]. Although operating robots at high speeds could enhance efficiency, doing so typically compromises safety. One promising strategy to address this challenge is to reliably predict human intentions. Many existing methods rely on tracking the motion of the entire human body—or specific parts thereof—to forecast the intended action [20, 28, 67, 92, 95, 107, 127]. However, a common limitation of these approaches is that they can only produce accurate predictions after the human has already initiated movement. This is particularly problematic

when potential targets are located close together, as the early segments of the hand's trajectory tend to be very similar.

To overcome this limitation, recent studies have investigated EEG-based techniques for earlier intention prediction [116]. For instance, the Bereitschaftspotential—a distinct deflection in the EEG signal that occurs approximately 500 ms before movement onset—has been used to predict not only when a human will start moving their arms but also the type of grasp that will be executed [11, 141].

Moreover, because hand and eye movements are closely coordinated during task execution, analyzing gaze direction offers another avenue for early prediction. Rather than relying on conventional optical eye-trackers, our approach uses a BCI to capture both the operator's gaze direction and aspects of their cognitive state. While previous work on BCI for robotics has primarily focused on developing the BCI component itself or testing it in simulation [11], our study advances the field by demonstrating an integrated, closed-loop system for BCI-controlled HRC.

4.4.1 BCI can improve HRC efficiency and enhance safety

We evaluated our approach in a study where participants performed a pick-and-place task in a narrow shared workspace alongside a robot arm. Our system continuously monitored participants' vigilance via EEG signals and adjusted the robot's velocity accordingly—speeding up when vigilance was high and slowing down when it was low. The results indicate that using BCI-based intention prediction, particularly when combined with vigilance-modulated velocity control, significantly enhances HRC performance and safety compared to a motion-tracking-only approach. Notably, the improved performance achieved with vigilance-controlled velocity did not compromise safety, as measured by the minimum distance between the operator's and the robot's arms.

Because the BCI enables the robot to detect the intended reach target before the human initiates movement, the robot has sufficient time to adjust its target position. This early target awareness is reflected in the performance improvements observed in the two BCI-enabled conditions compared to the arm-tracking condition, which relies solely on motion data. By dynamically adjusting the robot's velocity based on the operator's level of alertness, the overall performance was further enhanced in the BCI+VCV condition (see Fig. 4.5).

However, the BCI classifier's performance varied among participants, with some individuals achieving accuracies below 80%. This variability is a well-known phenomenon in BCI research, often referred to as "BCI illiteracy" [2]. Lower BCI performance may stem from a mismatch between the fixed stimulation parameters (e.g., flicker frequency, stimulus size, and stimulation duration) and individual response characteristics. Optimizing these parameters for each participant could improve intention prediction accuracy. Even if such optimization fails to yield the desired improvements, the system can default to using motion-tracking data for target prediction.

Another significant advantage of using BCI for HRC is the potential for enhanced safety. MPC has been employed for nearly 40 years to generate optimal robot control commands over a short future horizon by solving constrained optimization problems [84]. A key requirement for MPC is the ability to predict environmental changes

in advance. Prior studies have demonstrated that combining human motion prediction with MPC can improve HRC safety [58, 92, 152]. In our approach, we implemented an MPC-style trajectory optimization algorithm that, for simplicity, models the predicted reach trajectory as a cylinder connecting the human’s palm joint to the predicted target position. This approach accounts for the rapidity of human arm movements and the short duration of reaching motions, thereby enhancing safety. For more complex scenarios involving non-convex constraints or objective functions, learning-based methods—such as those described in [121]—could be used to further improve computation speed.

An interesting finding was that the BCI+VCV method, despite resulting in higher average robot arm velocities, maintained safety metrics comparable to the fixed-velocity BCI condition. Although faster robot movements typically increase collision risk in HRC [11], incorporating the operator’s attentional state into the velocity control strategy appears to mitigate this risk, allowing for higher speeds without reducing safety margins.

4.5 Summary

In conclusion, our study demonstrates that integrating a BCI-based intention prediction system with vigilance-controlled velocity modulation and an MPC-style trajectory optimization algorithm significantly enhances both the efficiency and safety of human–robot collaboration. By enabling the robot to anticipate the operator’s intended reach target before the movement is fully initiated and to dynamically adjust its speed based on real-time vigilance assessments, our approach allows for proactive trajectory planning that minimizes interference and maintains robust safety margins, even in tightly shared workspaces. Experimental results confirm that this integrated strategy outperforms traditional motion-tracking methods, and further improvements may be achieved through the use of advanced prediction models, such as artificial neural networks, to further refine trajectory adjustments and collision avoidance.

Chapter 5

Sim2Real Adaptive Dexterous Grasping

Following the predictive motion planning strategies presented in previous chapters, which enabled collaborative robots to anticipate human actions and adjust their trajectories accordingly, this chapter focuses on enhancing the grasping capabilities of these robots. We introduce a novel Sim2Real multimodal learning framework for adaptive dexterous grasping and grasp status prediction. Our approach employs a two-stage strategy implemented within the Isaac Gym environment, augmented by several pluggable modules, to simulate dexterous grasping using diverse multimodal sensing data—including RGB-D images, joint angles, and 3D tactile forces from soft fingertips. By generating over 500,000 synthetic grasping scenarios, we trained an adaptive dexterous grasping neural network (ADG-Net) that leverages an attention mechanism and a graph convolutional neural network module to fuse this multimodal information and learn robust grasping principles. The ADG-Net is designed to first detect feasible grasp parameters from an RGB-D image of a grasp scene and then to optimize these parameters based on additional sensory feedback when the dexterous hand contacts a target object. Extensive experiments in both simulated and real-world settings demonstrate that our ADG-Net method outperforms state-of-the-art grasping approaches, achieving average success rates of 92% for isolated unseen objects and 83% for stacked objects. This work, therefore, complements our earlier motion planning research by providing a robust grasping solution, thereby enhancing the overall manipulation ability of the cobots.

5.1 Introduction

Embodied artificial intelligence (Embodied AI) has garnered significant attention recently as it bridges the gap between computational intelligence and tangible, physical interactions. This field emphasizes the seamless integration of AI within physical agents, enabling them to perceive, interpret, and engage with complex, dynamic environments [114].

Central to the promise of Embodied AI are capabilities such as dexterous grasping

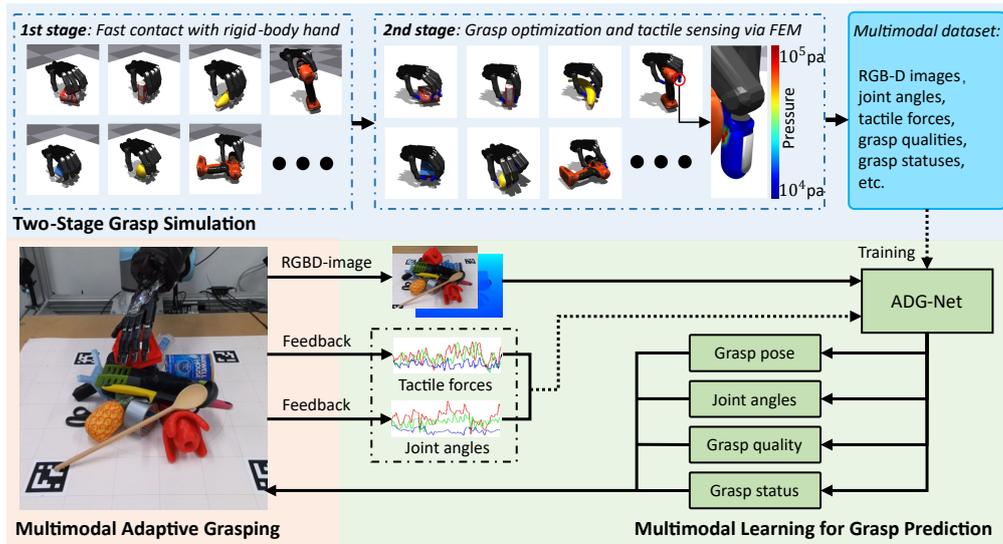


Figure 5.1: Overview of the proposed adaptive grasping method. [147]

and manipulation, which underpin its applications in sectors ranging from healthcare and manufacturing to service and domestic environments [6, 59]. However, despite notable advances, achieving robust and adaptive dexterous manipulation remains a challenging endeavor, particularly when multiple sensory modalities must be integrated for effective grasping.

Multimodal learning aims to integrate data from a range of sensory inputs—such as visual, auditory, tactile, haptic, and force-torque measurements—as well as information about an end effector’s joint states. Such comprehensive integration is essential for enabling a robot to dynamically assess grasp status and refine its grasping strategy in real time. Recent studies have demonstrated that multimodal deep neural networks are capable of learning feasible hand configurations for dexterous grasping [55, 57]. However, training these networks effectively requires access to large-scale datasets containing a vast number of multimodal grasping examples—a resource that is considerably more challenging to acquire in robotics than the abundant annotated data available for language models like ChatGPT or multimedia platforms such as Sora [91].

To address these challenges, the Sim2Real learning paradigm has emerged as a promising approach for dexterous grasp learning. In this framework, neural networks are trained in simulated environments or on synthetic datasets, with the aim of transferring the learned skills to real-world scenarios. Despite its promise, applying Sim2Real techniques to multimodal dexterous grasping remains a complex endeavor. The intricate nature of dexterous hand manipulation, combined with the need to fuse multiple sensory data streams, substantially increases the difficulty of creating accurate simulations and managing the simulation schedule. Moreover, bridging the gap between simulated and real environments requires neural networks that are both robust and adaptive [7, 101].

In this work, we introduce an advanced multimodal Sim2Real learning framework for adaptive dexterous grasping. Figure 5.1 presents a schematic overview of our framework, which is composed of three main modules: a two-stage dexterous grasp simulation

module, a multimodal learning module for grasp prediction, and a multimodal adaptive grasping module. By leveraging the flexibility of our two-stage simulation approach alongside the capabilities of ADG-Net, our framework not only facilitates the collection of rich, multimodal datasets but also supports the learning of novel grasping skills for Embodied AI robots.

5.2 Methodology

5.2.1 Motivations

The proposed multimodal Sim2Real learning framework tackles two key challenges in dexterous grasp learning. First, multimodal sensing—particularly tactile sensing with soft fingertips—entails significant computational complexity, which renders the collection of large-scale datasets both time-consuming and inefficient. To address this, we introduce a two-stage simulation method that minimizes the Sim2Real gap while efficiently generating an extensive multimodal grasp dataset. Second, it is essential to develop a neural network that effectively fuses diverse data types, such as RGB-D images, joint angles, and tactile force measurements, to learn robust grasping principles. To this end, we propose ADG-Net, a dexterous grasping method specifically designed for the multimodal adaptive grasping of ShadowHands.

5.2.2 Coordinate Systems

For dexterous grasping in both simulation and real-world settings, we establish five distinct coordinate systems: the Camera Coordinate System (CCS), World Coordinate System (WCS), Robot Coordinate System (RCS), Hand Coordinate System (HCS), and Fingertip Coordinate System (FCS). Detailed illustrations and descriptions of these coordinate systems are provided in Fig. 5.2.

5.2.3 Two-Stage Grasp Simulation

The pluggable two-stage grasp simulation framework presented in this work is designed to efficiently synthesize grasp scenarios, render comprehensive multimodal sensory data, and quantitatively evaluate grasp quality. In our approach, the multimodal data include RGB-D images of the grasping scene, joint angle configurations of the dexterous hand, and tactile force measurements at the fingertips. As illustrated in Fig. 5.1, the simulation framework operates in two stages: first, a fast contact phase during which the dexterous ShadowHand (\mathbf{H}) attempts to grasp an object (\mathbf{O}) using a pose $\mathbf{P}_{\mathbf{H}}$ in $\text{SE}(3)$ and joint angles $\theta \in \mathbb{R}^{dim}$ (with $dim = 22$ for a ShadowHand, excluding the wrist joints); and second, a grasp optimization and tactile sensing phase. During the grasp, contact forces \mathbf{f}^i and torques $\boldsymbol{\tau}^i$ are analyzed at each contact point \mathbf{p}^i , while tactile forces \mathbf{f}_{tac}^k (for $k = 1, 2, \dots, 5$) are computed in the FCS using finite element methods. Concurrently, an RGB-D image $I_{\mathbf{C}}$ is rendered in the CCS to document each grasping

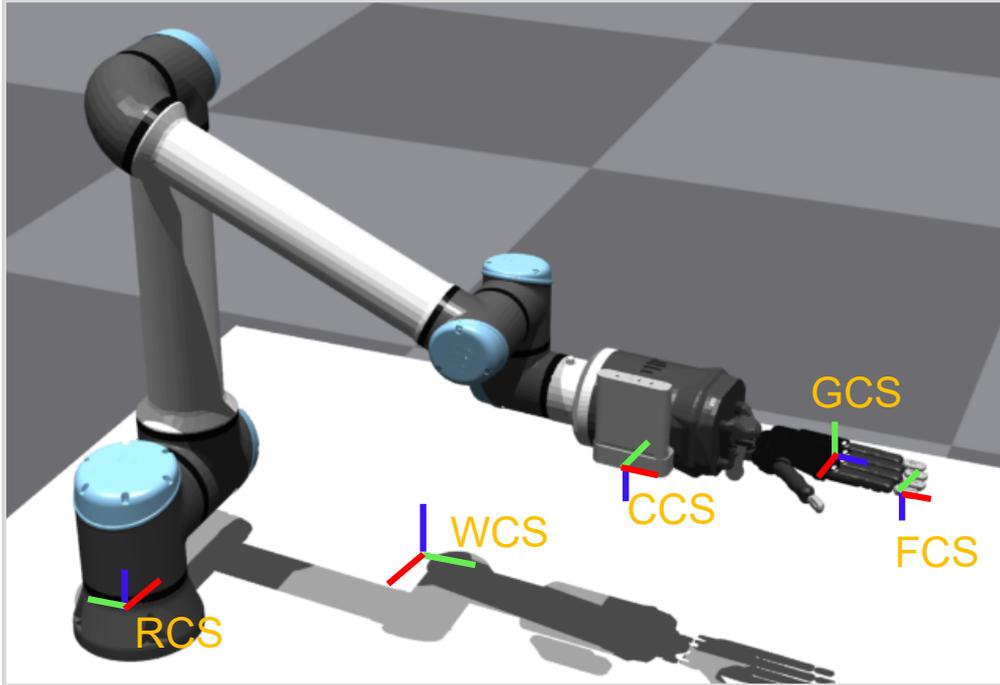


Figure 5.2: Coordinate Systems for Dexterous Grasp. Note that a unique FCS is established for each fingertip of the dexterous hand. [147]

scenario. A quantitative grasp quality metric $q \in [0, 1]$ is then evaluated based on variables such as the object’s pose \mathbf{P}_O , the hand’s pose \mathbf{P}_H , the contact forces and torques, the tactile feedback, and the visual data. Additionally, a binary grasp status s is recorded to indicate whether each grasp trial is successful (see Section 5.3).

5.2.4 Multimodal Learning for Grasp Prediction

In this work, we propose a versatile neural network, ADG-Net, designed to learn robust dexterous grasping principles and predict grasp status through multimodal learning. ADG-Net is trained on millions of synthetic grasp scenarios generated via our two-stage grasp simulation framework and operates in two distinct modes corresponding to the pre-grasp and grasp optimization phases. In the pre-grasp mode, the network processes an RGB-D image of the grasping scene to predict a feasible grasp pose $\hat{\mathbf{P}}_H$, corresponding joint angles $\hat{\theta}$, a grasp quality score \hat{q} , and a binary grasp status \hat{s} . This prediction is denoted as

$$(\hat{\mathbf{P}}_H, \hat{\theta}, \hat{q}, \hat{s}) = \text{ADG-Net}(I_C, \mathbf{0}, \mathbf{0}).$$

Subsequently, once the dexterous hand makes contact with the target object, the network refines its predictions by integrating real-time joint angles θ and tactile sensing data \mathbf{f}_{tac}^k , resulting in the updated prediction

$$(\hat{\mathbf{P}}_H, \hat{\theta}, \hat{q}, \hat{s}) = \text{ADG-Net}(I_C, \theta, \mathbf{f}_{tac}^k).$$

Detailed architecture and training procedure of ADG-Net are described in Section 5.4.

5.2.5 Multimodal Adaptive Grasping

During physical grasping experiments, RGB-D images of the grasp scenes are captured in the CCS. The proposed dexterous grasping method performs each trial in two phases: a pre-grasp phase and a grasp optimization phase, which correspond to the two operational modes of ADG-Net described in Section 5.2.4. A detailed explanation of the dexterous grasp execution process can be found in Section 5.5.

5.3 Two-Stage Grasp Simulation

In this work, we propose a pluggable two-stage simulation framework built on Isaac Gym to efficiently generate a large-scale grasp dataset—a critical resource for multimodal learning in dexterous grasping. To manage computational complexity, our simulation is based on two key assumptions: (1) the dexterous grasp simulation is performed under quasi-static physics with Coulomb friction, assuming that the maximum static friction force equals the sliding friction force during grasping; and (2) each target object is modeled as a rigid body. These assumptions, which are commonly employed in existing grasp simulation methods [61, 72, 149], not only simplify the simulation but also help to carefully evaluate grasping trials by reducing the risk of object slippage.

The following subsections (Sections 5.3.1 to 5.3.6) detail the comprehensive procedures involved in our dexterous grasp simulation using a ShadowHand [108]. These include an overall framework overview, modeling of the grasping scenarios, the two-stage grasp simulation process, and the methods used for large-scale dataset generation.

5.3.1 Overview

As illustrated in Fig. 5.3, our pluggable two-stage grasp simulation framework is organized into three packages: the core Isaac Gym environment, a set of custom pluggable modules, and additional functional modules from existing Python libraries.

Our framework leverages Isaac Gym’s robust simulation capabilities and GPU acceleration to form the foundation for dexterous grasp simulation. To manage complex 3D objects efficiently, a convex decomposition module divides these shapes into simpler convex parts, facilitating mesh management, collision detection, and object interaction. In addition, we introduce a novel grasp configuration module that employs motion planning to schedule hand configurations—adjusting joint angles and movement trajectories—to achieve dexterous grasping. Building on Isaac Gym’s collision detection functions, our framework incorporates dedicated modules for contact detection and force-torque calculation; an additional mathematical optimization module further refines this process by providing detailed data for each contact point, including coordinates, normals, normal forces \mathbf{f}_n^i , friction forces \mathbf{f}_r^i , and torques τ^i at contact points \mathbf{p}^i . Furthermore, three additional pluggable modules are integrated to assess grasp quality, simulate tactile sensing, and manage data recording and reconstruction. The mathematical optimization module is also used to refine the force-torque configurations at contact points and evaluate grasp robustness for each trial.

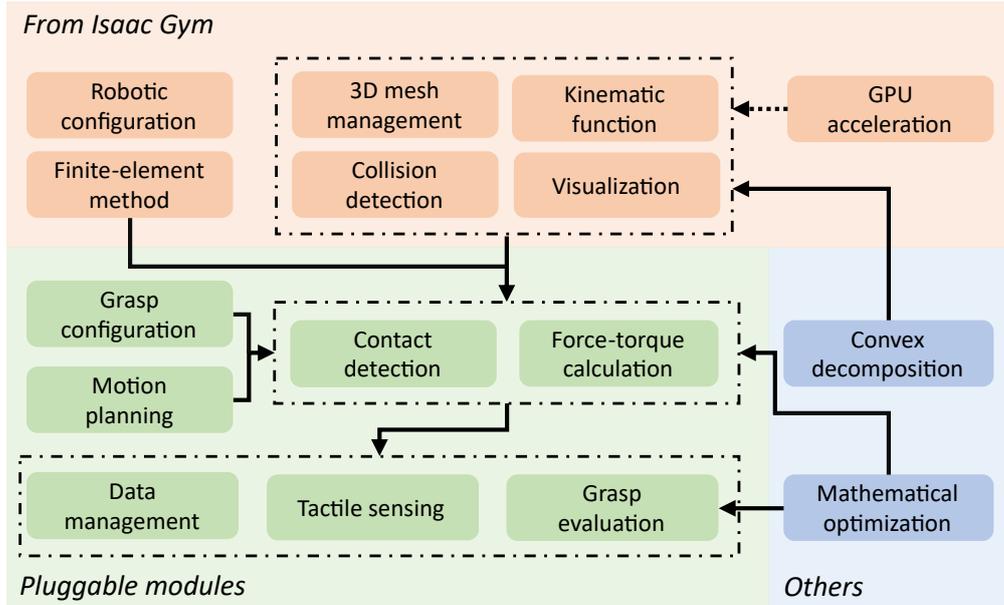


Figure 5.3: Framework of the pluggable two-stage grasp simulation based on Isaac Gym. [147]

To accelerate dataset generation, our two-stage simulation approach divides the grasping process into two distinct phases. In the first phase, a fast simulation using a rigid-body hand model is performed—this bypasses the need for tracking soft fingertip deformations and tactile sensor data until contact occurs, thereby improving simulation speed. In the second phase, the simulation is refined by modeling the hand with a rigid structure combined with soft fingertips, enabling precise optimization of the contact surface and detailed analysis of contact forces \mathbf{f}^i and torques $\boldsymbol{\tau}^i$ at each contact point \mathbf{p}^i using FEMs. This two-stage approach enables the efficient generation of a large-scale dexterous grasping dataset that incorporates soft tactile sensing.

Moreover, the proposed pluggable simulation framework is designed to be adaptable and can be integrated with other simulation platforms such as Isaac Sim, MuJoCo [126], PyBullet, and Kubric [35] to support a variety of novel simulation tasks.

5.3.2 Grasping Scenario Modeling

A typical grasping scenario in our two-stage simulation framework includes a left ShadowHand equipped with tactile sensors (\mathbf{H}), a target object (\mathbf{O}), a wrist-mounted Kinect Azure RGB-D camera (\mathbf{C}), a UR10e robot (\mathbf{R}), and a table, as illustrated in Fig. 5.2. In this setup, the table is defined by the plane $\Pi(x, y, 0)$ in the WCS of the grasp scene. Following the method described in [71], a sequence of stable poses for the object, denoted as $\mathbf{P}_{\mathbf{O}}$, is computed, and the object is then placed on the table in a randomly selected stable pose from $\mathbf{P}_{\mathbf{O}}$ to emulate realistic object placement.

