UNIVERSITY OF CALIFORNIA

Los Angeles

Multimodal Communication for Embodied Human-Robot Interaction with Natural Gestures

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Qi Wu

2021

ABSTRACT OF THE DISSERTATION

Multimodal Communication for Embodied Human-Robot Interaction
with Natural Gestures

by

Qi Wu
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2021
Professor Mohammad Khalid Jawed, Co-Chair
Professor Jungseock Joo, Co-Chair

Communication takes place in various forms and is an essential part of human-human communication. Researchers have done a plethora of studies to understand it biologically and computationally. Likewise, it also plays a significant role in Human-Robot Interaction (HRI) in order to endow Artificial Intelligence (AI) systems with humanlike cognition and sociality. With the advancement of realistic simulators, multimodal HRI is also a hotspot with embodied agents in the simulation. Human users should be able to manipulate or collaborate with embodied agents in multimodal ways, including verbal and non-verbal methods. Up to now, most prior works with embodied AI have been focusing on addressing embodied agent tasks using verbal cues along with visual perceptions, *e.g.*, using human languages as natural language instructions to assist embodied visual navigation tasks. Nonetheless, non-verbal means of communication like gestures, which are rooted in human communication, are rarely examined in embodied agent tasks. In this dissertation, I contemplate on existing research topics in embodied AI and propose to tackle embodied visual navigation tasks with natural human gestures to fill the deficiency of non-verbal communicative interfaces in embodied HRI. It has the following contributions:

- To this end, I first develop a 3D photo-realistic simulation environment, Gesture-based THOR (GesTHOR). In this simulator, the human user can wear a Virtual Reality (VR) Head-Mounted Display (HMD) to be immersed as a humanoid agent in the same environment as the robot agent and communicate with the robot interactively using instructional gestures by sensory devices that can track body and hand motions. I provide data collection tools so that users can generate their own gesture data in our simulation environment.
- I created Gesture ObjectNav Dataset (GOND) and standardized benchmarks to evaluate how gestures contribute to the embodied object navigation task. This dataset contains natural gestures collected from human users and object navigation tasks defined in GesTHOR.
- To demonstrate the effectiveness of gestures for embodied navigation, I build an end-to-end Reinforcement Learning (RL) model that can integrate multimodal perceptions for the robot to learn the optimal navigation policies. Through cases studies with GOND in GesTHOR, I illustrate that the robot agent can perform the navigation task successfully and efficiently with gestures instead of natural languages. I also show that the navigation agent can learn the underlined semantics of unpredefined gestures, which is beneficial to its navigation.

By introducing GesTHOR and GOND as well as related experimental results, I aim to spur growing interest in embodied HRI with non-verbal communicative interfaces toward building cognitive AI systems.

The dissertation of Qi Wu is approved.

University of California, Los Angeles

2021

To my parents and Siting

TABLE OF CONTENTS

1 Introduction **1**

2 Background & Motivation **6**

2.1 Gestures in Multimodal Communication 6

2.2 Human-Robot Interaction with Gestures 9

2.3 Embodied AI Learning 14

2.3.1 Embodied AI Simulators 15

2.3.2 Embodied Visual Navigation 17

2.3.3 Reinforcement Learning for Embodied Learning 22

3 GesTHOR: Simulation Framework for Interactive Embodied Agent . . . **29**

3.1 Introduction 29

3.2 Simulation Environment 31

3.2.1 Environment Overview 31

3.2.2 Agent Embodiment 32

3.2.3 VR Interaction Interface 33

3.3 Gesture ObjectNav Dataset 39

3.3.1 Implementation Details 39

3.3.2 Dataset Statistics 41

4 Embodied Navigation with Natural Gestures **46**

4.1 Introduction 46

4.2 Problem Formulation 47

4.3 RL Learning Architecture . . . 48

4.3.1 Learning Objective . . . 48

4.3.2 PPO Implementation . . . 50

4.3.3 Learning Network . . . 51

4.4 Experiments . . . 53

4.4.1 Evaluation Criteria . . . 53

4.4.2 Navigation Performance . . . 56

4.4.3 Ablation Studies . . . 58

4.4.4 Qualitative Results . . . 60

5 Conclusions & Discussions . . . 62

5.1 Conclusions . . . 62

5.2 Limitations . . . 64

5.3 Future Directions . . . 66

References . . . 69

LIST OF FIGURES

1.1 An example illustrating the significance of gestures in communication. 4

2.1 Different types of gestures used in our work. 7

2.2 Some typical predefined symbolic gestures. 13

2.3 Different research tasks in embodied AI. 18

2.4 Schematic diagram of reinforcement learning. 22

3.1 Overview of GesTHOR. 30

3.2 The humanoid agent and robot agent. 32

3.3 Device arrangement for VR interaction in GesTHOR 35

3.4 Record human motions in a program. 37

3.5 Human motion data processing. 37

3.6 Replay a recorded program. 39

3.7 Examples of referencing and intervention gestures. 40

3.8 Distribution of scene types in the train fold. 41

3.9 Distribution of object types in the train fold. 43

3.10 Object diversity in our gesture dataset. 44

3.11 Number of instances with the target category. 44

3.12 Spatial relationship in the scene. 45

4.1 The RL learning model. 53

4.2 Qualitative results for referencing gestures 61

4.3 Qualitative results for intervention gestures 61

LIST OF TABLES

2.1 Summary of embodied AI simulators. 28

3.1 Metadata of an example program. 36

3.2 Tracked human body bones. 36

3.3 Object categories in GOND for different scene types. 43

4.1 Hyperparameters for training Proximal Policy Optimization (PPO) agent. . . . 50

4.2 Navigation performance for baseline and our methods. 56

4.3 Evaluation results for train/validation/test split with different scene types. . . . 56

4.4 Evaluation results for test scenes on four scene types with the different number of allowed stops 57

4.5 Navigation performance on multi-instance scenes. 58

4.6 Navigation performance in blind mode. 59

4.7 Ablation study on visual perceptions. 60

ACKNOWLEDGMENTS

I feel grateful to meet a collection of people in my Ph.D. study:

Dr. Jungseock Joo, for his guidance and continuous support during my research, for showing me a broad view in the field of human-robot interaction and machine learning, and especially for giving insights on how to be a researcher and how to plan my career path.

Dr. M. Khalid Jawed, for being my advisor after I changed my research directions, for providing me the opportunity to collaborate with Dr. Jungseock Joo, and for having a lot of great discussions on my research topics.

Dr. Chang-Jin Kim, for being my advisor when I first came to UCLA. Although I didn't continue research in his lab for some reason, it is great to have such a remarkable professor to provide me with valuable academic guidance and research experiences at the beginning of my Ph.D. study.

Dr. Eric Pei-Yu Chiou, Dr. Jeffrey D. Eldredge, and Dr. Ximin He, for serving as my initial committee members and giving me helpful instructions on my prospectus and initial research explorations.

Dr. Veronica Santos and Dr. Sriram Narasimhan, for being members of my reconstituted committee and helping me with my dissertation and final defense.

Dr. Yixin Zhu, for collaborating with me and for teaching me how to do scientific research, write scientific reports.

Shuaihang Pan, Zhenyu She, Jinzhao Hu, Ning Yu, Zhengxiang Yi, Yi Luo, and so many else, for being my best friends while I was at UCLA.

My parents, for backing me up while I was down and supporting me to their best, my girlfriend Siting Liu, for being my companion both physically and mentally.

Portions of this work were supported by NSF NRI #1925360.

VITA

2021	AI Resident, X development LLC
2019-2021	Graduate Research Assistant, UCLA Communication Department, Dr. Jungseock Joo
2019-2021	Graduate Teaching Assistant, UCLA Life Science Department
2018	M.S. in Mechanical Engineering, UCLA
2018	Ph.D. Candidate in Mechanical Engineering, UCLA
2016	B.E. in Mechanical Engineering and Automatics, Shanghai Jiao Tong University, China
2016	B.Sc. in Electromechanical Engineering, Katholieke Universiteit Leuven, Belgium

PUBLICATIONS AND PRESENTATIONS

Communicative Learning with Natural Gestures for Embodied Navigation Agents with Human-in-the-Scene. **Qi Wu**, Cheng-Ju Wu, Yixin Zhu, Jungseock Joo. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021

Ges-THOR: Communicative Learning with Natural Gestures for Embodied Navigation Agents with Human-in-the-Scene. **Qi Wu**, Cheng-Ju Wu, Yixin Zhu, Jungseock Joo. In *International Conference on Computer Vision (ICCV) Workshop on Simulation Technology for Embodied AI (SEAI)*, 2021

CHAPTER 1

Introduction

What we call meaning must be connected with the primitive language of gestures.