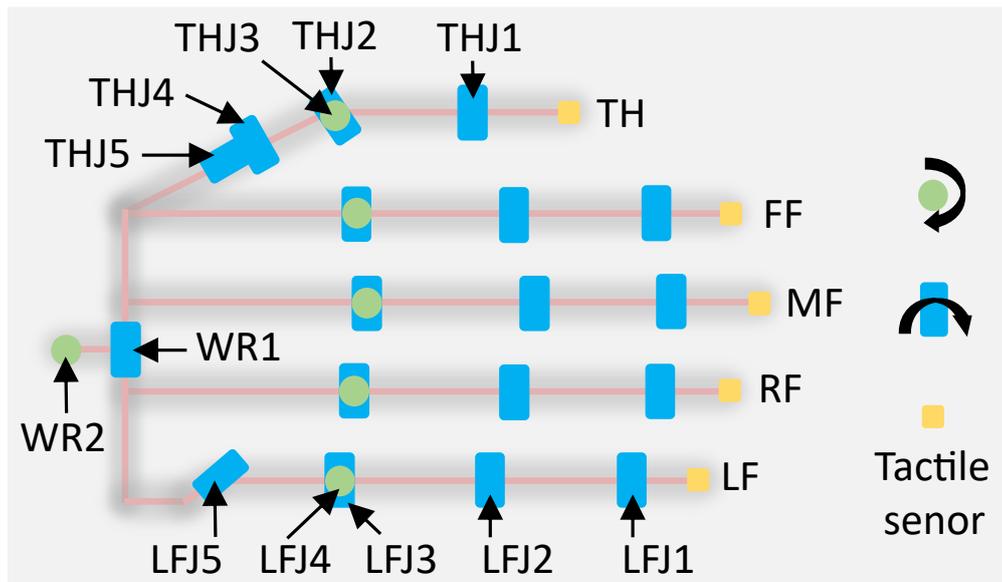


Figure 5.4: 24 DOFs of a left ShadowHand. [147]

5.3.3 First Stage: Fast Contact

In this stage, the dexterous ShadowHand is modeled as a rigid-body system with 24 DOFs, a configuration that has been widely adopted in previous studies [63, 135, 142]. Figure 5.4 illustrates the 24 DOFs of the ShadowHand. Unlike some approaches that limit collision detection to a few predefined fingertip contact candidates, our simulation considers the entire palm and all finger surfaces as potential collision regions [63, 135, 142]. In the figure, the labels TH, FF, MF, RF, and LF denote the thumb, first finger, middle finger, ring finger, and little finger, respectively. Corresponding joint names are assigned as THJ1 to THJ5, FFJ1 to FFJ4, MFJ1 to MFJ4, RFJ1 to RFJ4, and LFJ1 to LFJ5. The ShadowHand is driven by a cable-actuated system, with joint torques, speeds, and angles managed by dedicated motors. Some joints are controlled in pairs (e.g., FFJ1-FFJ2, MFJ1-MFJ2, RFJ1-RFJ2, and LFJ1-LFJ2), while the remaining joints are independently actuated by motor-cable units. The first stage of our proposed dexterous grasping simulation is carried out in two sequential steps.

Initialize pre-grasp pose

Prior research has shown that effective initialization can significantly boost the success rate of dexterous grasp simulation [135, 142]. In our work, we initialize the pre-grasp configuration of (\mathbf{H}) by combining a predefined open-hand pose with a truncated normal distribution to set the joint angles [135].

To achieve this, we first generate a set of candidate angles, denoted as Φ , for each joint. This is done by considering the joint's operational range and sampling from a truncated normal distribution with a standard deviation of $\Phi_\sigma = 1.0$ and a mean set to the midpoint of that range; for each joint, 10,000 candidate angles are produced. In a given grasping scenario, a random point $\mathbf{c}(x, y, z)$ on the upper surface of the target

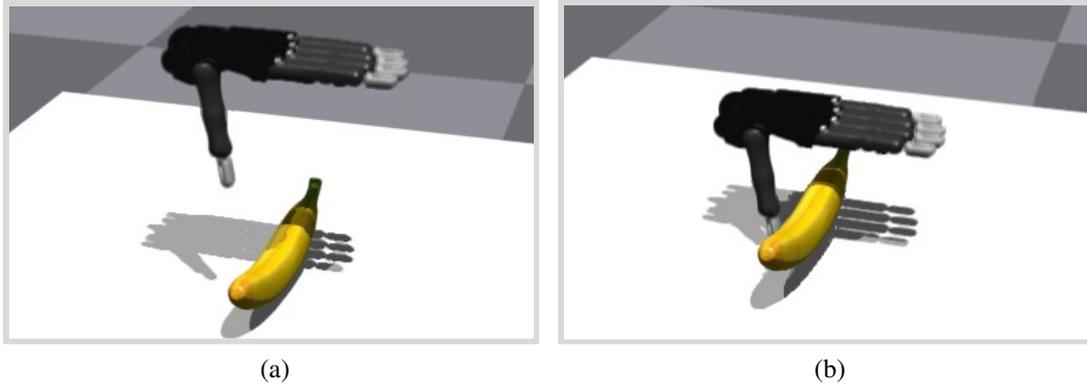


Figure 5.5: Example of pre-grasp initialization [147]: **(a)** The dexterous hand adopts an open-hand configuration; **(b)** the hand makes contact with the object or the table. Note: To accelerate large-scale dataset generation, the movement ranges of both the dexterous hand and the robot are constrained as described in Section 5.3.1, so the robot is not visualized.

object \mathbf{O} is selected to serve as the grasp center. We then establish the HCS by assuming MF is aligned with the x -axis and that a vertical z -axis passes through \mathbf{c} ; this ensures that the palm is properly oriented toward the object (see Fig. 5.2). The ShadowHand is positioned with a pose \mathbf{P}_H at a distance of 250.0 mm from \mathbf{c} , as illustrated in Fig. 5.5(a). To achieve the open-hand configuration, we set THJ4 to 70.0° and adjust joints THJ5, FFJ4, MFJ4, RFJ4, LFJ4, and LFJ5 using the corresponding angles selected from Φ . Once these joints reach their desired orientations, their positions are fixed for the subsequent simulation steps.

Move the dexterous hand

In our simulation, the dexterous hand \mathbf{H} advances along the negative z axis of the HCS until it contacts either the table or the object \mathbf{O} (see Fig. 5.5(b)). Next, for the remaining joints, we randomly select a set of desired angles from the candidate set Φ . These joints then move toward the selected target angles until they collide with either the table or the object, with each joint’s movement being moderated by its specific torque threshold τ_{jt} . Once all joints have ceased movement, the initial joints of THJ1, FFJ1, MFJ1, RFJ1, and LFJ1 are each adjusted by -1.0° to ensure that the fingertips do not inadvertently contact the object prior to the second-stage grasping process described in Section 5.3.4.

All these movements are strictly constrained in both Cartesian and joint spaces according to the physical setup of the robot, ensuring that the final grasp pose and joint configurations are achievable on the real-world system. This initialization schedule, which mirrors human grasping strategies, enables the generation of diverse grasp poses for large-scale dataset simulation. The grasping scenario and the resulting hand states—including the I_C , \mathbf{P}_O , \mathbf{P}_H , and θ —are recorded for subsequent processing in Section 5.3.4.

Figures 5.6(a)–(c) display three representative grasping trials from the first stage of our simulation. These examples demonstrate that our method can simulate dexter-

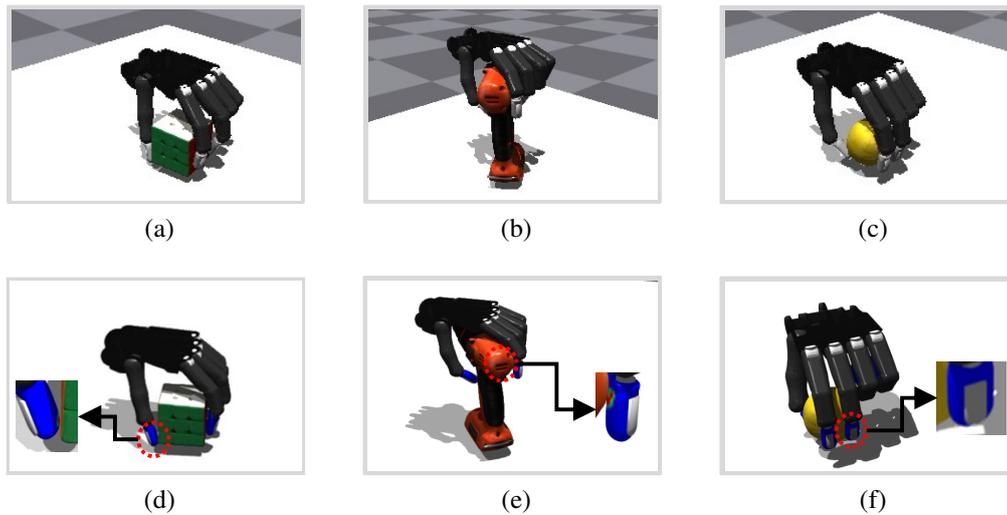


Figure 5.6: Three example grasping trials generated by our dexterous grasp simulation framework [147]. Panels (a)–(c) show the first stage of the simulation, where the hand is modeled as a rigid-body system, with fingertips visualized in grey. Panels (e)–(f) illustrate the second stage, which incorporates soft fingertip deformations alongside the rigid-body hand; these panels correspond to the grasp scenes shown in Fig. 5.6(a)–(c), with the soft fingertip deformations depicted using heat maps. Note that grasp scene masks are recorded during simulation to allow for the application of various textures to the white table, thereby increasing dataset diversity. Additionally, the viewpoints for these grasp scenes have been adjusted to ensure clear and informative visualization of the grasping trials.

ous grasps with a variety of contact regions. For instance, Fig. 5.6(c) shows a scenario in which the object is “squeezed” by the side of the ring finger, a case that would be challenging to simulate using approaches that limit contact candidates to predefined fingertip regions [135]. Furthermore, the robotic actions in these simulations conform to the physical limitations of our real-world setup, ensuring that the training data provided to ADG-Net is grounded in realistic grasping principles, thereby enabling it to learn to detect feasible grasps in practical applications.

5.3.4 Second Stage: Grasp Optimization and Tactile Sensing

In the second stage, the dexterous hand \mathbf{H} is modeled as a rigid-body hand augmented with soft fingertips. Unlike the first stage, each fingertip is now represented by a rigid skeleton combined with a high-resolution deformable shell composed of 4,246 vertices and 15,410 tetrahedrons [88] (see Fig. 5.7). Key material properties—including the friction coefficient μ , Young’s modulus ξ , and Poisson’s ratio ν —are incorporated into the soft fingertip model.

To further simulate grasping and tactile sensing, the grasping scenario described in Section 5.3.3 is reconstructed. In this refined stage, the soft fingertips approach the target object using the motions of joints THJ1, FFJ1, MFJ1, RFJ1, and LFJ1. A 3D FEM is then employed to track fingertip deformations and capture tactile interactions. The

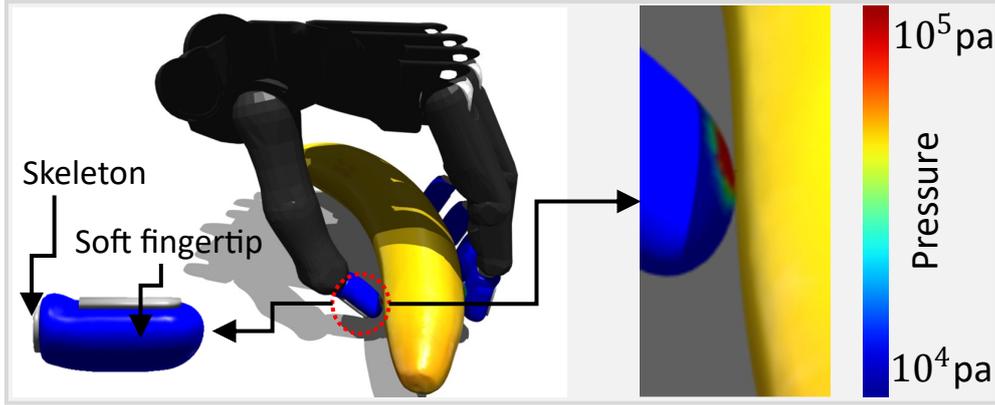


Figure 5.7: Modeling of the dexterous hand and tactile sensing via the FEM in the second stage. [147]

rotations of these joints continue until they reach their desired angles or until the torque threshold τ_{jt} is met. At that point, the joint angles θ and the hand pose \mathbf{P}_H are updated to record the final grasp configuration. Figures 5.6(d)–(f) present three representative grasping trials from the second stage of our dexterous grasp simulation.

Finally, the hand \mathbf{H} lifts upward by 20.0 mm along the z axis in the WCS to attempt to grasp the object. During this motion, the simulation identifies all contact points between \mathbf{H} and the object \mathbf{O} , denoted as $\mathbf{P} = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^i, \dots\}$. The tactile forces \mathbf{f}_{tac}^k at these contact points are then computed, and the overall grasp quality q is evaluated based on the measured forces \mathbf{f}^i and torques τ^i . Detailed principles and formulas for tactile sensing, grasp quality assessment, and grasp status labeling are provided in Section 5.3.5.

5.3.5 Tactile Sensing and Grasp Quality Evaluation

Assume that N contact points are detected between the dexterous hand \mathbf{H} and the object \mathbf{O} . These points can be represented as a set of N elements, denoted by

$$\mathbf{P} = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^N\}.$$

For each contact point \mathbf{p}^i , the corresponding contact force and torque are represented by \mathbf{f}^i and τ^i , respectively.

Tactile Sensing

In our two-stage simulation, contact points detected on the soft fingertips are incorporated into the overall set \mathbf{P} . As shown in Fig. 5.7 and detailed in Section 5.3.4, tactile forces are simulated using a finite element model composed of 4,246 nodes. For simplicity, the forces at all contact points on each fingertip are aggregated into a single three-dimensional force vector, \mathbf{f}_{tac}^k (with $k = 1, 2, \dots, 5$ corresponding to the five fingertips). To facilitate the training of ADG-Net, these tactile force vectors are normalized according to Equation (5.1) and then transformed into the FCS. A similar normalization

process is applied to the joint angles θ using Equation (5.2), where each joint angle θ_i is scaled based on its movement range, defined by $\min(\theta_i)$ and $\max(\theta_i)$.

$$\mathbf{f}_{tac}^k = \frac{\mathbf{f}_{tac}^k}{\max(\|\mathbf{f}_{tac}^1\|, \|\mathbf{f}_{tac}^2\|, \|\mathbf{f}_{tac}^3\|, \|\mathbf{f}_{tac}^4\|, \|\mathbf{f}_{tac}^5\|)} \quad (5.1)$$

$$\theta_i = \frac{\theta_i - \min(\theta_i)}{\max(\theta_i) - \min(\theta_i)} \quad (5.2)$$

Grasp Quality Evaluation

In this work, the force-torque wrench between the dexterous hand \mathbf{H} and the object \mathbf{O} is computed in the WCS using a set of orthogonal normal vectors \mathbf{n}_x , \mathbf{n}_y , and \mathbf{n}_z as defined in Equation (5.3). Given the object's mass m and the gravitational acceleration g , the hand \mathbf{H} is able to pick up the object \mathbf{O} when the force-torque wrench remains in equilibrium, i.e., when $\sum \mathbf{f}^i - mg\mathbf{n}_z = \mathbf{0}$ and $\sum \tau^i = \mathbf{0}$.

$$\mathbf{n}_x = (1, 0, 0), \mathbf{n}_y = (0, 1, 0), \mathbf{n}_z = (0, 0, 1) \quad (5.3)$$

$$\mathbf{f}^i = \mathbf{f}_\mu^i + \mathbf{f}_n^i \quad (5.4)$$

$$\mathbf{f}_\mu^i = \mu \|f_n^i\| \cdot (f_{\mu x}^i, f_{\mu y}^i, f_{\mu z}^i) \quad (5.5)$$

$$\mathbf{f}_\mu^i \cdot \mathbf{f}_n^i = \mathbf{0} \quad (5.6)$$

$$0 \leq \|(f_{\mu x}^i, f_{\mu y}^i, f_{\mu z}^i)\| \leq 1 \quad (5.7)$$

$$\mathbf{f}_n^j = \frac{\tau_{jt}}{r_{jt}^i} \mathbf{n}_p^j, \text{ for } \mathbf{p}^j \in \mathbf{P}_{fig} \quad (5.8)$$

$$\mathbf{f}_n^h = -\frac{1}{H} \left(\sum_{j=1}^J \mathbf{f}_n^j - mg\mathbf{n}_z \right) \cdot \mathbf{n}_p^h, \text{ for } \mathbf{p}^j \in \mathbf{P}_{fig} \text{ and } \mathbf{p}^h \in \mathbf{P}_{palm} \quad (5.9)$$

$$\mathbf{F} = \sum_{i=1}^N (\mathbf{f}_\mu^i + \mathbf{f}_n^i) - mg\mathbf{n}_z = \mathbf{0}, \text{ for } \mathbf{p}^i \in \mathbf{P} \quad (5.10)$$

$$\mathbf{T} = \sum_{i=1}^N (\mathbf{r}^i \times (\mathbf{f}_\mu^i + \mathbf{f}_n^i)) = \mathbf{0}, \text{ for } \mathbf{p}^i \in \mathbf{P} \quad (5.11)$$

$$q = 1 - \frac{\min(\sum_{i=1}^N \|\mu(f_{\mu x}^i, f_{\mu y}^i, f_{\mu z}^i)\|)}{\sum_{i=1}^N \mu} \quad (5.12)$$

$$s = \begin{cases} False, & \text{if (5.12) is unsolvable or } q < 0.3 \\ True, & \text{others} \end{cases} \quad (5.13)$$

As illustrated in Fig. 5.8, each contact force \mathbf{f}^i is computed as the sum of the normal component \mathbf{f}_n^i and the frictional component \mathbf{f}_μ^i , where the friction coefficient between \mathbf{H} and \mathbf{O} is denoted by μ . These forces— \mathbf{f}^i , \mathbf{f}_μ^i , and \mathbf{f}_n^i —are calculated and bounded by Equations (5.4) through (5.7). Within these constraints, the scalars $f_{\mu x}^i$, $f_{\mu y}^i$, and $f_{\mu z}^i$ represent the components of \mathbf{f}_μ^i along the directions defined by \mathbf{n}_x , \mathbf{n}_y , and \mathbf{n}_z , respectively.

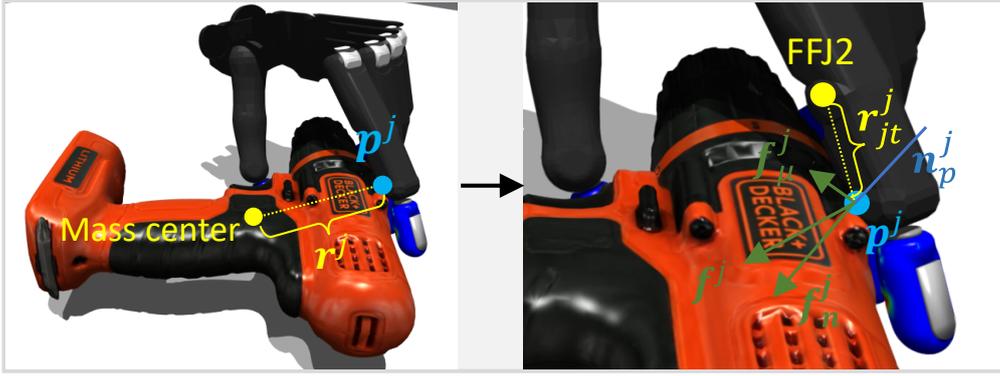


Figure 5.8: Basic information on a contact point. [147]

To determine \mathbf{f}_n^i , we first partition the overall contact point set \mathbf{P} into two subsets: one for the palm, $\mathbf{P}_{palm} = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^h, \dots, \mathbf{p}^H\}$, and one for the fingers, $\mathbf{P}_{fig} = \{\mathbf{p}^1, \mathbf{p}^2, \dots, \mathbf{p}^j, \dots, \mathbf{p}^J\}$. Thus, $\mathbf{P} = \mathbf{P}_{palm} \cup \mathbf{P}_{fig}$. For any finger contact point $\mathbf{p}^j \in \mathbf{P}_{fig}$, the normal force \mathbf{f}_n^j is determined based on the corresponding joint torque τ_{jt} , moment arm r_{jt}^j , and the contact surface normal \mathbf{n}_p^j , as described in Equation (5.8) and depicted in Fig. 5.8. For contact points on the palm, $\mathbf{p}^h \in \mathbf{P}_{palm}$, we compute \mathbf{f}_n^h using Equation (5.9) to simplify calculations. By combining Equations (5.3)–(5.11), we obtain the overall force-torque wrench for a grasping trial.

A grasp is considered successful when a stable force-torque wrench is established between \mathbf{H} and \mathbf{O} , as defined in Equations (5.10) and (5.11). Here, \mathbf{r}^i denotes the moment arm between the object’s center of mass and the contact point \mathbf{p}^i . Recall that the total force \mathbf{f}^i is composed of the normal force \mathbf{f}_n^i (which arises from τ_{jt}) and the frictional force \mathbf{f}_μ^i (which depends on μ). Given that friction coefficients vary across different gripper-object pairs, an optimal force-torque wrench for a stable grasp should minimize the friction required to maintain equilibrium. Thus, the grasp quality q is evaluated based on the minimum friction necessary to sustain a stable wrench, with the normalized grasp quality $q \in [0, 1]$ defined in Equation (5.12). By combining Equations (5.3)–(5.12), we compute q by minimizing the frictional components $(f_{\mu x}^i, f_{\mu y}^i, f_{\mu z}^i)$ via a quadratic programming problem solved using IPOPT. A grasp is deemed unsuccessful if the resulting force-torque wrench does not meet the criteria of Equation (5.12) or if q is less than 0.3; otherwise, the grasp status s is set to True, as specified in Equation (5.13).

5.3.6 Dataset Generation

Our framework enables the generation of a large-scale multimodal dataset by iteratively executing the procedures outlined in Sections 5.3.2 to 5.3.4. For dexterous grasp generation, we selected 40 high-resolution 3D meshes from the YCB dataset [12] and rescaled each mesh by factors of 0.8, 1.0, and 1.2 to simulate a variety of object sizes. The simulation was powered by Isaac Gym, configured with a 0.005-second time step, and deployed across 100 parallel environments—allowing 100 grasping scenarios to be simulated concurrently.

Table 5.1: Basic Information of the Proposed Dataset and Five SOTA Datasets

Dataset	Framework	Gripper	Visual data	Joint angle	Tactile sensing	Scenario
DexGraspNet [135]	Sim	SH	-	Yes	-	1.32M
RealDex [57]	Real	SH	RGB-D image, point cloud	Yes	-	59K
GPPNs [117]	Real	SH Lite	RGB image	-	-	8K
DDGdata [63]	Sim2Real	SH	-	Yes	-	6.9K
GEN [29]	Sim2Real	RF	Point cloud	Yes	-	36K
Ours	Sim2Real	SH	RGB-D image	Yes	Yes	500K

During the two-stage grasp simulation, the RGB-D images I_C were rendered at a resolution of $240 \times 240 \times 4$ pixels. We set the joint torque threshold to $\tau_{jt} = 0.01$ N·m and used a friction coefficient of $\mu = 1.2$. After extensive calibration with a real-world ShadowHand, we determined that setting the Young’s modulus to $\xi = 1.5$ M and the Poisson’s ratio to $\nu = 0.3$ yielded the best performance in terms of minimizing the Sim2Real gap. For each target object, we generated 100 random poses to diversify the grasping scenarios. Depending on the object’s size, each scenario involved sampling between 50 and 500 grasp centers \mathbf{c} , and for each center, 1,000 grasping trials were executed using varied hand poses \mathbf{P}_H and joint angles. The grasp trial achieving the highest quality score q was then recorded as the optimal configuration for that scenario. Consequently, each grasp example in our dataset includes the RGB-D image I_C , the object pose \mathbf{P}_O , the hand pose \mathbf{P}_H , joint angles θ , tactile sensing data \mathbf{f}_{tac}^k , grasp quality q , and a binary grasp status s .

Dataset generation was performed on a PC running Ubuntu 20.04, equipped with an AMD Ryzen 9-5960X CPU, 64 GB of DRAM, and two Nvidia RTX 3090 GPUs. Over a total runtime of 700 hours, more than 1 million grasp trials were synthesized. From these, 500,000 grasps were selected for multimodal grasp learning, evenly split between successful and unsuccessful trials (250K each). Furthermore, the synthetic grasps were archived with object-wise separation to ensure that the objects used for ADG-Net training were not included in the testing dataset.

Finally, Table 5.1 provides a comparative overview of our proposed dataset alongside five state-of-the-art dexterous grasping datasets. This comparison demonstrates that our dataset not only encompasses a wide range of grasp scenarios and multimodal sensing data but also uniquely includes tactile sensing data from soft fingertips—a feature that is often absent in existing datasets.

5.4 Multimodal Learning for Grasp Prediction

5.4.1 Architecture of ADG-Net

ADG-Net operates in two distinct modes to infer grasp parameters by fusing information from both RGB-D images and multimodal sensing data, as outlined in Section 5.2.4. Specifically, the network accepts as input a $240 \times 240 \times 4$ RGB-D image I_C , together with joint angles θ and tactile force measurements \mathbf{f}_{tac}^k . From these inputs, ADG-Net predicts a suite of grasp parameters: an estimated grasp pose $\hat{\mathbf{P}}_H$, refined joint angles $\hat{\theta}$, a grasp quality score \hat{q} , and a binary grasp status \hat{s} .

The architecture of ADG-Net, depicted in Fig. 5.9, is built with multiple input branches that feed into shared processing layers and then diverge into separate output streams. In one branch, GCN integrates the joint angles and tactile data; the GCN’s connectivity is defined based on the kinematic chain of the ShadowHand, as illustrated in Fig. 5.4. This graph-based module enables the network to capture the relational structure inherent in the sensory inputs, thereby learning key principles of dexterous grasping. In parallel, an RGB-D image branch processes visual information using ResNet encoders arranged in a pyramid topology and enhanced by CBAM [139]. The CBAM allows the network to focus on critical regions of the image that are most informative for determining the grasp pose. Finally, ADG-Net features four dedicated output branches that individually predict $\hat{\mathbf{P}}_{\mathbf{H}}$, $\hat{\theta}$, \hat{q} , and \hat{s} . Each output branch concludes with a Sigmoid activation to normalize the predictions, helping to avoid extreme output values and enhancing overall robustness.

5.4.2 Training of ADG-Net

A total of 420K synthetic grasp examples were used to train ADG-Net, with an additional 30K examples reserved for testing and the remaining 50K allocated for the experiments described in Section 5.6.1. The network was trained on a PC equipped with two Nvidia H100 GPUs. The training process was divided into two phases—one focusing on image-based learning and the other on multimodal integration—corresponding to the dual operational modes outlined in Section 5.4.1.

Image-Based Learning

In this phase, ADG-Net processes an input RGB-D image $I_{\mathbf{C}}$ to predict feasible grasp parameters as specified in (5.14), which is critical for determining a viable grasp configuration prior to any contact with the target object. Because joint angles and tactile feedback are not available until after grasping, we disable the input channels for θ and \mathbf{f}_{tac}^k (as well as the associated shared layers) during the image-based training phase to prevent any unintended gradient flow through these components.

$$(\hat{\mathbf{P}}_{\mathbf{H}}, \hat{\theta}, \hat{q}, \hat{s}) = \text{ADG-Net}(I_{\mathbf{C}}, \mathbf{0}, \mathbf{0}) \quad (5.14)$$

Given that ADG-Net exhibits a linear fitting behavior for grasp parameter estimation, we employ Mean Squared Error (MSE) loss functions—defined in (5.15)–(5.19)—as the objective for optimizing network parameters, using the Adam optimizer [48]. Specifically, four separate sub-loss functions (outlined in (5.15)–(5.18)) are assigned to each output branch. The overall training loss, as detailed in (5.19), is computed as the sum of these sub-losses, following established practices from previous research [85, 150]. After preliminary training trials, we set the learning rate to 0.0001 and use a batch size of 1000 for the image-based learning phase. The loss weights $\lambda_{\mathbf{p}}$, λ_{θ} , λ_q , and λ_s are all fixed at 1.0, as this configuration has been shown to yield robust performance.

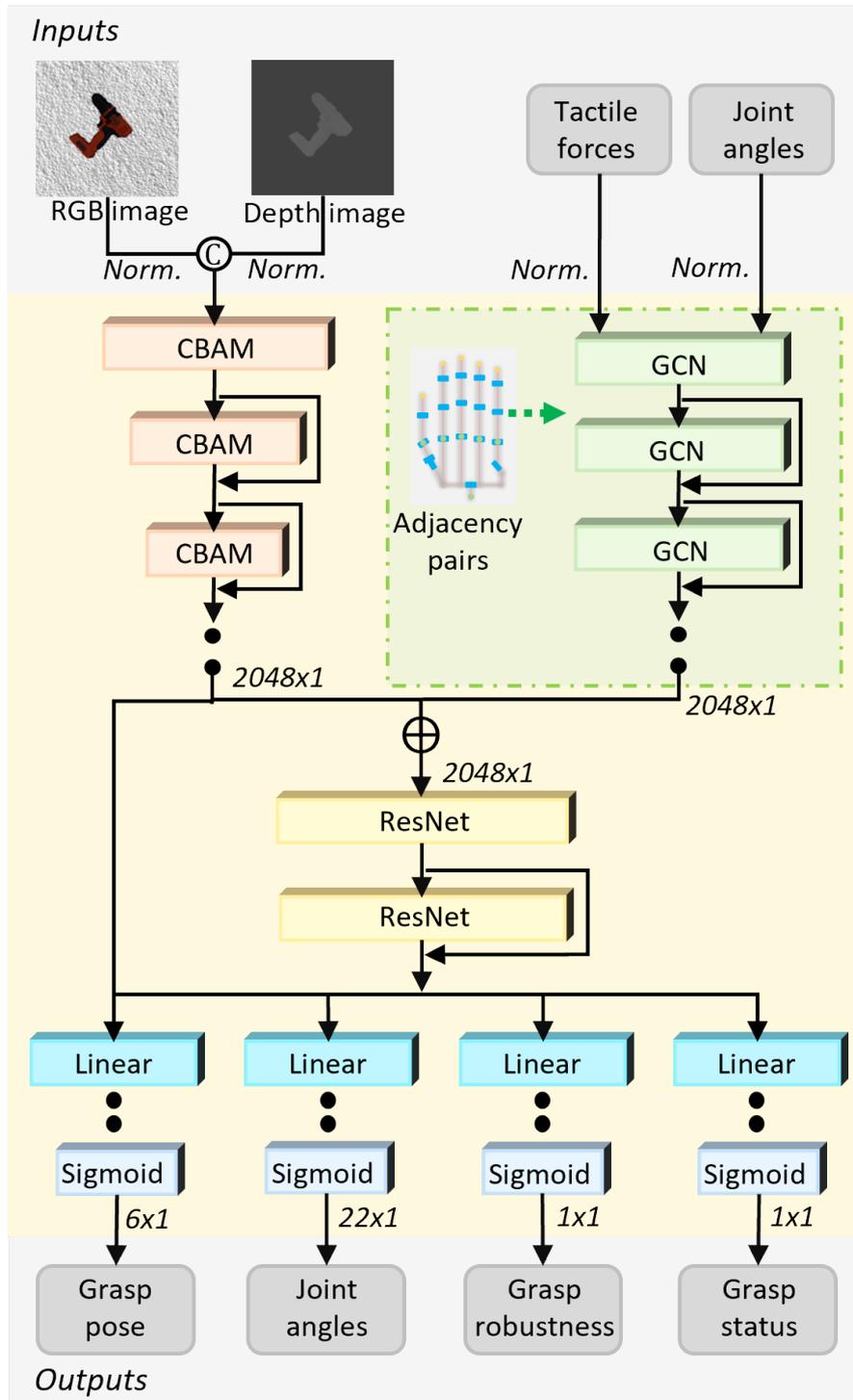


Figure 5.9: Architecture of the proposed ADG-Net. Note: 1) The ADG-Net takes in 22-DOF joint angles without WR1 and WR2, since WR1 and WR2 are integrated into the kinematic chain of the robot for robotic motion planning. 2) \otimes means the Channel Concatenation, and \oplus is the Element-Wise Addition in the neural network. 3) “Norm.” is the abbreviation of “normalization”.

$$\mathbf{L}_P = \text{MSE}(\hat{\mathbf{P}}_H - \mathbf{P}_H) \quad (5.15)$$

$$\mathbf{L}_\theta = \text{MSE}(\hat{\theta} - \theta) \quad (5.16)$$

$$\mathbf{L}_q = \text{MSE}(\hat{q} - q) \quad (5.17)$$

$$\mathbf{L}_s = \text{MSE}(\hat{s} - s) \quad (5.18)$$

$$\mathbf{L}_{rgbD} = \lambda_P \mathbf{L}_P + \lambda_\theta \mathbf{L}_\theta + \lambda_q \mathbf{L}_q + \lambda_s \mathbf{L}_s \quad (5.19)$$

Multimodal Learning

In this phase, ADG-Net processes the RGB-D image I_C together with joint angle information θ and tactile force data \mathbf{f}_{tac}^k to estimate the grasp parameters as defined in (5.20). These predicted parameters are then used to refine and optimize the grasp configuration as the dexterous hand contacts the target object. Importantly, because adjusting the predicted hand pose $\hat{\mathbf{P}}_H$ after contact could lead to an unstable grasp (and potentially cause the object to be dropped), the output branch responsible for $\hat{\mathbf{P}}_H$ is deactivated during this phase to maintain a fixed hand pose.

Following a similar strategy as in the image-based learning phase, we define the MSE loss functions for the multimodal learning stage in Equations (5.16)–(5.18) and (5.21). After preliminary training runs, we set an initial learning rate of 0.0001 and a batch size of 1,000, with the loss weights $\lambda_\theta = \lambda_q = \lambda_s = 1.0$.

$$(\hat{\mathbf{P}}_H, \hat{\theta}, \hat{q}, \hat{s}) = \text{ADG-Net}(I_C, \theta, \mathbf{f}_{tac}^k) \quad (5.20)$$

$$\mathbf{L}_{mlt} = \lambda_\theta \mathbf{L}_\theta + \lambda_q \mathbf{L}_q + \lambda_s \mathbf{L}_s \quad (5.21)$$

To optimize the architecture of ADG-Net, we trained over 50 network variants and evaluated them based on their computational times t_{rgbD} and t_{mlt} for image-based and multimodal processing, respectively, as well as the average prediction errors ϵ_P , ϵ_θ , ϵ_q , and ϵ_s (as defined in (5.22)–(5.23)), where \oplus represents the exclusive OR operation. During training, we experimented with various configurations for the RGB-D image branch by varying the depth of the CBAM modules from 1 to 10 layers, adjusting the channel dimensions from 32 to 256, and modifying kernel sizes from 1 to 9. Similarly, we optimized the depth of the GCNs from 1 to 5 layers and varied the depth of the shared ResNet modules between 1 and 5 layers. The final architecture of ADG-Net comprises 230 million parameters. Specifically, the RGB-D fusion branch incorporates four CBAMs configured with channel sizes 128-64-32-32 and kernel sizes 9-3-3-1. Additionally, five GCN units with 2048 features are used to fuse tactile force and joint angle data, while two shared ResNets integrate this multimodal information before it is passed through three linear layers in each output branch. More detailed information on the ADG-Net parameters can be found in the source code linked in this work. In operation, ADG-Net achieves a processing time of $t_{rgbD} = 22$ ms for image-based grasp estimation and $t_{mlt} = 25$ ms for multimodal grasp estimation.

Detailed performance metrics are discussed in Section 5.6.

$$\varepsilon_{\mathbf{P}} = \|\hat{\mathbf{P}}_{\mathbf{H}} - \mathbf{P}_{\mathbf{H}}\|, \quad \varepsilon_{\theta} = \|\hat{\theta} - \theta\|, \quad \varepsilon_q = |\hat{q} - q|_{abs} \quad (5.22)$$

$$\varepsilon_s = \hat{s} \oplus s \quad (5.23)$$

5.5 Multimodal Adaptive Grasping

As described in Section 5.2.5, our adaptive dexterous grasping approach using ADG-Net is implemented in two separate stages: an initial pre-grasp stage and a subsequent grasp optimization stage. These stages correspond to the two operational modes of ADG-Net, which are further elaborated in Section 5.4.

Pre-Grasp Stage

In this stage, we first apply a bilateral filter to the real-world depth images to suppress random noise by averaging pixel values based on both spatial proximity and radiometric similarity. The resulting normalized depth image, denoted as I_{real} , is then provided as input to ADG-Net. The network subsequently estimates a feasible grasp pose $\hat{\mathbf{P}}_{\mathbf{H}}$ and the corresponding joint angles $\hat{\theta}$ required for the dexterous hand to effectively wrap around the target object. Simultaneously, ADG-Net predicts a grasp quality score \hat{q} and a binary grasp status \hat{s} , as defined in (5.24). Based on these predictions, the hand proceeds to approach and secure the object. Because ADG-Net's outputs are normalized using Sigmoid activations, all predicted parameters must be mapped back to their original value ranges to ensure that the executed grasp is valid (see Fig. 5.7).

$$(\hat{\mathbf{P}}_{\mathbf{H}}, \hat{\theta}, \hat{q}, \hat{s}) = \text{ADG-Net}(I_{real}, \mathbf{0}, \mathbf{0}) \quad (5.24)$$

$$(\hat{\mathbf{P}}_{\mathbf{H}}, \hat{\theta}, \hat{q}, \hat{s}) = \text{ADG-Net}(I_{real}, \theta_{real}, \mathbf{f}_{tac}^k) \quad (5.25)$$

Grasp Optimization Stage

During the second stage of grasp optimization, once the dexterous hand makes contact with the object, real-time tactile forces \mathbf{f}_{tac}^k and current joint angles θ_{real} are acquired and normalized (see Section 5.3.5). These multimodal sensing inputs are then fed into ADG-Net, which refines the joint angles and re-assesses both the grasp quality and status as outlined in (5.25). Based on the refined predictions, the ShadowHand's joint angles are adjusted to improve the grasp configuration, thereby enhancing the overall grasp execution through better optimization of its motion.

The updated joint angles $\hat{\theta}$ produced by ADG-Net are set as the target configuration for the dexterous hand. MPC is then used to drive the five fingers toward these target angles. Each finger's joint configuration is represented as a vector \mathbf{x}_i , with the complete configuration denoted by $\theta = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_5)$. MPC is formulated as an optimal control problem over a finite horizon N , starting from the current state θ_0 and following the linear dynamic model $\theta_{k+1} = \theta_k + \mathbf{v}_k \Delta t$. In addition, the Jacobian $\mathbf{J}_{m,0}$ for each fingertip is computed at the current time step. The primary goal of the MPC is to drive

the hand toward θ while avoiding collisions and respecting the kinematic constraints defined in (5.26), where d_m is the predefined safe distance between the fingers and the table. This results in a quadratic programming (QP) problem that is efficiently solved using an interior point optimizer (IPOPT). To prevent collisions with the table, the table is modeled as a separate plane in WCS with a normal vector $\mathbf{n} = (0, 0, 1)$ and a reference point \mathbf{c}_t . In our experiments, two different table configurations were used for tests with ShadowHands, one with BioTac Sensors and one without (details in Section 5.6.2). Given that ADG-Net requires 25 ms for each multimodal grasp estimation, the adaptive control loop is executed at 5 Hz during real-world grasping. Additionally, we integrated a collision avoidance algorithm from our previous work [148] to prevent unintended contacts between the robot and non-target objects, even though ADG-Net learns some collision-avoidance strategies from the synthetic data. Finally, the range of motion for the WR2 joint (shown in Fig. 5.4) is restricted to protect the electrical cable on the ShadowHand; this limitation does not hinder grasping performance, as the kinematics of WR1 and WR2 are coordinated with other joints to compensate for the restriction.

$$\begin{aligned}
& \underset{\theta}{\text{minimize}} && \sum_{k=0}^{N-1} \|\theta_k - \hat{\theta}\|^2 \\
& \text{subject to} && \theta_0 = \theta_{init}, \\
& && \theta_{k+1} = \theta_k + \mathbf{v}_k \Delta t, \\
& && \underline{\dot{\theta}} \leq \mathbf{v}_k \leq \bar{\dot{\theta}}, \\
& && \mathbf{x}_{m,k} = \mathbf{x}_{m,init} + \mathbf{J}_{m,init} (\theta_{m,k} - \theta_{m,init}), \\
& && \mathbf{n}^\top \mathbf{x}_{m,k} \geq d_m, \quad m = 1, \dots, 5.
\end{aligned} \tag{5.26}$$

5.6 Experiments

This section outlines a series of extensive experiments conducted to evaluate the performance of our proposed ADG-Net and the associated dexterous grasping method. All experiments were performed on the computer system described in Section 5.3.6.

5.6.1 Experiments in Simulation

This subsection details experiments conducted on synthetic grasp scenes to evaluate ADG-Net’s performance in simulation. We examined two distinct ShadowHand configurations, as depicted in Fig. 5.10. In real-world scenarios, a ShadowHand equipped with BioTac tactile sensors can capture 3D tactile force data from multiple sensing points on each fingertip. In contrast, a ShadowHand without BioTac sensors is limited to 1D tactile force measurements along the z -axis in FCS. To simulate this latter scenario in our dataset, we reformulated the tactile force data \mathbf{f}_{tac}^k according to Equation (5.27). Consequently, we trained two versions of ADG-Net: ADG-Net I for the ShadowHand without BioTac sensors and ADG-Net II for the configuration with BioTac sensors.

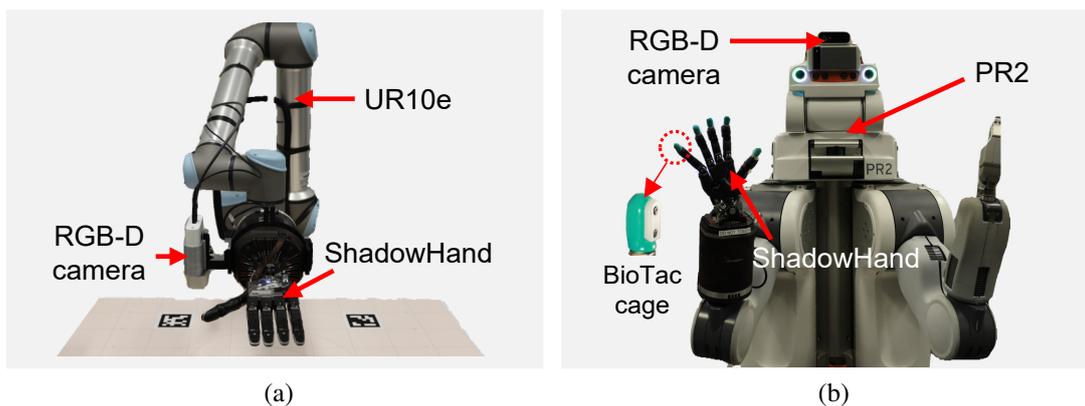


Figure 5.10: Two robotic setups with different ShadowHands for our experiments. (a) A UR10e robot and a ShadowHand without BioTac tactile sensors. (b) A PR2 robot and a ShadowHand with BioTac tactile sensors. [147]

$$\mathbf{f}_{tac}^k = \mathbf{f}_{tac}^k \cdot (0, 0, 1) \quad (5.27)$$

Table 5.2 summarizes the average prediction errors for the ADG-Nets over 50K grasping scenarios, comparing networks trained with and without BioTac sensor data. The results clearly indicate that the multimodal ADG-Nets yield lower errors in grasp parameter estimation. In particular, compared to image-based models, ADG-Net II (which utilizes BioTac data) reduces the prediction errors for joint angles (ϵ_θ), grasp quality (ϵ_q), and grasp status (ϵ_s) by 6.0%, 0.23, and 20.8% respectively—demonstrating its enhanced capability for successful real-world grasping. In contrast, ADG-Net I, lacking 3D tactile force inputs, shows higher prediction errors, underscoring the importance of BioTac’s 3D tactile data in boosting both prediction accuracy and the overall efficacy of ShadowHand’s grasping performance.

Figure 5.11 provides a joint-wise visualization of the average prediction errors calculated over 30K synthetic grasping scenarios, using ADG-Nets that combine RGB-D and multimodal data. In this figure, the errors ϵ_θ are computed according to (5.22) and presented separately for each joint.

Overall, the ADG-Nets achieve an average joint prediction error of less than 0.6° for all hand joints, which compares favorably with the average movement error of 1° reported for ShadowHands [108]. Notably, the incorporation of multimodal sensing data leads to lower average errors for joints such as THJ3, THJ4, FFJ3, FFJ4, MFJ3, MFJ4, RFJ3, RFJ4, LFJ3, and LFJ4. However, relatively higher errors are observed for the distal joints FFJ1, MFJ1, RFJ1, and LFJ1. This disparity can be attributed to the fact that joints like THJ3 and THJ4, which control the base regions of the ShadowHand fingers (analogous to the metacarpal bones and proximal phalanges of a human hand, as shown in Fig. 5.4), typically contact the upper surface of the object—a region that is well captured in a top-down RGB-D image. Conversely, distal joints such as FFJ1, MFJ1, RFJ1, and LFJ1, which correspond to the middle phalanges and fingertips, interact with

Table 5.2: Prediction Errors of two ADG-Nets for ShadowHands with/without BioTac sensors.