—Ludwig Wittgenstein [1]

Communication is an essential component in society for mutual understanding between individuals or groups, and providing multi-component signals in multiple modalities resides naturally in all animals [2], including humans. Human communicates not only with audio-visual integration but also accompanied by non-verbal cues like gestures [3, 4]. Gestures play a significant role in human-human communication. It contextualizes languages and serves as an emblem to substitute or coexist with speech or text. Indeed, human communication is rooted in fundamental cooperative gestures [5]. Gestures enhance human communication in diverse forms of social settings, including interactive speech, expressing semantic signals, and drawing the attention of recipients by disambiguating complex contextual information and assisting speech with additional clues [6].

Communication is also one of the key features of autonomous robots and systems. Analogous to human-human communication, human-robot communication could also be realized with multimodal cues [7, 8]. Natural human-robot communication, just like natural human-human communication, would reply heavily on the natural languages, either verbally or through text. Therefore, most existing communication interfaces assume that the artificial agent and the human share joint attention with the only verbal exchange. However, a robot could use various communication cues to conduct conversations with a human on the attended object [9]. This is especially important for some collaborative or assistive human-robot systems where nonverbal means of communication, *e.g.*, gestures, are indispensable [9, 10]. Different from natural languages, which require dedicated attention and suffer from intermittent conveyance, nonverbal cues, including body postures, eye gazing, and gestures, convey messages in an immediate and intuitive manner and are less susceptible to interruptions. In view of that, understanding how gestures are incorporated in HRI is crucial toward building AI systems with humanlike reasoning and cognition.

With a paradigm shift from "internet AI" to "embodied AI" [11], there is a surge of interest for HRI in simulated environments that faithfully replicate the real world. Those simulators serve as virtual testbeds for their counterparts in reality and are more affordable and salable compared with deploying the physical robots and objects. Therefore, we have seen extensive studies on HRI conducted in different simulators, *e.g.*, Vision-Language Navigation (VLN) [12], Embodied Question Answering (EQA) [13], Interactive Question Answering (IQA) [14]. By instructing with multimodal cues, we see how the intelligent agents complete a series of tasks and facilitate the understanding of HRI with learning or non-learning methods. Despite the prosperity in HRI and embodied AI, most research tasks are only concerned with verbal cues [12, 13, 15–24]. They boil down to transcode sequence-to-sequence natural language instructions in visually grounded scenes. In the vision-language domain, there are millions of speech audio datasets on the Internet for data-driven approaches. However, nonverbal signals have not yet been well documented [25]. For the embodied navigation, rich photo-realistic environments [26–30] sprout with the rapid development of Computer Vision (CV) and Natural Language Processing (NPL). Researchers from the community have the opportunity to conduct experiments with plenty of visual and verbal signals. However, there lacks a comprehensive environment and dataset that accommodate non-verbal means of communication despite the importance of non-verbal communication in HRI. Admittedly, verbal cues arise naturally and constitute the primary communicative interfaces in human-human and human-robot. Nevertheless, non-verbal means of communication should also take a significant part in embodied AI for a transition to "cognitive AI" [7], where intelligent agents are capable of understanding human intents in multiple ways. We need an environment where the human user and the robot could interactively communicate and a dataset with natural human gestures that could be utilized for embodied navigation.

To illustrate the significance above of non-verbal communication in embodied AI, let us take Fig. 1.1 as an example where the embodied human and robot are situated in a virtual kitchen. In this example, a human intends to instruct the robot to navigate to the target location or object in the scene. Previous works in embodied visual navigation with language-based human interactions may require a lengthy text message, such as "go to the second brown chair next to the big white table in the living room". Of note, it is necessary to include the keywords to avoid ambiguities. On the contrary, gestures allow the human to express the intent in a simpler and more natural way, *e.g.*, pointing with a finger and saying "go ther", "clean her", or "bring it to me". The non-verbal cues contextualize the verbal communications but are more immediate and succinct compared with using languages solely. Since the target is attended by both the speaker (*i.e.*, human) and the recipient (*i.e.*, robot) in the same space, sharing this *joint attention* [31] allows the multimodal message to be interpreted correctly by the robot who infers the human intents by mutually understanding the underlined semantics of gestures and languages.