Neural network		RGB-D image	Multimodal data
ADG-Net I	ϵ_P	0.12	0.12
	ϵ_θ	14.3°	10.8°
	ϵ_q	0.28	0.07
	ϵ_s	24.6%	8.5%
ADG-Net II	ϵ_P	0.12	0.12
	ϵ_θ	14.3°	8.3°
	ϵ_q	0.28	0.05
	ϵ_s	24.6%	3.8%

Note: 1) ADG-Net I was trained using synthetic grasp examples without 3D BioTac tactile forces, while ADG-Net II was trained with these 3D forces. 2) The error ϵ_θ listed in the table represents the total error for all joint angles, as defined in (5.22). Detailed prediction errors for individual joint angles are illustrated in Fig. 5.11.

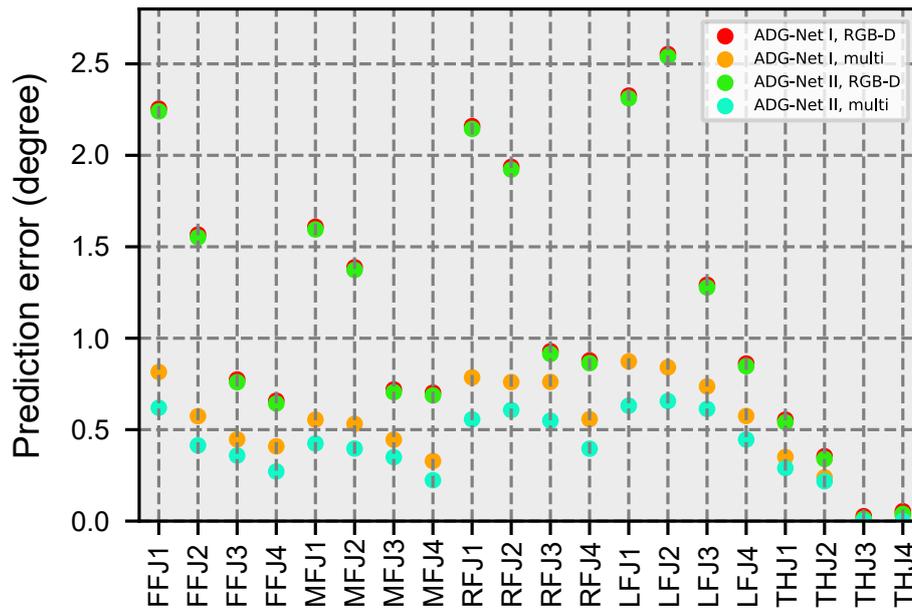


Figure 5.11: Prediction errors of joint angles based on both the RGB-D image and multimodal data. Note: Red and green dots are overlapping. [147]

the bottle's sides or lower portions and are less visible from a top-down perspective. These results highlight that multimodal sensing data can notably improve prediction accuracy for the distal joints, which are critical for successful grasping in real-world scenarios.

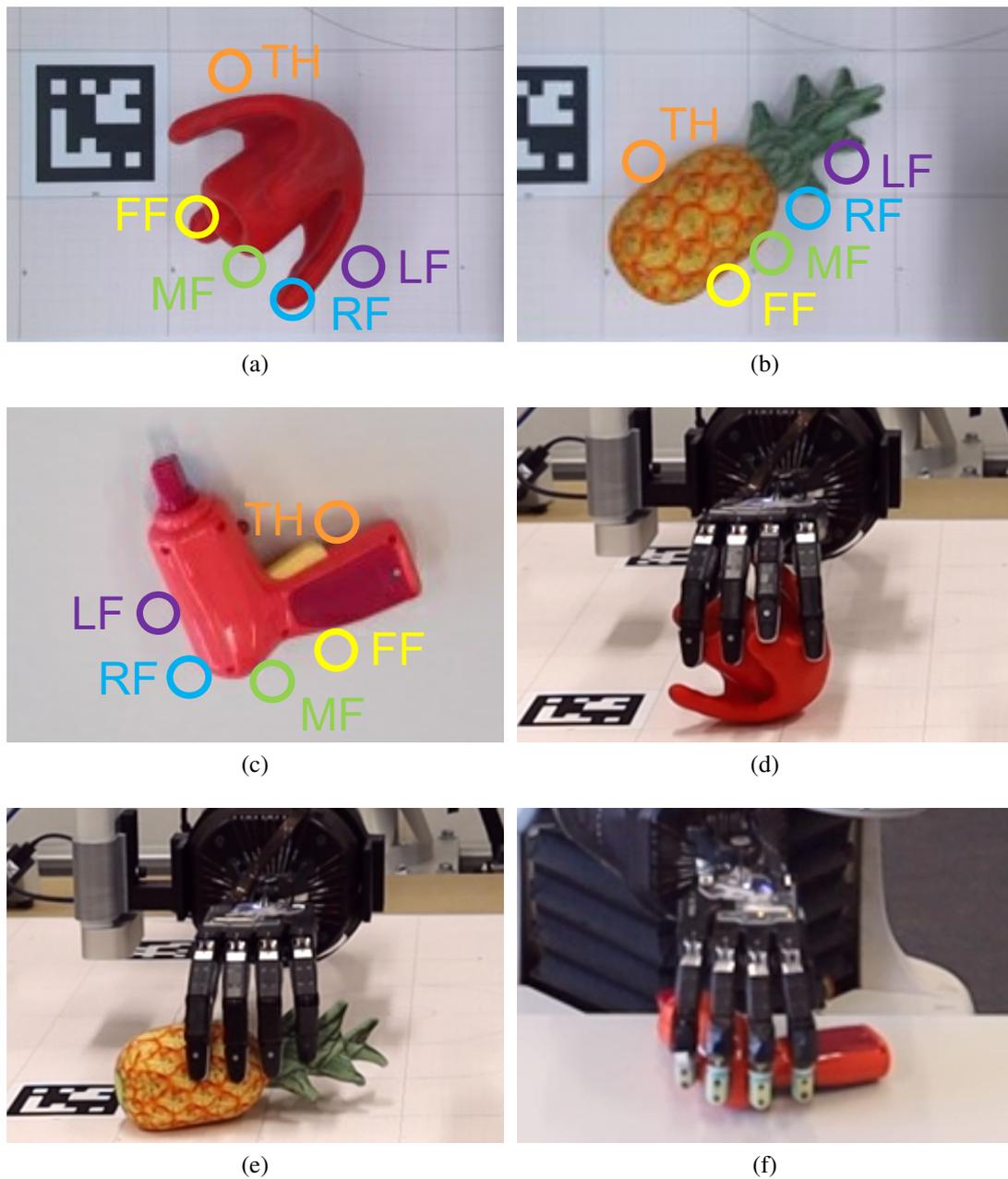


Figure 5.13: Three grasp poses detected via the ADG-Net and the corresponding grasp executions. (a), (b), (d) and (e) Two grasp poses detected via the ADG-Net I for a ShadowHand without BioTac sensors. (c) and (f) A grasp pose detected via the ADG-Net II for a ShadowHand with BioTac sensors. [147]

as opposed to the left-handed configuration used during dataset generation and in the UR10e setup, RGB-D images were mirrored before being processed by ADG-Net II. In these experiments, 3D tactile forces for each fingertip were computed based on contact point detection and data collected from multiple sensing nodes on the soft fingertips, and each method was evaluated over 100 grasping trials.

Table 5.3: Results of the Benchmark Experiments [147]

Method	Isolated		Stacked	
	<i>SRGP</i>	<i>SRG</i>	<i>SRGP</i>	<i>SRG</i>
Baseline I	-	57%	-	<30%
Baseline II	-	62%	-	<30%
GRU-M3PCA5	-	82%	-	<30%
ADG-Net I, RGB-D	71%	84%	71%	65%
ADG-Net I, multimodal	86%	91%	80%	81%
	(+15%)	(+7%)	(+9%)	(+16%)
ADG-Net II, RGB-D	72%	83%	70%	67%
	(+1%)	(-1%)	(-1%)	(+2%)
ADG-Net II, multimodal	94%	93%	92%	85%
	(+23%)	(+9%)	(+21%)	(+20%)

Note: 1) *SRG* is the abbreviation of “success rate of grasping”. 2) *SRGP* is the abbreviation of “success rate of grasp status prediction”.

The performance outcomes, summarized in Table 5.3, are reported in terms of the success rate of grasp status prediction (*SRGP*) and the actual grasping success rate (*SRG*) for both isolated and stacked objects. Baseline I, Baseline II, and GRU-M3PCA5 yielded relatively low success rates for stacked objects, with GRU-M3PCA5 achieving an *SRG* of 82% for isolated objects but still underperforming on stacked configurations. In contrast, advanced methods such as ADG-Net I and ADG-Net II show significant improvements—especially when multimodal data are incorporated. For example, multimodal ADG-Net I achieved an *SRGP* of 80% and an *SRG* of 81% for stacked objects, while multimodal ADG-Net II reached an *SRGP* of 92% and an *SRG* of 85%. Compared to the RGB-D-only version of ADG-Net I, the multimodal variants typically increased the grasping *SRG* by 9% for isolated objects and by 20% for stacked objects, while also improving *SRGP* by over 20% in various conditions. These findings clearly demonstrate the advantages of using advanced neural network architectures combined with multimodal data fusion in enhancing robotic grasping, particularly in narrowing the performance gap between isolated and stacked object scenarios. However, despite the notable improvements in prediction accuracy offered by ADG-Net II, its actual grasping success rates did not increase dramatically compared to ADG-Net I. This limitation is partly attributable to the design constraints of the real-world ShadowHand with BioTac sensors, where the first joint of each finger is fixed at 20° to accommodate the sensors, thereby reducing the hand’s overall flexibility. A more detailed discussion of failure cases and error analyses is provided in Section 5.6.3.

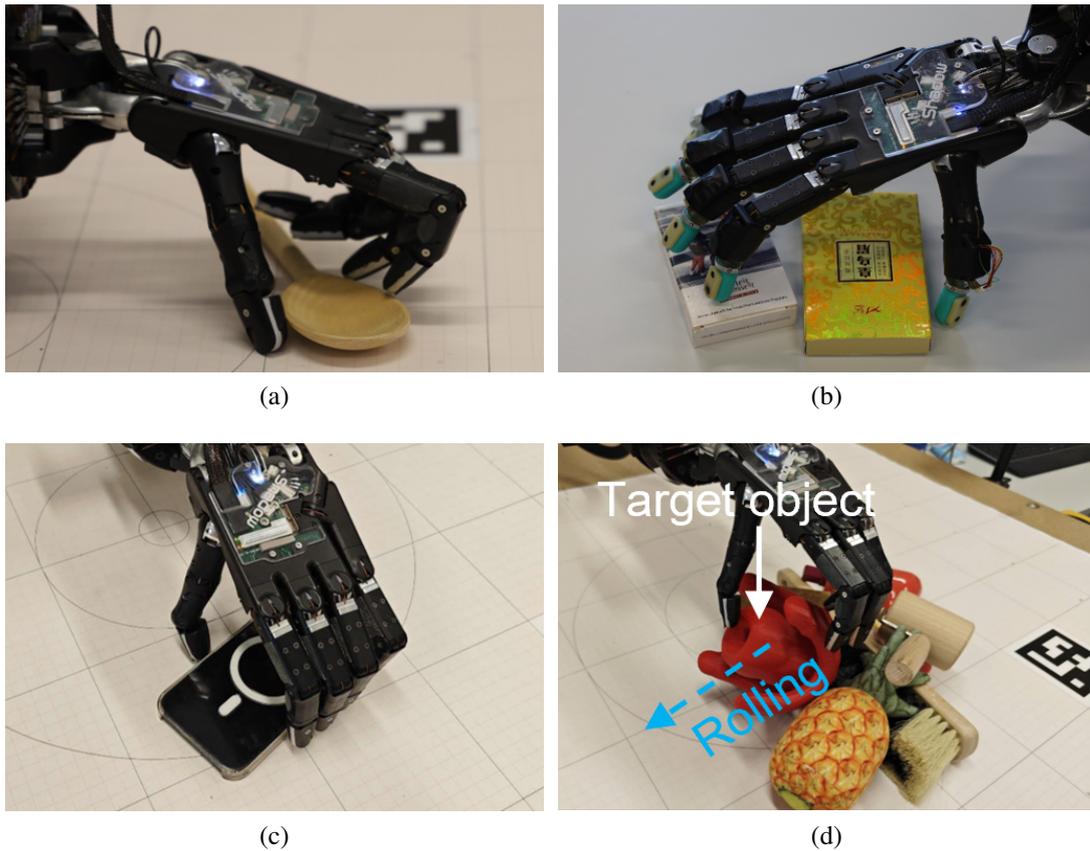


Figure 5.14: Four failed grasping trials. (a) A grasping trial for a slim spoon using the ShadowHand without BioTac sensors. (b) A grasping trial for two neighbor objects using the ShadowHand with BioTac sensors. (c) A grasping trial for a thin and flat mobile phone using the ShadowHand without BioTac sensors. (d) A grasping trial for a poorly-supported object using the ShadowHand without BioTac sensors. [147]

5.6.3 Failed Trials and Limitations

Four typical failure cases were observed during real-world grasping experiments. For instance, Fig. 5.14(a) illustrates an attempt to grasp a slim object that was unsuccessful, while Fig. 5.14(b) depicts a failure when trying to pick up two objects positioned close together. These issues largely stem from inherent limitations in the mechanical design of ShadowHands. Unlike the independently operating thumb joints, the first and second joints of the other fingers—such as FFJ1 and FFJ2—are mechanically coupled and driven by a single motor. In the configuration that uses BioTac sensors, additional constraints are imposed because joints FFJ1, MFJ1, RFJ1, and LFJ1 are fixed at 20° to accommodate the sensors. This restriction limits the hand’s ability to execute subtle, flexible movements required for grasping certain objects, even though ADG-Nets can predict feasible grasp poses and joint configurations, as shown in Fig. 5.13. Improving the dexterity of the ShadowHand remains a promising yet challenging direction for future research [111].

Furthermore, during the experiments, the robot’s movement range was intentionally limited to prevent collisions with the table and potential damage to the ShadowHand. As a result, the ShadowHand was unable to make contact with the table or successfully grasp thin, flat objects placed on it, as evidenced in Fig. 5.14(c). Additionally, Fig. 5.14(d) presents a case where a grasp failed because the target object, being part of a cluttered stack with insufficient support, rolled off the table upon being lightly touched by a finger during approach.

5.7 Summary

In this study, we present a novel Sim2Real multimodal learning framework specifically designed for dexterous grasping. Our approach features a two-stage simulation process that not only replicates the complexities of dexterous grasping but also enables the efficient collection of a comprehensive multimodal grasp dataset. Leveraging this dataset, we developed ADG-Net—a versatile neural network that learns the fundamental principles of dexterous grasping and accurately predicts grasp parameters. The resulting adaptive grasping method reliably identifies feasible grasp configurations, and both simulation and physical experiments demonstrate its promising performance in challenging dexterous grasping tasks.

Chapter 6

Joint Motion and Grasp Planning for HRC

Building on the advances in predictive motion planning and adaptive dexterous grasping presented in the previous chapters, this chapter addresses the critical challenge of planning motion and grasping jointly in dynamic HRC scenarios. Unlike prior approaches that treated reaching and grasping as independent tasks, our integrated framework combines an Actor-Critic RL strategy with a DNLS optimization module to coordinate these actions simultaneously. The RL component generates target poses and dynamically tunes cost function weights, mitigating local optimality issues, while the DNLS module refines these target poses to ensure that kinodynamic and safety constraints are met. This unified approach not only accelerates the training process but also enables the seamless integration of the predictive capabilities and grasping accuracy demonstrated in earlier chapters. Extensive simulations and real-world experiments show that our joint planning framework achieves higher success rates, smoother trajectories, and enhanced safety, highlighting its potential for advancing collaborative robotics in dynamic and safety-critical environments.

6.1 Introduction

In dynamic HRC environments, jointly planning both the robot’s motion and grasping actions is essential for efficient and safe object manipulation. The selection of an appropriate grasp pose critically influences the feasibility of generating a valid trajectory that not only minimizes travel distance but also satisfies stringent safety constraints to avoid collisions with humans and obstacles. While current robotic grasp generation techniques are largely designed for static settings and evaluate grasp candidates through time-consuming processes, few approaches address the simultaneous coordination of motion and grasp planning in dynamic, complex environments.

RL, particularly through the Actor-Critic paradigm [123], offers a powerful solution by combining policy-based and value-based strategies. In this framework, the Critic estimates the long-term expected reward—encompassing both immediate safety considerations (like collision avoidance) and long-term outcomes (such as a successful

grasp)—while the Actor updates its policy to select actions that optimize this balance. However, end-to-end RL methods typically perform well only within the training distribution and struggle to generalize to unpredictable, safety-critical scenarios [110, 131].

Complementing RL, MPC is a robust control strategy that predicts future system behavior over a finite horizon by solving a constrained optimization problem at each time step. Although MPC ensures that control inputs adhere to input, state, and output constraints, its reliance on simplified dynamic models and hand-crafted cost functions means that its solutions are optimal only in a local sense over a short horizon [50].

To overcome these limitations and to unify motion and grasp planning, we propose an integrated framework termed "Actor-Critic Differentiable MPC." In our approach, a differentiable MPC module is embedded as the final layer of the Actor network, thereby leveraging both RL's adaptive policy learning and MPC's constraint-handling capabilities. The RL component generates target joint angles and dynamically adjusts cost function weights to address local optimality challenges, while the differentiable MPC module refines these target poses to ensure that kinodynamic and safety constraints are met. To mitigate the computational burden associated with solving large-scale constrained optimization problems at every time step, we reformulate the MPC problem as an unconstrained DNLS optimization problem [96]. In this formulation, separation planes between the robot and human—computed following the approach in [69]—are incorporated using a hinge loss function to guide the robot away from potential collisions, thereby preserving the convexity of the problem and enabling efficient GPU acceleration.

At each time step, our integrated Actor-Critic framework uses current observations to update both the target joint angles and the DNLS cost function weights, ensuring that the resulting trajectory is both collision-free and compliant with the robot's kinodynamic constraints. Extensive simulations and real-world experiments demonstrate that this joint motion and grasp planning approach not only improves task success rates and trajectory smoothness but also enhances overall safety in dynamic human-robot shared workspaces.

The remaining sections are organized as follows. In Section 6.2, we present the architecture of our proposed approach. In Section 6.3, we discuss the experimental results that highlight the effectiveness of our framework. Finally, we conclude with a discussion and suggest directions for future work in Section 6.4.

6.2 Methodology

In this section, we provide an in-depth discussion of our system's key components—the DNLS module and the RL module—and explain how they are integrated to function cohesively. We begin by outlining the theoretical basis and implementation details of the DNLS module, which reformulates the traditional MPC problem into an unconstrained nonlinear least squares framework. Next, we describe the Actor-Critic RL module, which is responsible for generating target joint angles and dynamically adjusting the cost function weights to guide the DNLS optimization toward safe, feasible trajectories. Finally, we detail our integration strategy that ensures both modules work seamlessly

in real time, allowing the overall system to adapt effectively to dynamic human–robot interactions while satisfying all kinodynamic and safety constraints.

6.2.1 General Formulation of MPC

MPC relies on a mathematical model to forecast the system’s future behavior over a fixed, finite time horizon. Using these predictions, MPC computes a sequence of control actions that optimize a given cost function while adhering to various constraints—such as limits on system states, control inputs, and safety requirements. Crucially, at each timestep, only the first control action from this sequence is executed. The process is then repeated with updated state information in a receding horizon manner, ensuring that the control strategy remains optimal as the system evolves.

Mathematically, MPC solves an optimization problem to determine the optimal control $u = (u_0, \dots, u_{N-1})$ and corresponding system states $x = (x_0, \dots, x_N)$ by solving a nonlinear optimization problem:

Mathematically, MPC is formulated as a nonlinear optimization problem as Eq. 6.1 that aims to determine the optimal sequence of control inputs, $u = (u_0, u_1, \dots, u_{N-1})$, along with the corresponding state trajectory, $x = (x_0, x_1, \dots, x_N)$. This optimization minimizes a specified cost function while ensuring that the system dynamics and all constraints—such as state, control, and safety limits—are satisfied.

$$\begin{aligned}
 & \min_{x,u} \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\
 \text{s.t. } & x_0 = \bar{x}_0 \\
 & x_{k+1} = f(x_k, u_k) \\
 & 0 \geq h_i(x_k, u_k), \quad i = 1, \dots, m \\
 & 0 = g_j(x_k, u_k), \quad j = 1, \dots, p
 \end{aligned} \tag{6.1}$$

In Eq. 6.1, the state at time step k is represented by $x_k \in \mathcal{X}$ and the control input by $u_k \in \mathcal{U}$, where \mathcal{X} and \mathcal{U} denote the continuous state and input spaces, respectively. The stage cost for non-terminal states is defined by a function $l : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$, while the terminal cost is given by $E : \mathcal{X} \rightarrow \mathbb{R}$. The system begins at an initial state \bar{x}_0 , and its evolution is governed by the dynamic model $f : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{X}$. In addition, the optimization problem incorporates constraint functions $h_i : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ and $g_j : \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ to enforce the required restrictions on states and control actions.

6.2.2 Actor-Critic RL

Actor-Critic methods are a hybrid class of reinforcement learning algorithms that blend the strengths of both value-based and policy-based approaches. In these methods, the Actor is responsible for selecting actions by following a policy—typically represented as a probability distribution over actions conditioned on the current state s , denoted by $\pi(a | s, \theta)$, where θ are the policy parameters. The Critic, on the other hand, evaluates the actions chosen by the Actor by estimating a value function, which can take the form

of either the state-value function $V(s, \mathbf{w})$ or the action-value function $Q(s, a, \mathbf{w})$, with \mathbf{w} representing the value function parameters. Both sets of parameters, θ and \mathbf{w} , are iteratively updated using gradient descent to minimize appropriate loss functions, as detailed in Algorithm 1 [123].

Algorithm 1 One-step Actor–Critic (episodic), for estimating $\pi_\theta \approx \pi^*$

Input: a differentiable policy parameterization $\pi(a|s, \theta)$
Input: a differentiable state-value function parameterization $\hat{v}(s, \mathbf{w})$
Parameters: step sizes $\alpha^\theta > 0, \alpha^\mathbf{w} > 0$
 Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$
 Loop forever (for each episode):
 Initialize S (first state of episode)
 $I \leftarrow 1$
 Loop while S is not terminal (for each time step):
 $A \sim \pi(\cdot|S, \theta)$
 Take action A , observe S', R
 $\delta \leftarrow R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$
 $\mathbf{w} \leftarrow \mathbf{w} + \alpha^\mathbf{w} \delta \nabla_{\mathbf{w}} \hat{v}(S, \mathbf{w})$
 $\theta \leftarrow \theta + \alpha^\theta I \delta \nabla_{\theta} \ln \pi(A|S, \theta)$
 $I \leftarrow \gamma I$
 $S \leftarrow S'$
 if S' is terminal, then $\hat{v}(S', \mathbf{w}) \doteq 0$

6.2.3 Integrating Differentiable MPC with RL

In Eq. 6.1, if the constraint functions $(h_i)_{i=1}^m$ and $(g_j)_{j=1}^p$ are linear with respect to the state x and control u , and if both the stage cost ℓ and the terminal cost E are quadratic functions of x and u , then the overall nonlinear optimization problem simplifies to a QP formulation, as shown in Eq. 6.2. In this QP problem, the state x_k belongs to a continuous subset $\mathcal{X} \subseteq \mathbb{R}^n$ and the control input u_k belongs to a continuous subset $\mathcal{U} \subseteq \mathbb{R}^l$. Additionally, $\mathbf{Q}_k \in \mathbb{R}^{(n+l) \times (n+l)}$ represents the quadratic cost matrix, $\mathbf{p}_k \in \mathbb{R}^{n+l}$ is the corresponding linear cost vector, $\mathbf{A}_k \in \mathbb{R}^{(m+p) \times (n+l)}$ is the matrix defining the linear constraints, and $\mathbf{b}_k \in \mathbb{R}^{m+p}$ is the associated constraint vector.

$$\begin{aligned} \min_{x, u} \quad & \sum_{k=0}^N \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top \mathbf{Q}_k \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + \mathbf{p}_k^\top \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \\ \text{s.t.} \quad & \mathbf{A}_k \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \preceq \mathbf{b}_k \end{aligned} \tag{6.2}$$

QP is a convex optimization method that efficiently solves problems using off-the-shelf solvers, yet its optimal solution is highly sensitive to the parameters \mathbf{Q}_k , \mathbf{p}_k , and \mathbf{A}_k . Finding the ideal values for these parameters is particularly challenging in complex,

dynamically changing tasks. Drawing inspiration from [3, 109, 110, 120], our approach employs RL to tune these parameters. Specifically, we incorporate a differentiable MPC module as the final layer of the Actor network, which takes as inputs $\mathbf{Q}_{\theta,k}$, $\mathbf{p}_{\theta,k}$, and $\mathbf{A}_{\theta,k}$ produced by earlier layers. This module outputs the optimal control $\mathbf{u}_{\theta,k}^*$ for the QP defined in Eq. 6.2, and its gradients with respect to the input parameters can be computed analytically [3]. However, training such a differentiable QP layer with RL is computationally intensive, as each forward and backward pass requires solving a large batch of optimization problems—making the computation time directly proportional to the batch size due to limited GPU acceleration for QP solvers.

To address this bottleneck, we reformulate the QP as an unconstrained DNLS optimization problem (see Eqs. 6.3 and 6.4). Our DNLS optimizer, built on PyTorch [94], leverages advanced features including second-order gradient computation, automatic vectorization, batching, and full GPU acceleration. These capabilities enable efficient optimization even with large batch sizes. The DNLS optimizer computes the gradients of the optimal solution $\mathbf{u}^*(\theta)$ with respect to the parameters θ from preceding layers, as detailed in [96]. Once the optimal control $\mathbf{u}^*(\theta)$ is determined and forwarded to the robot, we compute the overall Actor-Critic loss $L(\mathbf{u}^*(\theta))$. Using the chain rule, we then obtain the gradient of this loss with respect to θ and update the network parameters accordingly, as described in Eq. (6.5).

$$\mathbf{u}^*(\theta) = \arg \min_{\mathbf{u}} S(\mathbf{u}; \theta) \quad (6.3)$$

$$S(\mathbf{u}; \theta) = \sum_{k=0}^N \left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix}^\top \mathbf{Q}_{\theta,k} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} + \mathbf{p}_{\theta,k}^\top \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \right) + ELU \left(\mathbf{A}_k \begin{bmatrix} \mathbf{x}_k \\ \mathbf{u}_k \end{bmatrix} \preceq \mathbf{b}_k \right) \quad (6.4)$$

$$\theta^* = \arg \min_{\theta} L(\mathbf{u}^*(\theta)) \quad (6.5)$$

$$c_g = \sum_{k=0}^N \|w_{\theta,g}(\mathbf{q}_k - \mathbf{q}_{\theta,g})\|^2, \quad (6.6)$$

$$c_{init} = \|w_{init}(\mathbf{q}_k - \mathbf{q}_{init})\|^2, \quad (6.7)$$

$$c_q = \sum_{k=0}^N \|w_q ELU(\mathbf{q}_k - \mathbf{q}_{max})\|^2 + \|w_q ELU(\mathbf{q}_{min} - \mathbf{q}_k)\|^2, \quad (6.8)$$

$$c_v = \sum_{k=0}^{N-1} \|w_v ELU(\mathbf{q}_{k+1} - \mathbf{q}_k - \mathbf{v}_{max} d_t)\|^2 + \|w_v ELU(\mathbf{q}_k - \mathbf{q}_{k+1} - \mathbf{v}_{max} d_t)\|^2, \quad (6.9)$$

$$c_{ws} = \sum_{k=0}^N \sum_{i=1}^6 \sum_{j=0}^4 \|w_{ws} ELU(\mathbf{n}_i^\top (\mathbf{p}_{j,init} + \mathbf{J}_j(\mathbf{q}_k - \mathbf{q}_{init})) - b_i)\|^2, \quad (6.10)$$

$$c_h = \sum_{k=0}^N \sum_{i=1}^4 \sum_{j=1}^4 \|w_{\theta,h} ELU(-\mathbf{n}_i^\top (\mathbf{p}_{j,init} + \mathbf{J}_j(\mathbf{q}_k - \mathbf{q}_{init})) + b_i)\|^2, \quad (6.11)$$

$$S(\mathbf{q}; \theta) = c_g + c_{init} + c_{ang} + c_{vel} + c_{ws} + c_h. \quad (6.12)$$

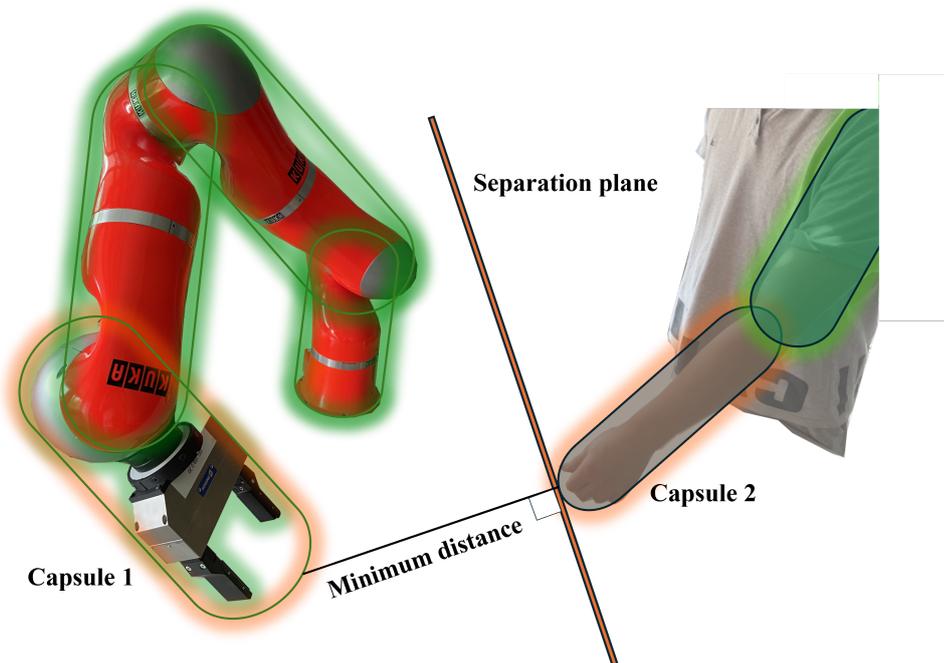


Figure 6.1: Separation plane. For each pair of human and robot arm links—both represented as capsules (e.g., capsule 1 and capsule 2)—we compute the minimal distance between them along with their respective closest points, following the method in [69]. Based on these computations, a separation plane is established that passes through the closest point on capsule 2. The normal vector of this plane is oriented from this point on capsule 2 toward the corresponding closest point on capsule 1. By enforcing the constraint that capsule 2 must remain on one side of this plane, potential collisions with capsule 1 are effectively prevented.

6.2.4 Dynamic Grasp in Shared Human-Robot Workspace

In our collaborative workspace, the human performs pick-and-place tasks while the robot is responsible for grasping objects on the table—all while ensuring collision-free interactions. In this framework, the Actor network’s preceding layer outputs two essential components: a set of cost weights $w_i \in \mathbb{R}$ and a target joint angle vector $\mathbf{q}_{\theta,g} \in \mathbb{R}^n$ for robot tracking. These outputs form the basis for constructing a composite cost function $S(\mathbf{q}; \theta)$, which is the sum of several individual costs. Specifically, this includes a goal tracking cost c_g (see Eq. 6.6), an initial state constraint cost c_{init} (Eq. 6.7), joint angle and velocity constraint costs c_q and c_v (Eqs. 6.8 and 6.9), as well as a workspace constraint cost c_{ws} and a human avoidance cost c_h (Eqs. 6.10 and 6.11). In these constraints, the workspace and human avoidance terms are defined using the normal vectors \mathbf{n}_i and offsets b_i of the respective planes. While c_g and c_{init} are modeled as quadratic functions of the trajectory $(\mathbf{q}_k)_{k=0}^N$, the remaining cost terms are formulated as hinge functions. Owing to this structure, the overall cost function $S(\mathbf{q}; \theta)$ is convex with respect to the joint trajectory, enabling its global optimum to be efficiently computed using our DNLS optimizer.

$$\mathbf{n}^\top \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{n}^\top \mathbf{P}_h = b \quad (6.13)$$

$$\mathbf{n}^\top (\mathbf{p}_{j,init} + \mathbf{J}_i(\mathbf{q}_k - \mathbf{q}_{init})) \geq b, \quad j = 1, 2 \quad (6.14)$$

The cost terms c_{ws} and c_h are designed to confine the robot’s movements within the specified workspace and prevent collisions with human operators. Following the approach in [69], both human arms and robot links are modeled as capsules for efficient collision detection, as illustrated in Fig. 6.1. Each capsule, defined by a cylinder capped with hemispheres at both ends, connects two adjacent joints along the kinematic chain and has a radius r . For each pair of human and robot capsules, the closest points \mathbf{P}_r (on the robot) and \mathbf{P}_h (on the human) are computed [122]. A separating plane is then defined to pass through \mathbf{P}_h with its normal vector \mathbf{n} directed from \mathbf{P}_h toward \mathbf{P}_r . To ensure collision avoidance, the center of the robot capsule is required to lie above this plane, as expressed in Eq. 6.14, which is reformulated into a hinge loss function in Eq. 6.11. Workspace constraints are similarly formulated in Eq. 6.10. The entire Actor-Critic DNLS algorithm, which integrates these constraints into the control framework, is detailed in Algorithm 2.

Algorithm 2 Actor-Critic DNLS

Input: a DiffNLS policy parameterization $\text{DiffNLS}(a|s, \theta)$

Input: a differentiable state-value function parameterization $\hat{v}(s, \mathbf{w})$

Parameters: step sizes $\alpha^\theta > 0, \alpha^\mathbf{w} > 0$

Initialize policy parameter $\theta \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^d$ (e.g., to 0)

Loop forever (for each episode):

 Initialize S (first state of episode)

$I \leftarrow 1$

Loop while S is not terminal (for each time step):

$A \sim \text{DiffNLS}(\cdot|S, \theta)$

 Take action A , observe S', R

$\delta \leftarrow R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$

$\mathbf{w} \leftarrow \mathbf{w} + \alpha^\mathbf{w} \delta \nabla_{\mathbf{w}} \hat{v}(S, \mathbf{w})$

$\theta \leftarrow \theta + \alpha^\theta I \delta \nabla_{\theta} \ln \text{DiffNLS}(A|S, \theta)$

$I \leftarrow \gamma I$

$S \leftarrow S'$

if S' is terminal, then $\hat{v}(S', \mathbf{w}) \doteq 0$

Details about Actor-Critic

Our observation space is constructed from a range of features, including the robot’s 9-DoF joint angles and corresponding joint velocities, the Cartesian coordinates of both grippers, the Cartesian pose of the robot’s tip link, the positions and velocities of the

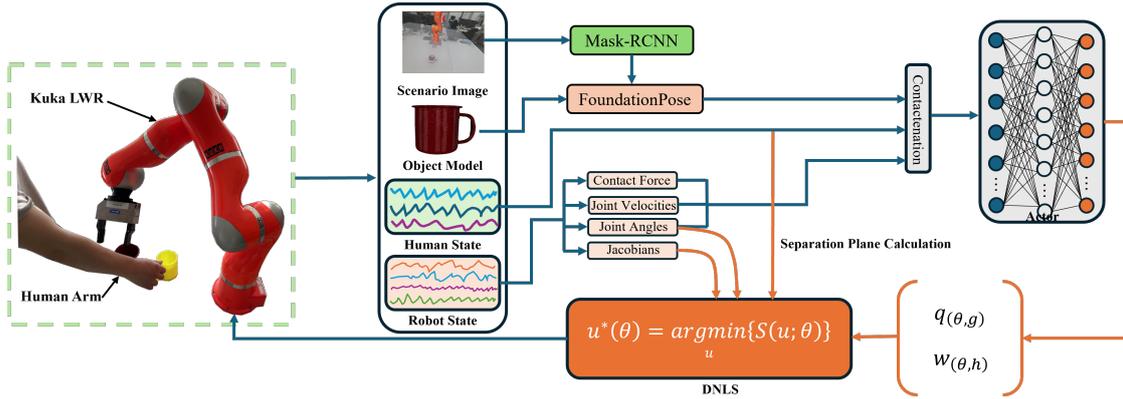


Figure 6.2: Method overview. FoundationPose [137] is used to estimate the Cartesian pose of the target object from its RGB image and 3D model. This estimated pose is then combined with the robot’s kinematic data to form a comprehensive input for the Actor module. The Actor processes this combined information and produces two outputs: the target joint angles and a set of cost function weights that will parameterize the subsequent DNLS module. The DNLS module—configured with these cost weights, the target joint angles, the robot’s current joint configuration, the relevant Jacobians, and the positions of the human arm—solves an optimization problem to compute the optimal control inputs. These optimal control inputs specify the desired joint angles for the robot in the next time step, thereby ensuring coordinated and safe joint motion and grasp planning.

human arm’s three key joints (wrist, elbow, and shoulder), and the Cartesian pose of the object. Although the gripper positions and tip link pose are redundant when considering the 9-DoF joint angles, we have found that incorporating this additional information helps accelerate the learning process.

The action space comprises the following components: the 7-DoF joint velocity commands for the robot arm, a 1-DoF velocity command for the parallel gripper for the upcoming time step, and an 8-dimensional weight vector used to modulate the robot–human collision avoidance cost functions. The action values for both the robot arm and gripper are constrained within the range $(-1.0, 1.0)$ and are scaled by a factor of 7.5 before being applied to the robot, while each element of the collision avoidance weight vector is clamped to the interval $(0, 1.0)$.

For object placement, the target object is randomly positioned on a table with its x -coordinate sampled from the interval $(0.3, 0.6)$ and its y -coordinate from $(-0.5, 0.5)$. The simulation runs with a time step of $1/60$ seconds, ensuring high-frequency updates for both observation and control.

The reward function R for the dynamic grasp task is composed of four components: a distance reward R_{dist} , a reward for successfully gripping the object R_{grip} , a reward for lifting the object R_{lift} , and a collision avoidance reward R_{coll} .

$$R = c_1 R_{dist} + c_2 R_{grip} + c_3 R_{lift} + c_4 R_{coll}. \quad (6.15)$$

where:

$$R_{dist} = \|P_{eef} - P_{obj}\| \quad (6.16)$$

$$R_{grip} = \mathbb{1}(\text{object_gripper_contact}) \quad (6.17)$$

$$R_{lift} = \mathbb{1}(\text{object_height} \geq 0.2) \quad (6.18)$$

$$R_{coll.} = \mathbb{1}(\text{robot_table_human_contact}) \quad (6.19)$$

Actor-Critic DNLS Framework

Figure 6.2 illustrates our proposed Actor-Critic DNLS framework. In this approach, an RGB image and the corresponding 3D model of an object are first processed by FoundationPose [137] to estimate the object’s Cartesian pose. This estimated pose is then combined with the robot’s kinematic data to form a unified input for the Actor network. The Actor subsequently outputs target joint angles and a set of cost function weights, which are passed to the DNLS module. Finally, the DNLS module solves an optimization problem to determine the optimal control commands that drive the robot’s movements, ensuring that both motion planning and grasp execution are performed efficiently and safely.

Environments	KL Threshold	Mini-batch Size	Horizon Length	PPO Epochs	Hidden Units	Value Loss Coef.	Learning Rate
8192	0.01	131072	128	8	128, 64, 32	2.0	Adaptive

Table 6.1: PPO Hyperparameters. The activation function used is ELU, and the learning rate is adapted with a KL threshold of 0.01.

6.3 Experiments

We first train the policy in a simulated environment. The hyperparameters for PPO training are summarized in Table 6.1, with the reward function coefficients set as $c_1 = 1$, $c_2 = 1$, $c_3 = 20$, and $c_4 = -5$. To assess the effectiveness of our method, we perform evaluations both in simulation and on a physical robot. For these experiments, we selected three objects from the YCB dataset [13]—a mustard bottle, a banana, and a mug—chosen to represent a range of shapes and sizes. All training and evaluation were conducted on an NVIDIA RTX A6000 GPU.

6.3.1 Simulation Evaluation

Simulation Setup

We employ IsaacLab [82] to construct a physical simulation environment that mimics a real-world setup, featuring a Kuka LWR robotic arm fitted with a WSG-50 parallel jaw gripper, an upper-body human model, and a YCB object, as depicted in Fig. 6.3. In this environment, the human torso remains stationary while only the arm is allowed to move.

Agent	Effort Limit	Velocity Limit	Stiffness	damping
Kuka LWR	87 N · m	2.269 rad s ⁻¹	100	10
WSG-50	50 N · m	0.2 rad s ⁻¹	500	50

Table 6.2: Kinodynamic Parameters for Kuka and WSG-50 in simulation. The velocity limits for the Kuka LWR and WSG-50 are set to 1 *rad/s* and 0.2 *rad/s*, respectively.

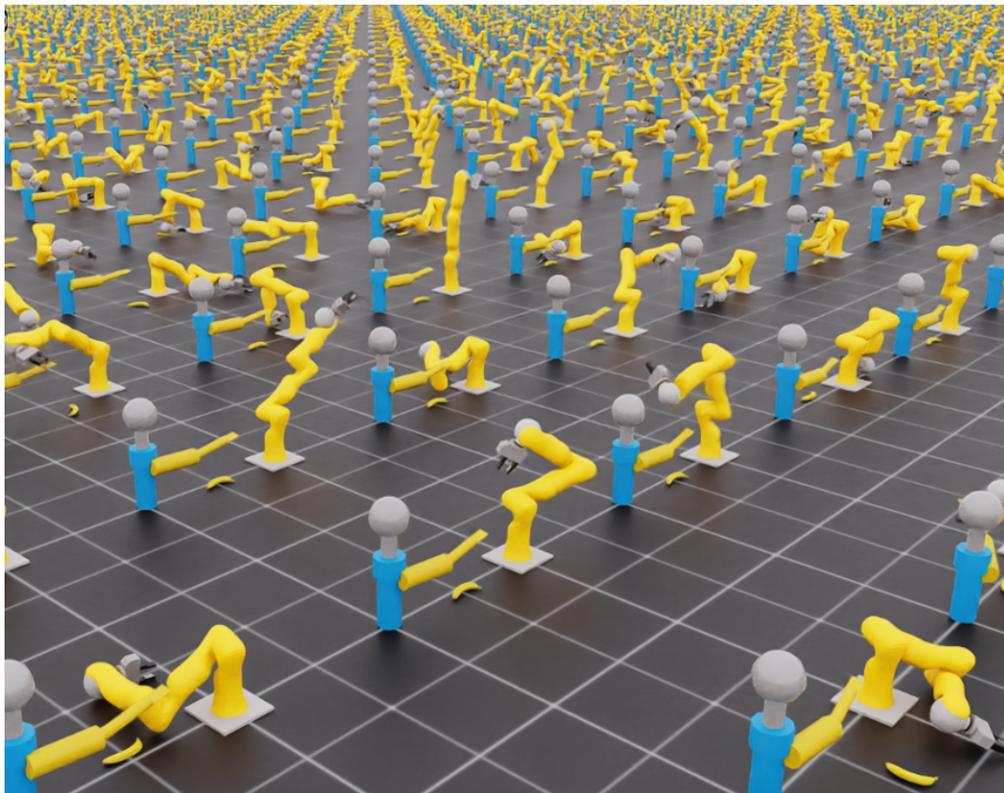


Figure 6.3: Learning dynamic grasp in human-robot shared workspace with massively parallel simulation environments. The human trunk is modeled with a blue cylinder; the human arm is modeled with three yellow cylinders; the human neck and head are modeled with a grey cylinder and sphere, separately. The robot is a Kuka LWR robot arm with a WSG-50 parallel jaw gripper.

To bridge the sim-to-real gap, we incorporate domain randomization during training. Specifically, at each reset of the simulation, the lengths of the human arm segments are perturbed using Gaussian distributions: the upper arm length is sampled from $\mathcal{N}(\mu = 0.25\text{ m}, \sigma^2 = 4)$ and the lower arm length from $\mathcal{N}(\mu = 0.3\text{ m}, \sigma^2 = 4)$. To emulate human arm motion, three shoulder joints and one elbow joint are actively controlled, with their velocities at each simulation step drawn from a Gaussian distribution $\mathcal{N}(\mu = 1.0\text{ m/s}, \sigma^2 = 2)$. When any joint reaches its angular limit, its velocity is instead re-sampled from $\mathcal{N}(\mu = -1.0\text{ m/s}, \sigma^2 = 2)$. This strategy improves the robustness of our policy by exposing it to a range of realistic variations. The kinodynamic parameters for the robot arm and gripper used in these simulations are detailed in Table 6.2.

		success (%)	mean grasp time (s)	failure (%)			
				contact	drop	timeout	total
Sim.	Yang <i>et al.</i> [144]	72	10.2	0	12	16	28
	Ours	88	4.9	0	6	6	12

Table 6.3: Evaluation of Grasping Performance in IsaacLab Simulation. Comparison of our proposed method with the baseline method from Yang *et al.* [144]. The table highlights the success rate, mean grasp time, and failure rates under different failure categories (contact, drop, timeout). Our method achieves a significantly higher success rate and reduced mean grasp time, which can be attributed to the integration of the global planner (RL) and the local optimization planner (DNLS).