Inspired by the above crucial observations, we intend to bring in nonverbal communication cues [25, 32, 33] into the embodied AI learning, especially the embodied agent navigation task—the most straightforward study of embodied AI that interacts with the environments and other agents. Specifically, we aim to use gestures to communicate with an embodied agent to navigate in a virtual environment. To provide gesture-based instructions for a navigation task, the agent needs a photo-realistic simulation environment, and a human player needs to be situated in the same scene to have the *joint attention* [31], and the agent requires a learning framework to complete the assigned tasks. Although human gestures have been widely adopted in human-robot systems to assist or collaborate with humans [34–38], prior studies mostly predefine the gestures according to their meanings in our social interactions. In

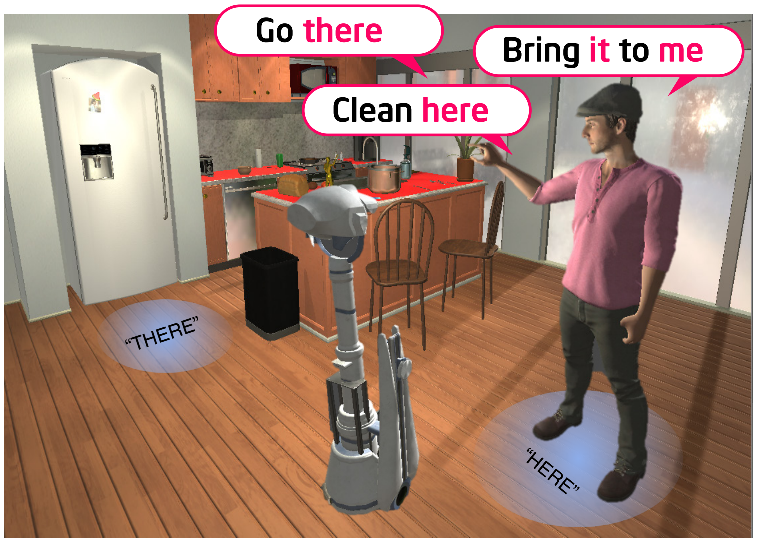

Figure 1.1: **Natural gestures can succinctly deliver complex semantic meaning in a physical space.** A human user can instruct a robot or a virtual agent to complete a navigation task by simply referring to a target location or object using gestures. The agent should infer the user's intent from the gesture.

their settings, gestures are tagged with determined labels regardless of the context changes. However, natural gestures are diverse and context-dependent [39]. The cognitive agent should learn to not only classify and cluster gesture patterns but reason *why* gestures are shown. However, learning the semantics of gestures is astounding because gestures are so diverse and context-dependent. Traditional machine learning methods are infeasible to figure out the meaning of gestures in a huge space. There requires a comprehensive framework that could address the variability and versatility of nearly-unlimited naturalistic human gestures without a predefined set of recognizable gestures.

In this dissertation, we explicitly talk about how to address the aforementioned problems. To support such a coexisting environment, we build our virtual environment with Oculus Rift, Microsoft Kinect v2, and Leap Motion Controller, based on the existing room environments from AI2-THOR [40]. Without defining any gestures and their meanings by ourselves, we collected demonstrations from a group of volunteers who have diverse gesture preferences for the same message to constitute our dataset. We also added learning modules supported by RL so that the agent could use acquired information from the environment as well as gesture instructions to complete the navigation tasks. In our proposed framework, an agent should therefore solve two tasks: multimodal target inference and navigation. Inferring the meanings of human gestures and finding a path to the target location are two major goals of the agent, which mutually help each other, *i.e.* ***learn to communicate and communicate to learn.*** Details can be found in Chapter 3 and Chapter 4 as well as our publish paper [41].

This dissertation aims to explore the new dimension for embodied AI learning, which is using a non-verbal communicative interface for human-robot interaction. We have made our dataset and learning framework publicly available so that it can be utilized by the community. This exploration could benefit the existing work with embodied navigation and other kinds of learning and bring more possibilities for reasoning and understanding how humans and robots could communicate with each other.