Metrics

An object-grasping trial is considered successful if the robot lifts the object to a height of 20 cm above the ground. Conversely, an episode is marked as a failure and terminated if the object collides with the ground or the human body, or if the maximum episode duration is reached. For each object, 100 grasp attempts are conducted. In every attempt, the object’s position is randomly initialized within a predefined region, and the robotic arm begins the grasp from its starting position. Throughout the grasping process, the human arm moves in a manner consistent with the patterns observed during training. To evaluate our method’s performance, we measure and analyze the grasp success rate, the trajectories of both the robot and the human, the number of collisions, and the total grasping time.

Baseline

Yang *et al.* [144] is employed as our baseline method. Although originally designed for human–robot handover tasks, their approach was adapted for our dynamic object-grasping scenario. In our version, instead of the object being held by a human, it is placed on a table, and the obstacle representation is modified—from a cylinder positioned between the camera and the hand to the actual links of a human’s right arm. Since human arm movements in our experiments occur independently of the object’s pose and exhibit greater randomness than those considered in [144], the obstacle avoidance challenge during grasping becomes significantly more complex in our task.

Results

In the IsaacLab simulation, we compared our approach against the baseline method described in [144]. As summarized in Table 6.3, our method consistently achieves a higher success rate while also significantly reducing the overall task completion time.

This performance improvement stems from the integration of a global planner—implemented via reinforcement learning (RL)—with a local planner based on Differentiable Nonlinear Least Squares (DNLS). The RL module generates globally optimal target joint angles and cost function weights, which the DNLS module then refines

	suc. (%)	fail. (%)				
		contact w/ human	contact w/ ground	drop	timeout	total
w/o human; w/ RL	86	0	7	2	5	14
w/ human; w/ RL	77	2	9	3	9	23
w/o human; w/ RL&DNLS	93	0	0	3	4	7
w/ human; w/ RL&DNLS	88	0	0	4	8	12

Table 6.4: Ablation Study. This table presents the success rates (succ.%) and failure modes (fail.%) for object-grasping tasks under various conditions, including the presence or absence of a human, using RL alone, and RL combined with DNLS. Failure modes are categorized into contact with the human, contact with the ground, object drops, and task timeouts. All results are averaged over 100 trials per condition.

to produce efficient, collision-free trajectories. This synergy not only minimizes timeouts but also results in shorter grasp paths and faster grasping times. By contrast, the baseline approach selects target poses from a precomputed set of grasp candidates [86] without considering the robot’s reachability. As a result, the baseline often chooses grasp poses that are not physically attainable, leading to more frequent timeouts and object drops. Additionally, determining suitable cost function weights in the baseline method is both complex and time-intensive, further contributing to longer trajectories and lower grasp success rates. A demonstration of the grasping process using our method is shown in Fig. 6.4.

Fig. 6.5 presents the robot’s configurations along with their associated value maps throughout the grasping process. When the robot is at its initial position (Fig. 6.5a), the value map shows low values, reflecting the distance from the final grasp target. As the robot moves closer to the object (Fig. 6.5b), the value map transitions from green to yellow, indicating an increase in value. Once the robot successfully grasps the object, the value map shifts from red (Fig. 6.5c) to dark red (Fig. 6.5d), signaling the attainment of the highest value. This progression effectively illustrates the robot’s advancement from initial approach to successful grasp execution.

Ablation Study

We performed an ablation study to examine the effectiveness of our Actor-Critic DNLS framework on object-grasping performance. In particular, we evaluated four configurations: (1) grasping using only RL without human presence; (2) grasping using only RL with human presence; (3) grasping using the combined RL and DNLS approach without human presence; and (4) grasping using the combined RL and DNLS approach with human presence. The results, summarized in Table 6.4, reveal that incorporating DNLS with RL leads to significant performance gains.

Specifically, in scenarios without a human, the success rate improved from 86% to 93% when DNLS was added, completely eliminating ground contact failures. Similarly, in scenarios with human involvement, the combined method increased the success rate from 77% to 88% while entirely preventing collisions with both the human and the ground. Timeout and drop failures remained roughly consistent across all configurations. These observations indicate that the DNLS module notably enhances the safety of the

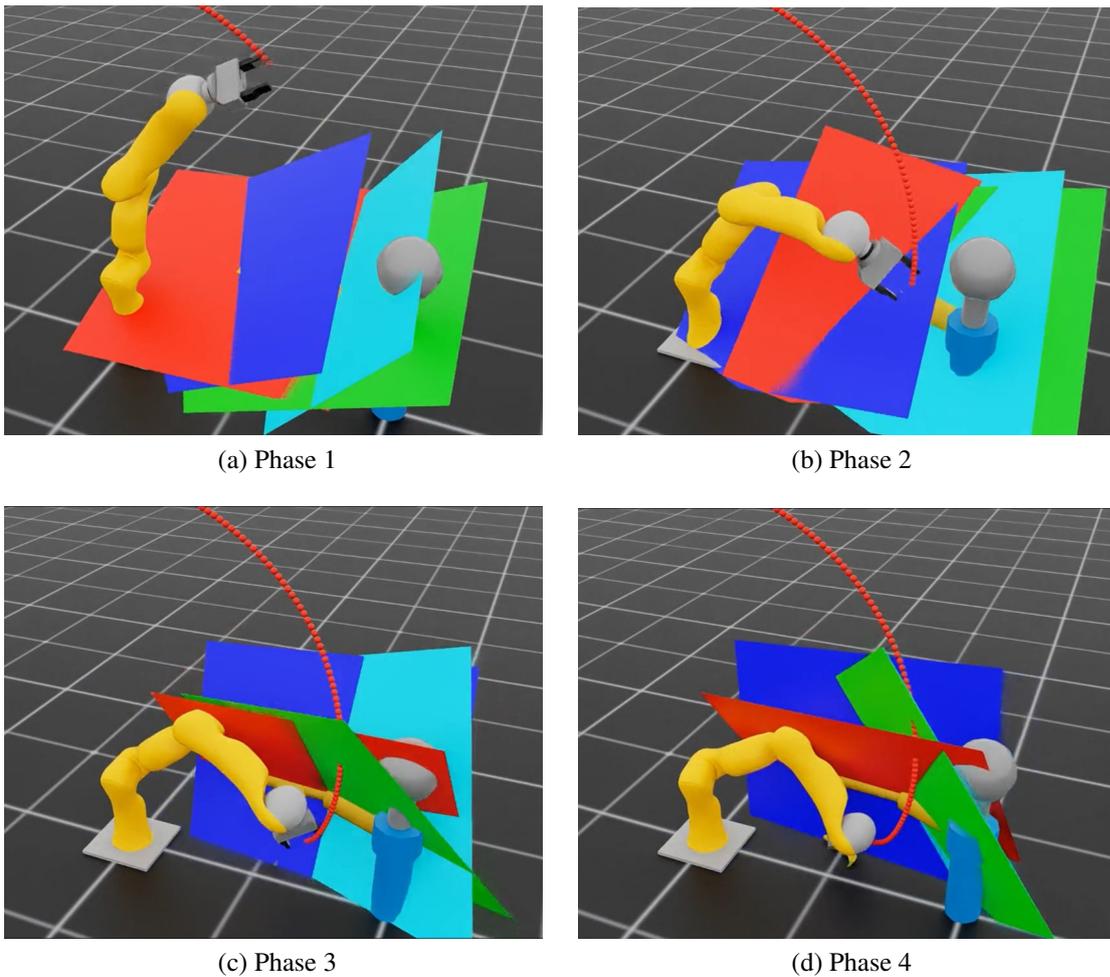


Figure 6.4: Robot motion in different grasp phases. Red spheres trace the robot’s trajectory, while various colored planes act as hard constraints that prevent the robot from colliding with the human arm. For example, the red plane delineates the boundary between the gripper’s capsule and the capsule representing the human’s lower arm, while the green plane separates the gripper capsule from the capsule of the human’s upper arm. Additionally, the blue plane isolates the robot’s second capsule from the human’s lower arm, and the light blue plane does the same for the robot’s second capsule relative to the human’s upper arm. In the depicted scenario, a banana is placed on the ground, and as the human arm moves between the robot and the banana, the robot must plan its grasping motion to successfully pick up the object while avoiding any collisions with the human arm.

grasping policy, particularly in environments where human interaction raises the risk of contact errors.

Although the integrated approach significantly improves overall performance, the persistence of timeout failures across all scenarios suggests that further optimization is needed. Overall, these findings underscore the importance of combining DNLS with RL to achieve safer, more robust object grasping in collaborative human–robot settings.

Fig. 6.6 displays the mean trajectory jerk magnitude per episode and the time cost per

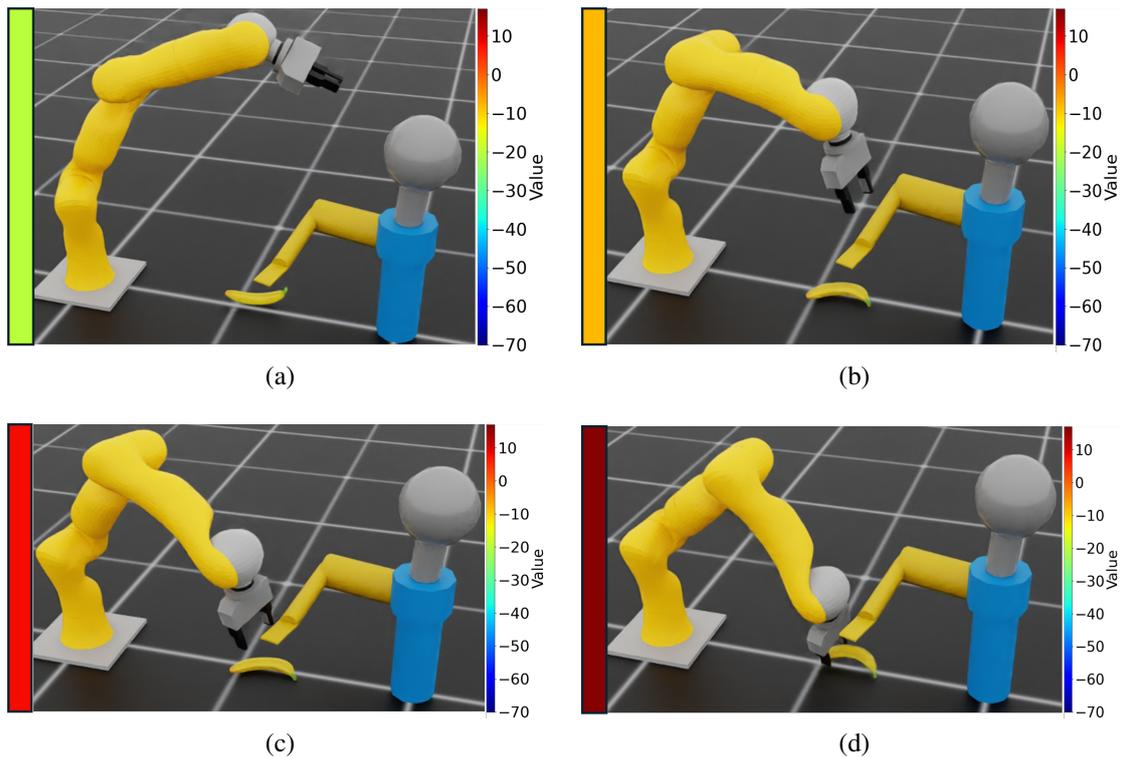


Figure 6.5: The images above depict the value maps generated by the critic network for various robot configurations, representing the expected return for each state. The color scale indicates the magnitude of the value, with dark red corresponding to the highest values and blue to the lowest values. As the robot approaches the object, the value increases, as reflected by the color changes.

step for both RL and RL combined with DNLS. The data clearly show that integrating DNLS with RL results in lower median jerk values and less variability, as evidenced by a smaller interquartile range (IQR) compared to using RL alone. This reduction in jerk indicates that trajectories generated with DNLS are smoother and more consistent—an important factor for ensuring predictable and safe human–robot interactions. However, this improvement in smoothness comes at the expense of increased computational time per step. While the RL-only approach is highly efficient, with a per-step cost of roughly 0.0001 s , incorporating DNLS introduces additional computational overhead. Thus, the trade-off is between faster processing and the production of smoother, safer robot motions.

6.3.2 Real-World Experiments

Our experimental configuration employs two dedicated computers to operate the entire system. One computer is directly connected to the KUKA LWR robot’s control box, which is equipped with a WSG-50 parallel jaw gripper. The second computer, outfitted with two NVIDIA RTX A6000 GPUs, is responsible for executing the trained policy as well as running the object pose estimation and human body tracking modules. Data

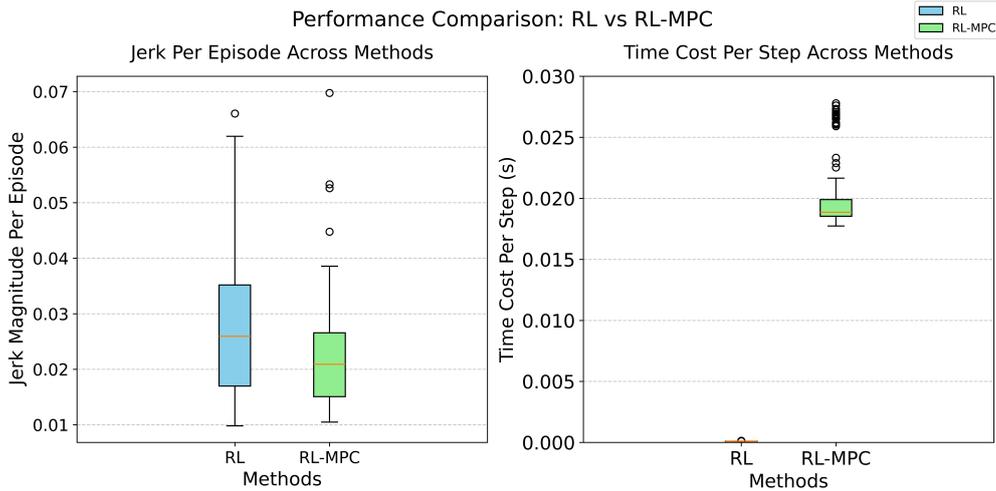


Figure 6.6: The left plot illustrates the jerk magnitude per episode for the RL and RL with DNLS methods, highlighting the smoother trajectories achieved with DNLS. The right plot compares the time cost per step, showing the increased computational overhead introduced by the DNLS module.

exchange between these two systems is facilitated by ROS [104], ensuring seamless communication. In this setup, the robot’s state information, the object’s pose, and human joint data are concatenated to form the input to the policy network, while a ROS-based controller handles the low-level trajectory tracking. Fig. 6.7 provides an overview of the robot setup, and the following sections offer detailed descriptions of each module.

Object Pose Estimation

The object pose estimation module is based on the unified model presented in [137], which supports both model-based and model-free 6D object pose estimation and tracking. This approach can generalize to novel objects at test time without requiring fine-tuning, as long as a CAD model or a few reference images are available.

In our experiments, we employ an Azure Kinect Camera [79] to capture the task scenarios. First, Mask R-CNN [39] is used to detect and segment the object from the background in the acquired images. The resulting RGB image of the segmented object is then input to the pose estimation network, which computes the object’s 6D pose relative to the camera coordinate system. This estimated pose is subsequently transformed into the robot base link frame for use in subsequent modules.

Fig. 6.8 visualizes the performance of the object pose estimation module. Using the predicted pose and the corresponding object mesh file, we generate a colored point cloud that overlays the object. The close alignment between this point cloud and the actual object demonstrates the accuracy and reliability of our pose estimation framework, thereby confirming its suitability for integration with downstream components in our system.

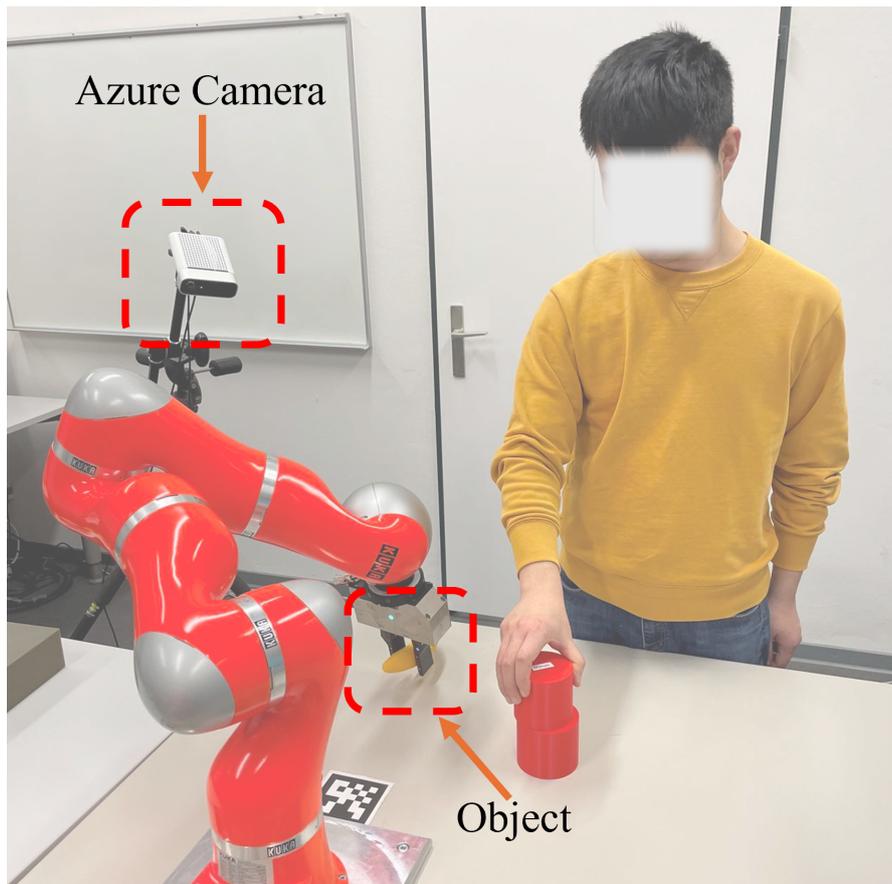


Figure 6.7: Experiment setup. An Azure camera is deployed to track human body movements and estimate the pose of objects on the table. The robot is programmed to grasp these objects while actively avoiding contact with the human arm. To accurately determine the camera's position and orientation, an AprilTag is affixed to the table.

Human Body Tracking

MediaPipe [1] is an open-source platform developed by Google that enables the creation of cross-platform, customizable machine learning solutions for live and streaming media. Its component, MediaPipe Pose, is tailored for precise body pose estimation, extracting 33 three-dimensional landmarks and producing a background segmentation mask for the entire body from RGB video frames. Thanks to its low computational overhead, it supports real-time processing even on mobile devices. An example of the output from MediaPipe Pose is shown in Fig. 6.9. Additionally, human joint velocities are computed by taking the time derivatives of the joint position data.

Results

Table 6.5 summarizes the success and failure rates for grasping three different objects, each evaluated over 10 trials. The results reveal that performance varies with object type: the banana and the mustard bottle achieved success rates of 80% and 90%, respectively, while the mug only reached a 70% success rate, leading to an overall success rate of

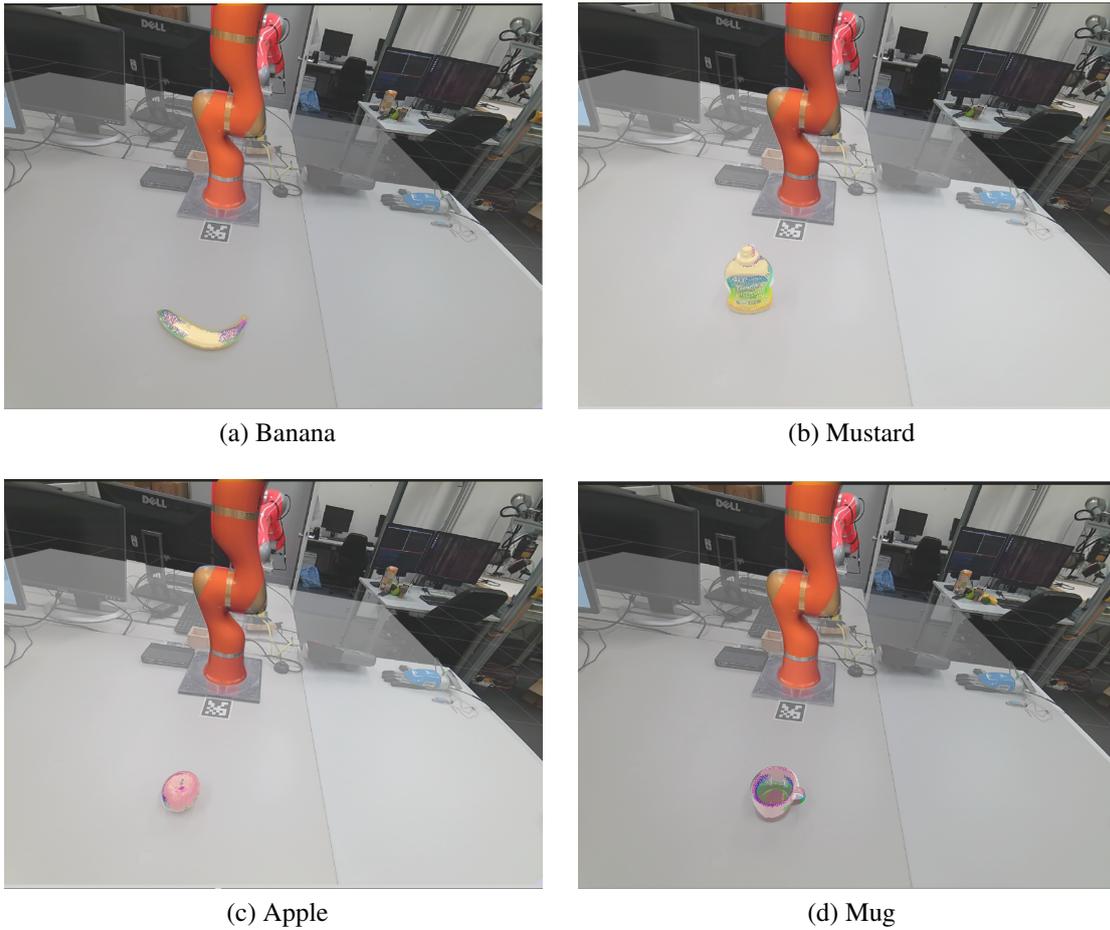


Figure 6.8: The images above illustrate the object pose estimation accuracy for four randomly selected YCB objects. The colored point cloud represents the predicted object pose.

80% across all objects. Failure modes were classified into three categories: contact with the ground, object drops, and timeouts. For the banana, there were 2 failures (1 drop and 1 timeout); the mustard bottle experienced 1 failure due to a timeout; and the mug exhibited the highest failure rate with 3 failures (2 drops and 1 timeout), resulting in a total of 6 failures. These findings indicate that the shape and characteristics of the object can significantly influence grasping performance, with irregularly shaped items like the mug posing greater challenges for the system.