CHAPTER 2

Background & Motivation

In this chapter, we will give a literature review on the prior works in the closely related areas and show our motivation behind this dissertation. We focus on the following part: in Section 2.1, we talk about why gestures are important and how gestures are applied to multimodal communication. In Section 2.2, we review the development of HRI and how gestures are incorporated in HRI. Lastly, in Section 2.3, we look at the progress of embodied AI learning, including simulated environments, VLN, and RL for visual navigation.

2.1 Gestures in Multimodal Communication

Human communicates with the world inherently multimodal [42, 43] grounded in our sensory systems. We use different channels to acquire information through various senses, *e.g.*, reading texts or images through vision, listening to audio through hearing, and feeling the temperature through touching. Exploration of multimodal communication has been a pivotal topic in human cognition [44], and it has drawn attention from people from a plethora of fields, *e.g.*, psychologists, sociologists, linguists, statisticians, and computer scientists. Because human exchanges language information primarily through face-to-face interactions, we often share the *joint attention* for scene understanding [45], which could be explained by both parties engaging in the communication both attending to the same object [46, 47]. Therefore, human speech is often accompanied or replaced by body language served as visual cues such as gestures, postures, eye gazes, and facial expressions [43] to manage this joint attention in a multimodal way [48]. Evidently, gesture, as an unconventionalized and

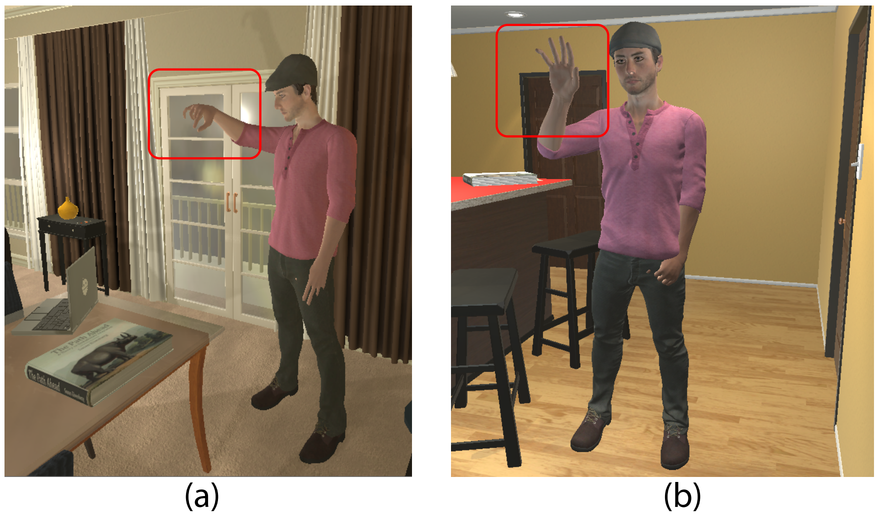

Figure 2.1: **Different types of gestures used in our work.** Two types of gestures are used in this dissertation: (a) deictic or referential, where the human points with a finger, (b) metaphoric or symbolic, where the human shows gestures in a rejective manner. Note that we have two types of classification methods in regards to human-human communication [53] and human-robot communication [54].

uncoded form compared with languages that have a definite symbol [49], is closely bonded with human language development [50, 51].

To understand the importance of gestures, we need to first know how gestures are involved in our communication. According to Susan [52], gestures contribute either directly or indirectly in the process of understanding cognitive change. Susan suggests that, by contributing indirectly, gestures can help the speakers to express their intents by displaying the unspoken thoughts or ideas, *e.g.*, we often emphasize the key points by moving our hands or arms in short strokes. The listener would be able to reckon on the gestures and associate it with the context. In this dyadic communication, both the speaker and listener can benefit from gestures in an active way of speaking and thinking interactively. Gestures can also play in a direct way to take over speech and act as a whole representation, *e.g.*, we refer to a target location by pointing with a finger instead of explicitly describing the context of the location. This especially helps when verbal expression is not available or less convenient.