6.4 Summary

In this chapter, we introduced a novel framework that combines Actor-Critic reinforcement learning with DNLS optimization module for dynamic object grasping in human-robot shared workspaces. In our approach, the RL component is tasked with generating target poses and assigning appropriate cost function weights, while the DNLS module refines these target poses to ensure that the resulting motion adheres to the

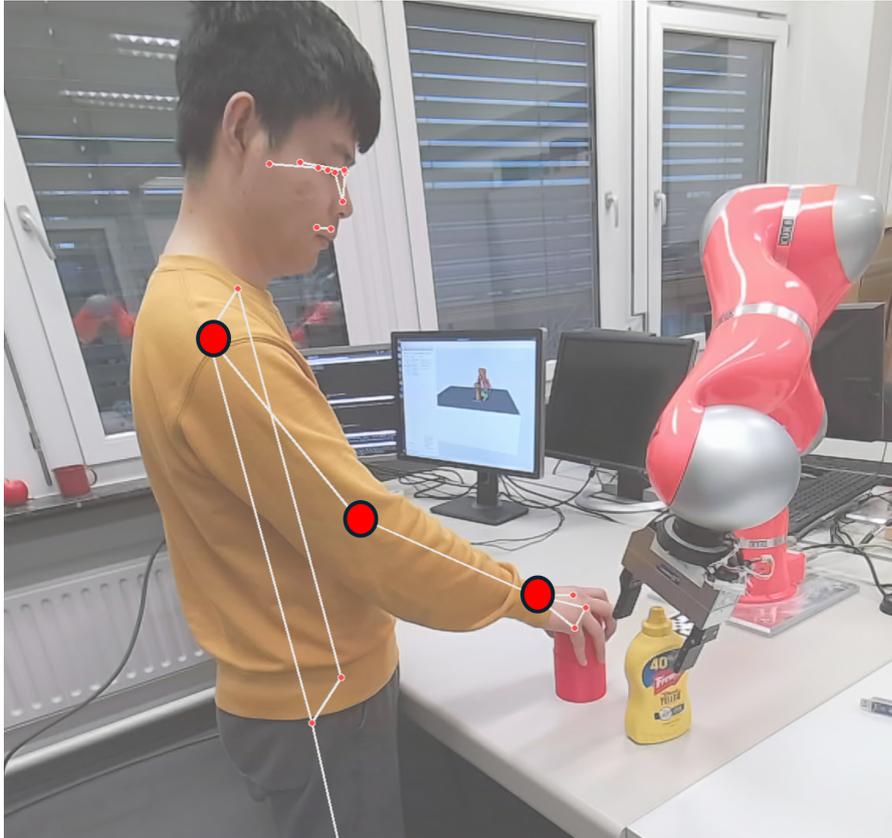


Figure 6.9: Human body tracking using MediaPipe Pose. The three red sphere markers represent the positions of the human shoulder, elbow, and wrist, respectively.

	suc. (%)	fail. (%)				
		contact w/ human	contact w/ ground	drop	timeout	total
banana	80	0	0	1	1	2
mustard	90	0	0	0	1	1
mug	70	0	0	2	1	3
overall	80	0	0	3	3	6

Table 6.5: Experiment Results for Grasping Different Objects. The table presents the success rates (succ.%) and failure rates (fail.%) for grasping three objects: a banana, a mustard bottle, and a mug. Failure modes are categorized into contact with the human, contact with the ground, object drops, and task timeouts. All results are based on 10 trials for each object.

robot’s kinodynamic and safety constraints. Notably, reformulating the problem using DNLS considerably accelerates the training process compared to previous methods, offering a more computationally efficient solution.

Our integrated framework has demonstrated significant advantages over baseline methods by achieving higher grasp success rates and smoother trajectory profiles in the challenging joint planning of motion and grasping. Both simulation and real-world

experiments confirm the robustness and effectiveness of our approach. However, it is important to note that the current system relies on having pre-estimated object poses, highlighting an area for future improvement to handle unknown objects more effectively.

Chapter 7

Conclusions and Future Work

This thesis has presented a series of integrated methods that collectively advance HRC in dynamic environments by unifying predictive planning, adaptive dexterous grasping, and joint motion and grasp planning.

Building on our work in predictive planning, we developed two complementary approaches—**Pred-HRC-Body** and **Pred-HRC-EEG**—that leverage both human motion and intention prediction. **Pred-HRC-Body** employs a Seq2Seq neural network combined with GMMs to forecast arm–hand trajectories and accurately estimate target positions, even during the early phases of motion when trajectories are highly similar. **Pred-HRC-EEG** further enhances this capability by incorporating EEG-based BCIs to predict human intentions and monitor operator vigilance, allowing the robot to proactively adjust its behavior to ensure both safety and efficiency.

Complementing these predictive strategies, we introduced **ADG-Net**, a novel framework for adaptive dexterous grasping. **ADG-Net** integrates multimodal data—including RGB-D images, tactile feedback, and joint angle information—using attention mechanisms and graph convolutional networks to learn robust grasping principles. To bridge the simulation-to-real gap, we generated an extensive synthetic dataset of over 500,000 grasp scenarios, which enabled **ADG-Net** to generalize effectively to real-world tasks, achieving state-of-the-art performance in grasp parameter prediction.

Finally, we presented **RL-DNLS**, a joint motion and grasp planning approach that merges an Actor-Critic reinforcement learning framework with DNLS optimization module. In this integrated system, the RL component generates target poses and dynamically tunes cost function weights, while the DNLS module refines these targets to satisfy kinodynamic and safety constraints. This combination not only produces smoother, collision-free trajectories but also accelerates training via GPU-accelerated computation, leading to higher grasp success rates compared to baseline methods.

Together, these contributions enhance the performance and safety of HRC by enabling predictive planning, adaptive grasping, and coordinated motion and grasp planning. Nevertheless, limitations remain, such as dependency on accurate object pose estimation and the computational overhead of the joint planning framework. Future work should focus on developing more robust real-time object detection, refining learning models to better account for human variability, and exploring more efficient optimization algorithms to further reduce computational costs. These efforts will pave the way

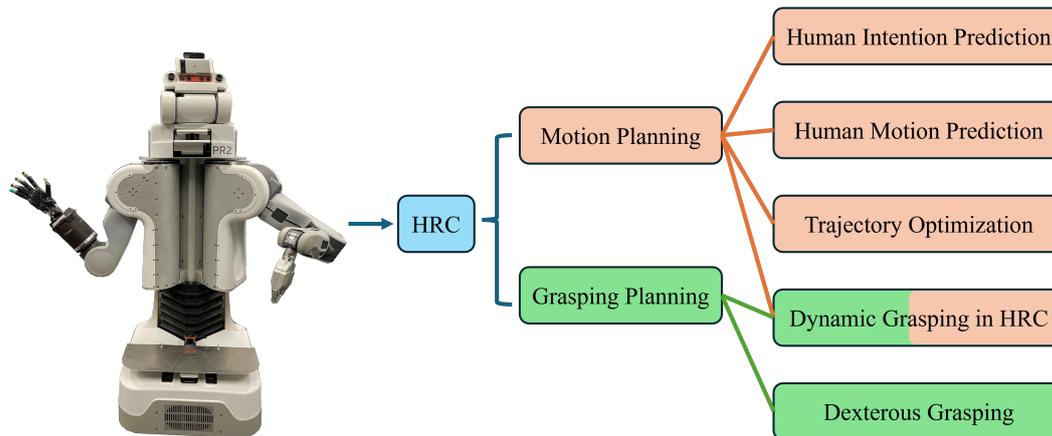


Figure 7.1: Overview of the research in this thesis. The yellow blocks are the related to motion planning modules, and the green blocks are related to the grasping planning modules. These modules represents the main research fields in this thesis.

for more autonomous and human-friendly robotic systems capable of operating in complex, dynamic environments.

7.1 Limitations

Although the approaches presented in this thesis have yielded promising results, several limitations remain that restrict their broader applicability. First, the methods developed for human motion and intention prediction are tailored to the specific HRC scenarios investigated in this work and may not directly transfer to other applications. For example, the EEG-based human intention prediction requires a brief individual calibration phase (approximately 10 trials) for each new participant. Moreover, the strategy used to integrate vigilance into the trajectory optimizer is relatively simplistic and provides only modest benefits in overall HRC performance. Additionally, fine-tuning the cost function weights for the trajectory optimizer to achieve an effective balance between task execution and collision avoidance remains a time-consuming and non-trivial process.

Second, the dynamic grasping framework based on RL and DNLS suffers from significant computational overhead due to the need to solve large-batch optimization problems during training. Although the plane-based collision constraint is efficient, it is somewhat conservative, which may lead to suboptimal policies. Furthermore, the multi-modal dexterous grasping system does not incorporate the material properties or deformation characteristics of objects, which can result in grasp failures. Grasping small or thin objects on the table continues to be particularly challenging. In addition, the current implementation does not perform object-level grasping, as segmentation information is not encoded in ADG-Net, and the persistent Sim2Real gap remains an unresolved issue.

These limitations highlight areas for future research, including the development of more generalized intention prediction methods, more sophisticated integration of vigilance signals, more efficient optimization techniques for large-batch training, and im-

provements in object modeling and segmentation to further enhance grasping performance in real-world scenarios.

7.2 Future Research

This work opens several promising avenues for future research. Building on the advances presented in this thesis, the following directions are particularly worth exploring:

Human Motion and Intention Prediction Currently, our prediction modules rely primarily on single modalities, such as human body motion or EEG signals. An exciting future direction is to fuse additional data sources—such as eye gaze and semantic information extracted from visual scenes—to further improve target estimation accuracy and robustness. Enhancing multimodal integration could broaden the applicability of our methods across more diverse and complex scenarios. In particular, reducing the data collection window for EEG-based intention prediction (which in our current setup requires a 2-second focus period) to only a few hundred milliseconds would markedly accelerate response times and enhance overall HRC performance. Moreover, exploring alternative strategies for adjusting robot velocity based on operator alertness—such as using neural networks to learn more complex relationships from EEG or other physiological signals—could further refine our approach. Finally, integrating additional sensing modalities, such as EMG alongside motion tracking and EEG, holds the promise of delivering faster, more precise intention and movement predictions.

Multimodal Dexterous Grasping The ADG-Net architecture shows strong promise for dexterous grasping; however, exploring novel neural network architectures, such as Transformers, may help overcome the limitations inherent in graph convolutional networks. These new architectures could enable a more seamless integration of multimodal sensory inputs—including RGB-D images and tactile data—and further enhance grasp prediction accuracy. Additionally, combining ADG-Net with vision-language models offers a compelling direction for enabling affordance-based and task-oriented manipulation in industrial settings, potentially allowing the system to understand and execute more complex, context-aware tasks.

Dynamic Grasping in HRC Our current dynamic grasping framework assumes that object poses are known a priori, which limits its utility in unstructured environments. Future work should focus on enabling the system to operate on previously unseen objects by directly incorporating object point clouds into the decision process. In addition, while our RL-DNLS approach has demonstrated improved performance, the training speed remains a bottleneck. Leveraging more advanced GPU-accelerated solvers and optimization techniques could further reduce computation times. Refining collision avoidance methods to be less conservative would also help in optimizing the balance between safety and performance, particularly in dynamic and cluttered workspaces.

Together, these future research directions aim to build on our current contributions by addressing the remaining challenges and expanding the scope and robustness of human–robot collaboration in dynamic, real-world environments.

Appendix A

List of Abbreviations

RNN Recurrent Neural Network

GCN Graph Convolutional Neural Network

MPC Model Predictive Control

DMP Dynamical Movement Primitives

IOC Inverse Optimal Control

AH Arm and Hand

DNLS Differentiable Nonlinear Least-Squares

HRI Human Robot Interaction

HRC Human Robot Collaboration

EEG Electroencephalography

RL Reinforcement Learning

BCI Brain-Computer Interaction

SSVEP Steady-State Visual Evoked Potential

SNR Signal-to-Noise

GMMs Gaussian Mixture Models

LSTM Long Short-Term Memory

MRCPs Movement-related Cortical Potentials

QP Quadratic Programming

FEMs Finite Element Methods

DL Deep Learning

SOTA State-of-the-Art

Sim2Real Sim-to-Real

SQP Sequential Quadratic Programming

DoF Degree of Freedom

AT Arm Tracking

VCV Vigilance-controlled Velocity

CCA Canonical Correlation Analysis

LDA Linear Discriminant Analysis

Appendix B

Publications

The following is the list of publications included in this thesis. The list is ordered by date.

- **Jianzhi Lyu**, Philipp Ruppel, Norman Hendrich, Shuang Li, Michael Görner, and Jianwei Zhang. Efficient and collision-free human-robot collaboration based on intention and trajectory prediction. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1, 2022.
- **Jianzhi Lyu**, Alexander Maÿe, Michael Görner, Philipp Ruppel, Andreas K Engel, and Jianwei Zhang. Coordinating human-robot collaboration by EEG-based human intention prediction and vigilance control. *Frontiers in Neurorobotics*, 16:1068274, 2022.
- **Jianzhi Lyu**, and Jianwei Zhang. Efficient and Safe Human-Robot Collaboration for Assistive Assembly, Workshop on Emerging paradigms for assistive robotic manipulation: from research labs to the real world, In *2023 IEEE International Conference on Robotics and Automation (ICRA)*.
- Hui Zhang, **Jianzhi Lyu***, Chuangchuang Zhou, Hongzhuo Liang, Yuyang Tu, Fuchun Sun, and Jianwei Zhang. Adg-net: A sim2Real multimodal learning framework for adaptive dexterous grasping. *IEEE Transactions on Cybernetics*, pages 1–14, 2025.
- **Jianzhi Lyu**, Hui Zhang, Ge Gao, Youshuang Ding, Fuchun Sun, and Jianwei Zhang. Dynamic Grasping based on Reinforcement Learning and Differentiable MPC. *IEEE Transactions on Automation Science and Engineering*, under review.

The following is a list of publications generated during the PhD period but not included in this thesis.

- Kecheng Shi, Rui Huang, Zhe Li, **Jianzhi Lyu**, Hong Cheng, and Jianwei Zhang. Plug-and-Play Multi-Domain Fusion Adaptation for Cross-subject EEG-based Motor Imagery Classification. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, accepted, 2024.

- Kecheng Shi, Fengjun Mu, Zhe Li, **Jianzhi Lyu**, Rui Huang, Chaobin Zou, Hong Cheng, and Jianwei Zhang. Corrupted eeg self-robust motion intention recognition method with multi-graph fusion enhanced convolutional network. In *2024 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 152–158, 2024.
- Hao Zhang, Hongzhuo Liang, Lin Cong, **Jianzhi Lyu**, Long Zeng, Pingfa Feng, and Jianwei Zhang. Reinforcement learning based pushing and grasping objects from ungraspable poses. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3860–3866, 2023.
- Song Tang, Yan Zou, Zihao Song, **Jianzhi Lyu**, Lijuan Chen, Mao Ye, Shouming Zhong, and Jianwei Zhang. Semantic consistency learning on manifold for source data-free unsupervised domain adaptation. *Neural Networks*, 152:467–478, 2022.
- Song Tang, Yuji Shi, Zhiyuan Ma, Jian Li, **Jianzhi Lyu**, Qingdu Li, and Jianwei Zhang. Model adaptation through hypothesis transfer with gradual knowledge distillation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5679–5685, 2021.
- Jinpeng Mi, **Jianzhi Lyu**, Song Tang, Qingdu Li, and Jianwei Zhang. Interactive natural language grounding via referring expression comprehension and scene graph parsing. *Frontiers in Neurobotics*, 14, 2020.
- Song Tang, Yunfeng Ji, **Jianzhi Lyu**, Jinpeng Mi, Qingdu Li, and Jianwei Zhang. Visual domain adaptation exploiting confidence-samples. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1173–1179, 2019.

Appendix C

Acknowledgements

Time flies, and my seven years of doctoral study with the TAMS research group are now drawing to a close. Reflecting on the journey filled with both challenges and triumphs, I am deeply grateful for the support and guidance I have received along the way.

First and foremost, I would like to express my sincere gratitude to my doctoral advisor, Professor Jianwei Zhang, for giving me the opportunity to pursue my studies abroad. When I received his invitation, I was filled with excitement, as it opened a new chapter in my academic journey. Throughout my doctoral studies, I encountered moments of uncertainty and formidable challenges. Professor Zhang was always by my side, providing unwavering support and guidance—ensuring that I stayed on track, even in unfamiliar research areas. He connected me with experts when needed. Without his steadfast encouragement, I would not have been able to complete my doctoral research.

I am also profoundly thankful to our group secretary, Lu. Whether handling reimbursements, organizing group activities, or assisting with everyday matters such as residence permits, and housing, Lu has always provided patient and thoughtful help. Over the years, Lu has cared for each cohort of students with warmth and consistency, giving us a true sense of home here in Hamburg and within the TAMS group. Even after I leave, this place will always hold a special place in my heart. Thank you, Lu.

I would like to extend my gratitude to my close collaborators: Norman, Alex, Philipp, Michael, Hui, Shuang, and Hongzhuo. Norman and Alex's passion for research and high standards have been a constant source of inspiration. Michael and Philipp's exceptional engineering skills and collaborative spirit were invaluable, especially during the early stages of my research career, making the journey less solitary. The clear planning and determined efforts for research of Hui, Shuang, and Hongzhuo inspired me a lot. I am also grateful to Ge, Chao, Shiqun, Kecheng, Song, Lin, Di, Wenkai, and Shang-Ching for the good times we spent together.

Finally, I would like to thank my family and my girlfriend. The selfless support of my older siblings (Jianshe Lyu, Yingying Lyu) and the unconditional love from my parents (Chuanbin Lyu, Zihua Ren) have allowed me to devote myself fully to my research overseas. One of the most precious personal outcomes of my time abroad has been meeting my girlfriend (Jean). The past seven years together have only deepened our bond, and we look forward to growing and embracing the future side by side.

Thank you all for being part of this journey.

Bibliography

- [1] Google AI. Mediapipe: A framework for building cross-platform ml solutions. <https://github.com/google/mediapipe>, 2023. Accessed: 2024-12-01.
- [2] Brendan Allison, Thorsten Luth, Diana Valbuena, Amir Teymourian, Ivan Volosyak, and Axel Graser. Bci demographics: How many (and what kinds of) people can use an ssvep bci? *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(2):107–116, 2010.
- [3] Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J Zico Kolter. Differentiable mpc for end-to-end planning and control. *Advances in neural information processing systems*, 31, 2018.
- [4] Thomas Timm Andersen. Optimizing the universal robots ros driver. Technical report, Technical University of Denmark, Department of Electrical Engineering, 2015.
- [5] Federico Bartoli, Giuseppe Lisanti, Lamberto Ballan, and Alberto Del Bimbo. Context-aware trajectory prediction. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1941–1946. IEEE, 2018.
- [6] Dhruv Batra, Angel X Chang, Sonia Chernova, Andrew J Davison, Jia Deng, Vladlen Koltun, Sergey Levine, Jitendra Malik, Igor Mordatch, Roozbeh Mottaghi, et al. Rearrangement: A challenge for embodied ai. 2020. *arXiv:2011.01975*. [Online]. Available: <https://arxiv.org/abs/2011.01975>.
- [7] David Blanco-Mulero, Oriol Barbany, Gokhan Alcan, Adrià Colomé, Carme Torras, and Ville Kyrki. Benchmarking the sim-to-real gap in cloth manipulation. 9(3):2981–2988, Mar. 2024.
- [8] BMW. Human-robot collaboration in the axle drive assembly, bmw group plant dingolfing, 2017.
- [9] Oliver Brock and Oussama Khatib. Elastic strips: A framework for motion generation in human environments. *The International Journal of Robotics Research*, 21(12):1031–1052, 2002.

- [10] Stanislas Brossette and Pierre-Brice Wieber. Collision avoidance based on separating planes for feet trajectory generation. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 509–514. IEEE, 2017.
- [11] Achim Buerkle, William Eaton, Niels Lohse, Thomas Bamber, and Pedro Ferreira. EEG based arm movement intention recognition towards enhanced safety in symbiotic human-robot collaboration. *Robotics and Computer-Integrated Manufacturing*, 70:102137, 2021.
- [12] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: using the Yale-CMU-Berkeley object and model set. 22(3):36–52, Sep. 2015.
- [13] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics and Automation Magazine*, 22(3):36–52, 2015.
- [14] Teng Cao, Feng Wan, Chi Man Wong, Janir Nuno da Cruz, and Yong Hu. Objective evaluation of fatigue by EEG spectral analysis in steady-state visual evoked potential-based brain-computer interfaces. *Biomedical Engineering Online*, 13(1):1–13, 2014.
- [15] Afonso Castro, Filipe Silva, and Vitor Santos. Trends of Human-Robot Collaboration in Industry Contexts: Handover, Learning, and Metrics. *Sensors*, 21(12):4113, 2021.
- [16] Jiayi Chen, Yuxing Chen, Jialiang Zhang, and He Wang. Task-oriented dexterous grasp synthesis via differentiable grasp wrench boundary estimator. 2023. *arXiv:2309.13586*. [Online]. Available: <https://arxiv.org/abs/2309.13586>.
- [17] Xiaogang Chen, Bing Zhao, Yijun Wang, and Xiaorong Gao. Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm. *Journal of Neural Engineering*, 16(2):026012, 2019.
- [18] Xiaogang Chen, Bing Zhao, Yijun Wang, Shengpu Xu, and Xiaorong Gao. Control of a 7-DOF robotic arm system with an SSVEP-based BCI. *International Journal of Neural Systems*, 28(08):1850018, 2018.
- [19] Zixi Chen, Shixin Zhang, Shan Luo, Fuchun Sun, and Bin Fang. Tacchi: A plug-gable and low computational cost elastomer deformation simulator for optical tactile sensors. 8(3):1239–1246, Jan. 2023.
- [20] Yujiao Cheng, Liting Sun, Changliu Liu, and Masayoshi Tomizuka. Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction. *IEEE Robotics and Automation Letters*, 5(2):2602–2609, 2020.

-
- [21] Yujiao Cheng, Weiye Zhao, Changliu Liu, and Masayoshi Tomizuka. Human motion prediction using semi-adaptable neural networks. In *2019 American Control Conference (ACC)*, pages 4884–4890. IEEE, 2019.
- [22] Sachin Chitta, Eitan Marder-Eppstein, Wim Meeussen, Vijay Pradeep, Adolfo Rodríguez Tsouroukdissian, Jonathan Bohren, David Coleman, Bence Magyar, Gennaro Raiola, Mathias Lüdtkke, et al. ros_control: A generic and simple control framework for ROS. *The Journal of Open Source Software*, 2(20):456–456, 2017.
- [23] Sammy Christen, Wei Yang, Claudia Pérez-D’Arpino, Otmar Hilliges, Dieter Fox, and Yu-Wei Chao. Learning human-to-robot handovers from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9654–9664, 2023.
- [24] Matei T Ciocarlie and Peter K Allen. Hand posture subspaces for dexterous robotic grasping. *Int. J. Robot. Res.*, 28(7):851–867, Jun. 2009.
- [25] David Coleman, Ioan Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a Moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.
- [26] Ian Coleman, Ioan A Sucan, Sachin Chitta, and Nikolaus Correll. Moveit!: An intelligent motion planning framework for robotic manipulation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pages 1748–1754. IEEE, 2014.
- [27] William Jay Conover. *Practical nonparametric statistics*, volume 350. John Wiley & Sons, 1999.
- [28] Sara Cooper, Samuel FP Fensome, Dimitrios Kourtis, Stuart Gow, and Mauro Dragone. An EEG investigation on planning human-robot handover tasks. In *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, pages 1–6. IEEE, 2020.
- [29] Matt Corsaro, Stefanie Tellex, and George Konidaris. Learning to detect multi-modal grasps for dexterous grasping in dense clutter. In *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 4647–4653, Sep. 2021.
- [30] Bolun Dai, Rooholla Khorrambakht, Prashanth Krishnamurthy, Vinícius Gonçalves, Anthony Tzes, and Farshad Khorrami. Safe navigation and obstacle avoidance using differentiable optimization based control barrier functions. *IEEE Robotics and Automation Letters*, 8(9):5376–5383, 2023.
- [31] DLR. Planning and execution of daily cleaning tasks with the humanoid service robot rollin’ justin, 2017.

- [32] Jing Dong, Mustafa Mukadam, Frank Dellaert, and Byron Boots. Motion Planning as Probabilistic Inference using Gaussian Processes and Factor Graphs. In *Robotics: Science and Systems*, volume 12, page 4, 2016.
- [33] Le Fu and Jie Zhao. Maxwell-Model-Based Compliance Control for Human–Robot Friendly Interaction. *IEEE Transactions on Cognitive and Developmental Systems*, 13(1):118–131, 2020.
- [34] Letian Fu, Huang Huang, Lars Berscheid, Hui Li, Ken Goldberg, and Sachin Chitta. Safe self-supervised learning in real of visuo-tactile feedback policies for industrial insertion. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 10380–10386, May 2023.
- [35] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 3749–3761, Jun. 2022.
- [36] Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José MF Moura. Few-shot human motion prediction via meta-learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 432–450, 2018.
- [37] Di Guo, Fuchun Sun, Bin Fang, Chao Yang, and Ning Xi. Robotic grasping using visual and tactile sensing. *Inf. Sci.*, 417:274–286, Nov. 2017.
- [38] Kris Hauser. On responsiveness, safety, and completeness in real-time motion planning. *Autonomous Robots*, 32(1):35–48, 2012.
- [39] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [40] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [41] Chenxi Huang, Yutian Xiao, and Gaowei Xu. Predicting human intention-behavior through EEG signal analysis using Multi-Scale CNN. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2020.
- [42] Jeffrey Ichnowski, Michael Danielczuk, Jingyi Xu, Vishal Satish, and Ken Goldberg. Gomp: Grasp-optimized motion planning for bin picking. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5270–5277, 2020.
- [43] Roland S Johansson, Göran Westling, Anders Bäckström, and J Randall Flanagan. Eye–hand coordination in object manipulation. *Journal of Neuroscience*, 21(17):6917–6932, 2001.

- [44] Christian Juelg, Andreas Hermann, Arne Roennau, and Rüdiger Dillmann. Efficient, collaborative screw assembly in a shared workspace. In *Intelligent Autonomous Systems 15: Proceedings of the 15th International Conference IAS-15*, pages 837–848. Springer, 2019.
- [45] Tzyy-Ping Jung, Scott Makeig, Magnus Stensmo, and Terrence J Sejnowski. Estimating alertness from the EEG power spectrum. *IEEE transactions on biomedical engineering*, 44(1):60–69, 1997.
- [46] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on robot learning*, pages 651–673. PMLR, 2018.
- [47] Mitsuhiro Kamezaki, Tomohiro Wada, and Shigeki Sugano. Dynamic collaborative workspace based on human interference estimation for safe and productive human-robot collaboration. *IEEE Robotics and Automation Letters*, 9(7):6568–6575, 2024.
- [48] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proc. Int. Conf. Learn. Representions*, May 2015.
- [49] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pages 4015–4026, Oct. 2023.
- [50] Dimche Kostadinov and Davide Scaramuzza. Online weight-adaptive nonlinear model predictive control. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1180–1185. IEEE, 2020.
- [51] Danica Kragic and Yulia Sandamirskaya. Effective and natural human-robot interaction requires multidisciplinary research. *Science Robotics*, 6(58):eabl7022, 2021.
- [52] Florian Kuhnt, Ralf Kohlhaas, Thomas Schamm, and J Marius Zöllner. Towards a unified traffic situation estimation model—street-dependent behaviour and motion models. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 1223–1229. IEEE, 2015.
- [53] Michael Land, Neil Mennie, and Jennifer Rusted. The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28(11):1311–1328, 1999.
- [54] Chiara Talignani Landi, Yujiao Cheng, Federica Ferraguti, Marcello Bonfè, Cristian Secchi, and Masayoshi Tomizuka. Prediction of human arm target for robot reaching movements. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5950–5957. IEEE, 2019.

- [55] Bo-Yeon Lee, Seonggi Kim, Sunjong Oh, Youngoh Lee, Jonghwa Park, Hyunhyub Ko, Ja Choon Koo, Youngdo Jung, and Hyuneui Lim. Human-inspired tactile perception system for real-time and multimodal detection of tactile stimuli. *Soft Robot.*, Dec. 2023.
- [56] Kang-Won Lee, Yuzhe Qin, Xiaolong Wang, and Soo-Chul Lim. Dex-Touch: Learning to seek and manipulate objects with tactile dexterity. 2024. *arXiv:2401.12496*. [Online]. Available: <https://arxiv.org/abs/2401.12496>.
- [57] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. RealDex: Towards human-like grasping for robotic dexterous hand. 2024. *arXiv:2402.13853*. [Online]. Available: <https://arxiv.org/abs/2402.13853>.
- [58] Shen Li and Julie A Shah. Safe and efficient high dimensional motion planning in space-time with time parameterized prediction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5012–5018. IEEE, 2019.
- [59] Shuang Li, Norman Hendrich, Hongzhuo Liang, Philipp Ruppel, Changshui Zhang, and Jianwei Zhang. A dexterous hand-arm teleoperation system based on hand pose estimation and active vision. *54(3):1417–1428*, Mar. 2024.
- [60] Hongzhuo Liang, Lin Cong, Norman Hendrich, Shuang Li, Fuchun Sun, and Jianwei Zhang. Multifingered grasping based on multimodal reinforcement learning. *7(2):1174–1181*, Apr. 2022.
- [61] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Gornier, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. PointNetGPD: detecting grasp configurations from point sets. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 3629–3635, May 2019.
- [62] Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. PointNetGPD: Detecting grasp configurations from point sets. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [63] Min Liu, Zherong Pan, Kai Xu, Kanishka Ganguly, and Dinesh Manocha. Deep differentiable grasp planner for high-dof grippers. In *Robot.: Scien. Sys*, Jul. 2020.
- [64] Xibai Lou, Yang Yang, and Changhyun Choi. Collision-aware target-driven object grasping in constrained environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6364–6370, 2021.
- [65] Jiaxin Lu, Hao Kang, Haoxiang Li, Bo Liu, Yiding Yang, Qixing Huang, and Gang Hua. UGG: Unified generative grasping. 2023. *arXiv:2311.16917*. [Online]. Available: <https://arxiv.org/abs/2311.16917>.

- [66] Ren C Luo and Licong Mai. Human intention inference and on-line human hand motion prediction for human-robot collaboration. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5958–5964. IEEE, 2019.
- [67] Ruikun Luo, Rafi Hayne, and Dmitry Berenson. Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces. *Autonomous Robots*, 42(3):631–648, 2018.
- [68] Jianzhi Lyu, Alexander Maÿe, Michael Görner, Philipp Ruppel, Andreas K Engel, and Jianwei Zhang. Coordinating human-robot collaboration by EEG-based human intention prediction and vigilance control. *Frontiers in Neurorobotics*, 16:1068274, 2022.
- [69] Jianzhi Lyu, Philipp Ruppel, Norman Hendrich, Shuang Li, Michael Görner, and Jianwei Zhang. Efficient and collision-free human–robot collaboration based on intention and trajectory prediction. *IEEE Transactions on Cognitive and Developmental Systems*, 15(4):1853–1863, 2023.
- [70] Miles Macklin, Matthias Müller, Nuttapong Chentanez, and Tae-Yong Kim. Unified particle physics for real-time applications. *ACM Trans. Graph.*, 33(4), jul 2014.
- [71] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Robot.: Scien. Sys.*, Jul. 2017.
- [72] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Sci. Robot.*, 4(26), Jan. 2019.
- [73] Jim Mainprice, Rafi Hayne, and Dmitry Berenson. Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces. *IEEE Transactions on Robotics*, 32(4):897–908, 2016.
- [74] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU based physics simulation for robot learning. In *Proc. Conf. NeurIPS*, Oct. 2021.
- [75] Monica Malvezzi, Guido Gioioso, Gionata Salvietti, Domenico Prattichizzo, and Antonio Bicchi. SynGrasp: a MATLAB toolbox for grasp analysis of human and robotic hands. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 1088–1093, May 2013.
- [76] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 6169–6176, May 2021.

- [77] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 734–743. PMLR, 2018.
- [78] Alexander Maye, Dan Zhang, and Andreas K Engel. Utilizing retinotopic mapping for a multi-target SSVEP BCI with a single flicker frequency. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(7):1026–1036, 2017.
- [79] Microsoft. Azure kinect dk. <https://learn.microsoft.com/en-us/azure/kinect-dk/>. Accessed: 2024-12-05.
- [80] Andrew T. Miller and Peter K. Allen. GraspIt! 11(4):110–122, Dec. 2004.
- [81] A.T. Miller and P.K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.
- [82] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023.
- [83] Andreas Møgelmoose, Mohan M Trivedi, and Thomas B Moeslund. Trajectory analysis and prediction for improved pedestrian safety: Integrated framework and evaluations. In *2015 IEEE intelligent vehicles symposium (IV)*, pages 330–335. IEEE, 2015.
- [84] Manfred Morari, Carlos E Garcia, and David M Prett. Model predictive control: theory and practice. *IFAC Proceedings Volumes*, 21(4):1–12, 1988.
- [85] Douglas Morrison, Peter Corke, and Jürgen Leitner. Learning robust, real-time, reactive robotic grasping. *Int. J. Robot. Res.*, 39(2-3):183–201, Jun. 2020.
- [86] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [87] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6232–6238, 2020.
- [88] Yashraj Narang, Balakumar Sundaralingam, Miles Macklin, Arsalan Mousavian, and Dieter Fox. Sim-to-real for robotic tactile sensing via physics-based simulation and learned latent projections. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 6444–6451, May 2021.

- [89] Rhys Newbury, Morris Gu, Lachlan Chumbley, Arsalan Mousavian, Clemens Eppner, Jürgen Leitner, Jeannette Bohg, Antonio Morales, Tamim Asfour, Danica Kragic, Dieter Fox, and Akansel Cosgun. Deep learning approaches to grasp synthesis: A review. *39(5):3994–4015*, Oct. 2023.
- [90] NVIDIA Corporation. Nvidia physx sdk, 2024. Accessed: December 28, 2024.
- [91] OpenAI. Creating video from text: Sora is an ai model that can create realistic and imaginative scenes from text instructions. *Openai.com*. <https://openai.com/sora> [Accessed: Mar. 07, 2024].
- [92] Jae Sung Park, Chonhyon Park, and Dinesh Manocha. I-planner: Intention-aware motion planning using learning-based human motion prediction. *The International Journal of Robotics Research*, 38(1):23–39, 2019.
- [93] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.
- [94] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037. Curran Associates, Inc., 2019.
- [95] Claudia Pérez-D’Arpino and Julie A Shah. Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 6175–6182. IEEE, 2015.
- [96] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, et al. Theseus: A library for differentiable nonlinear optimization. *Advances in Neural Information Processing Systems*, 35:3801–3818, 2022.
- [97] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jul. 2017.
- [98] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017.
- [99] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

- [100] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dex-point: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. *Conference on Robot Learning (CoRL)*, 2022.
- [101] Yuzhe Qin, Binghao Huang, Zhao-Heng Yin, Hao Su, and Xiaolong Wang. Dex-Point: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation. In *Proc. Conf. Rob. Learn.*, pages 594–605, Dec. 2023.
- [102] Shiyin Qiu, Wei Guo, Darwin Caldwell, and Fei Chen. Exoskeleton online learning and estimation of human walking intention based on dynamical movement primitives. *IEEE Transactions on Cognitive and Developmental Systems*, 13(1):67–79, 2020.
- [103] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.
- [104] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.
- [105] Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6284–6291, 2018.
- [106] Nathan Ratliff, Matt Zucker, J. Andrew Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pages 489–494, 2009.
- [107] Harish Chaandar Ravichandar, Avnish Kumar, and Ashwin Dani. Gaze and motion information fusion for human intention inference. *International Journal of Intelligent Robotics and Applications*, 2(2):136–148, 2018.
- [108] Shadow Robot. Shadow dexterous hand technical specification. *Shadowrobot.com*. <https://www.shadowrobot.com/dexterous-hand-series> [Accessed: Mar. 07, 2024].
- [109] Angel Romero, Shreedhar Govil, Gonca Yilmaz, Yunlong Song, and Davide Scaramuzza. Weighted maximum likelihood for controller tuning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1334–1341. IEEE, 2023.
- [110] Angel Romero, Yunlong Song, and Davide Scaramuzza. Actor-critic model predictive control. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14777–14784. IEEE, 2024.

- [111] Wilson Ruotolo, Dane Brouwer, and Mark R. Cutkosky. From grasping to manipulation with gecko-inspired adhesives on a multifinger gripper. *Sci. Robot.*, 6(61):eabi9773, Dec. 2021.
- [112] Philipp Ruppel and Jianwei Zhang. Learning Object Manipulation with Dexterous Hand-Arm Systems from Human Demonstration. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5417–5424. IEEE, 2020.
- [113] Philipp Ruppel and Jianwei Zhang. Learning object manipulation with dexterous hand-arm systems from human demonstration. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5417–5424. IEEE, 2020.
- [114] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied ai research. In *Proc. Int. Conf. Comput. Vis.*, pages 9338–9346, Oct. 2019.
- [115] Narcís Sayols, Alessio Sozzi, Nicola Piccinelli, Albert Hernansanz, Alicia Casals, Marcello Bonfè, and Riccardo Muradore. Global/local motion planning based on dynamic trajectory reconfiguration and dynamical systems for autonomous surgical robots. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8483–8489. IEEE, 2020.
- [116] SM Shafiul Hasan, Masudur R Siddiquee, Roozbeh Atri, Rodrigo Ramon, J Sebastian Marquez, and Ou Bai. Prediction of gait intention from pre-movement eeg signals: a feasibility study. *Journal of NeuroEngineering and Rehabilitation*, 17(1):1–16, 2020.
- [117] Weiwei Shang, Fangjing Song, Zengzhi Zhao, Hongbo Gao, Shuang Cong, and Zhijun Li. Deep learning method for grasping novel objects using dexterous hands. *52(5):2750–2762*, May 2022.
- [118] Richard B Silberstein, Mark A Schier, Andrew Pipingas, Joseph Ciorciari, Stephen R Wood, and David G Simpson. Steady-state visually evoked potential topography associated with a visual vigilance task. *Brain Topography*, 3(2):337–347, 1990.
- [119] Yonghao Song, Siqi Cai, Lie Yang, Guofeng Li, Weifeng Wu, and Longhan Xie. A practical EEG-Based Human-Machine Interface to Online Control an Upper-Limb Assist Robot. *Frontiers in Neurorobotics*, 14:32, 2020.
- [120] Yunlong Song and Davide Scaramuzza. Learning high-level policies for model predictive control. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7629–7636. IEEE, 2020.

- [121] Hang Su, Yingbai Hu, Hamid Reza Karimi, Alois Knoll, Giancarlo Ferrigno, and Elena De Momi. Improved recurrent neural network-based manipulator control with remote center of motion constraints: Experimental results. *Neural Networks*, 131:291–299, 2020.
- [122] Daniel Sunday. Practical geometry algorithms: With c++ code. *KDP Print US*, 2021.
- [123] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2 edition, 2018.
- [124] Andreas Ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.
- [125] Tesla. Tesla optimus gen2, 2024.
- [126] Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 5026–5033, Oct 2012.
- [127] Susanne Trick, Dorothea Koert, Jan Peters, and Constantin A Rothkopf. Multi-modal uncertainty reduction for intention recognition in human-robot interaction. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7009–7016. IEEE, 2019.
- [128] Vaibhav V Unhelkar, Shen Li, and Julie A Shah. Decision-making for bidirectional communication in sequential human-robot collaborative tasks. In *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 329–341. IEEE, 2020.
- [129] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023.
- [130] Guido Van Rossum, Fred L Drake, et al. *Python reference manual*. iUniverse Indiana, 2000.
- [131] Kim P Wabersich, Lukas Hewing, Andrea Carron, and Melanie N Zeilinger. Probabilistic model predictive safety certification for learning-based control. *IEEE Transactions on Automatic Control*, 67(1):176–188, 2021.
- [132] Hongsong Wang, Jian Dong, Bin Cheng, and Jiashi Feng. PVRED: A Position-Velocity Recurrent Encoder-Decoder for Human Motion Prediction. *IEEE Transactions on Image Processing*, 30:6096–6106, 2021.

- [133] Liangliang Wang, Qiang Li, James Lam, Zheng Wang, and Zhengyou Zhang. Intent inference in shared-control teleoperation system in consideration of user behavior. *Complex & Intelligent Systems*, pages 1–11, 2021.
- [134] Lirui Wang, Yu Xiang, Wei Yang, Arsalan Mousavian, and Dieter Fox. Goal-auxiliary actor-critic for 6d robotic grasping with point clouds. In *Conference on Robot Learning*, pages 70–80. PMLR, 2022.
- [135] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. DexGraspNet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 11359–11366, May 2023.
- [136] Yiwei Wang, Yixuan Sheng, Ji Wang, and Wenlong Zhang. Optimal collision-free robot trajectory generation based on time series prediction of human motion. *IEEE Robotics and Automation Letters*, 3(1):226–233, 2017.
- [137] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17868–17879, June 2024.
- [138] Thomas Weng, David Held, Franziska Meier, and Mustafa Mukadam. Neural grasp distance fields for robot manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1814–1821. IEEE, 2023.
- [139] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. In *Proc. Eur. Conf. Comput. Vis.*, pages 3–19, Sep. 2018.
- [140] Pengwen Xiong, Xiaobao Tong, Aiguo Song, and Peter X. Liu. Robotic multifinger grasping state recognition based on adaptive multikernel dictionary learning. 71:1–14, May 2022.
- [141] Baoguo Xu, Dalin Zhang, Yong Wang, Leying Deng, Xin Wang, Changcheng Wu, and Aiguo Song. Decoding different reach-and-grasp movements using non-invasive electroencephalogram. *Frontiers in Neuroscience*, 15:1206, 2021.
- [142] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, Tengyu Liu, Li Yi, and He Wang. UniDexGrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pages 4737–4746, Jun. 2023.
- [143] Wei Yang, Chris Paxton, Arsalan Mousavian, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Reactive human-to-robot handovers of arbitrary objects. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3118–3124, 2021.

- [144] Wei Yang, Balakumar Sundaralingam, Chris Paxton, Iretoiyo Akinola, Yu-Wei Chao, Maya Cakmak, and Dieter Fox. Model predictive control for fluid human-to-robot handovers. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6956–6962, 2022.
- [145] Chao Zeng, Shuang Li, Zhaopeng Chen, Chenguang Yang, Fuchun Sun, and Jianwei Zhang. Multifingered robot hand compliant manipulation based on vision-based demonstration and adaptive force control. *34(9):5452–5463*, Sep. 2023.
- [146] Dalin Zhang, Lina Yao, Kaixuan Chen, Sen Wang, Xiaojun Chang, and Yunhao Liu. Making sense of spatio-temporal preserving representations for EEG-based human intention recognition. *IEEE Transactions on Cybernetics*, *50(7):3033–3044*, 2019.
- [147] Hui Zhang, Jianzhi Lyu, Chuangchuang Zhou, Hongzhuo Liang, Yuyang Tu, Fuchun Sun, and Jianwei Zhang. Adg-net: A sim2Real multimodal learning framework for adaptive dexterous grasping. *IEEE Transactions on Cybernetics*, pages 1–14, 2025.
- [148] Hui Zhang, Jef Peeters, Eric Demeester, and Karel Kellens. A CNN-based grasp planning method for random picking of unknown objects with a vacuum gripper. *J. Intell. Robot. Syst.*, *103(64):1–19*, Nov. 2021.
- [149] Hui Zhang, Jef Peeters, Eric Demeester, and Karel Kellens. Deep learning reactive robotic grasping with a versatile vacuum gripper. *39(2):1244–1259*, Apr. 2023.
- [150] Hui Zhang, Yanming Wu, Eric Demeester, and Karel Kellens. BIG-Net: Deep learning for grasping with a bio-inspired soft gripper. *8(2):584–591*, Feb. 2023.
- [151] Can Zhao, Jieji Ren, Hexi Yu, and Daolin Ma. In-situ mechanical calibration for vision-based tactile sensors. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 10387–10393, May 2023.
- [152] Weiye Zhao, Liting Sun, Changliu Liu, and Masayoshi Tomizuka. Experimental evaluation of human motion prediction toward safe and efficient human robot collaboration. In *2020 American Control Conference (ACC)*, pages 4349–4354. IEEE, 2020.
- [153] Zengzhi Zhao, Weiwei Shang, Haoyuan He, and Zhijun Li. Grasp prediction and evaluation of multi-fingered dexterous hands using deep learning. *Robot. Autom. Sys.*, *129:103550*, Jul. 2020.
- [154] Pu Zheng, Pierre-Brice Wieber, and Olivier Aycard. Online optimal motion generation with guaranteed safety in shared workspace. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9210–9215. IEEE, 2020.

Erklärung der Urheberschaft

Hiermit versichere ich an Eides statt, dass ich die vorliegende Dissertation im Studiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Hamburg, 20.02.2025

Ort, Datum



Unterschrift

Erklärung zur Veröffentlichung

Ich stimme der Einstellung der Dissertation in die Bibliothek des Fachbereichs Informatik zu.

Hamburg, 20.02.2025

Ort, Datum

A handwritten signature in blue ink that reads "Jianzhi Lyu". The signature is written in a cursive style.

Unterschrift