Following the terminology used by McNeil [53], gestures can be classified into 4 different types (1)*deictics*, (2)*beats*, (3)*iconics*, and (4)*metaphorics*. Deictics refer to the jointly attended object by pointing toward it. Beats are repeated motions of hands or arms to show the importance of the speech. Iconics are used to depict the concrete object, while metaphorics can make abstract concepts visualized. Different types of gestures are of utmost importance to co-occur with a speech in an assistive or representative manner. In this dissertation, we focus on deictics (see Fig. 2.1(a)) and metaphorics (see Fig. 2.1(b)). Deictics are the most straightforward way to use gestures as joint attention on the object, such as pointing. This is what differentiates humans from wild animals in terms of gestures in the sense of cooperative communication [39]. Metaphorics can be used to visualize abstract ideas. They are both great candidates for gestures in embodied navigation which we will cover in Section 2.3.

To date, researchers have attempted to understand gestures from different perspectives spanning computer science, psychology, linguistics, and cognitive science. Early studies rooted in the psychology of human-human interaction analyze the proxemics and kinesics of gestures, studying the spatial and temporal characteristics [55, 56]. Those works use manual behavioral coding procedures that are several decades old [57]. With the advancement of computer vision and deep learning, we see more and more computational works with gesture recognition or motion forecasting. In terms of gesture recognition, vision-based method is the most prevalent with the Convolutional Neural Network (CNN) [58–62] either in 2D or 3D. For motion forecasting, many approaches address the future poses prediction using state-of-art deep neural networks and recurrent units [63, 64]. In this dissertation, unlike other works contributing to gesture type classification [58], we don't predefine gestures—we want the embodied agent to uncover the semantics via the learning network. We hope the human gestures collected will only have the spatial indication or conventional signals where the intent and context are known to ourselves, and it is the agent's task to reach "common ground" [5]. We will cover more detailed gesture recognition technics in Section 2.2.

To pursue a data-driven approach to gesture understanding, it is also important to measure and collect data with non-verbal communication information. There are many datasets enclosing rich 3D body motion information acquired from videos or advanced motion capture technics [65, 66]. A group of works in recent years also capture gesture datasets in a social interactive setting [67]. Despite the prosperity of gesture datasets, there is no comprehensive non-verbal dataset for embodied AI learning to our knowledge. We devise the motion capture procedures to generate a dataset with both body motions and embodied tasks that could be directly used in an embodied learning setup.

2.2 Human-Robot Interaction with Gestures

HRI has been intensively investigated in AI, Human-Computer Interaction, and robotics. HRI systems can either use human supervisory controls or work in an automated way, which help human complete routine tasks, reach inaccessible areas, or engage in social interactions [68] in diverse settings like home, healthcare, education, manufacturing, agriculture, mining, aviation, and so on [69]. We see teleoperated robots, which are directly and continuously controlled by humans defined by Sheridan [70], work in a wide variety of situations ranging from rehabilitation and surgery in healthcare to rescue tasks in disastrous hazards. We have also noticed that semiautonomous or autonomous robots are being applied widely from domestic to underwater to space [69, 71]. Those robots either work individually in a single robot system or collaboratively as a multi-robot system [72]. We endow intelligent robots or systems, humanoid or otherwise, with humanlike behaviors and expect one day there will be fully autonomous robots like the famous movie character WALL-E.

As said, we are making more and more intelligent robots. In , we have the transition from considering the robots as tools to making them our companions and teammates by assigning them more cognitive and social meanings [73]. In other words, we witness dramatic changes in human-robot communication as the dynamics of the human-robot interaction change. Early human-robot communication could be addressed as Human-Computer Interaction (HCI).

We operate a wide variety of media to control the robot directly or indirectly [74]. With modern AI approaches, we are able to view the communication as a language-based multimodal interaction [69], which makes it a mirror of human-human communication. Recent advances in Machine Learning (ML) solutions, *e.g.*, NPL, promise to make the robot capable of interpreting and responding to human languages [75]. However, using verbal cues merely in HRI is not able to cover every aspect. Vocal communication is closely related to human languages, which are developed through explicit social and cultural experiences and suffer from the intricacies and ambiguities that are inherent in the real world [76]. These difficulties hinder the development of NLP in vocal HRI.

Nonetheless, similar to nonverbal communication as inherited behaviors in human [77], we could resort to the nonverbal communicative interface, *e.g.*, gesture-based interface, in HRI as well. We have seen gesture or haptic control on robots via hand-wearable device [78] or HMD in a VR or Augmented Reality (AR) setting [79], and it provides us with more dimensions to contemplate on how we can manipulate robots in a more natural and multimodal way and how robots can facilitate us collaboratively. Ideally, those interfaces should work seemingly comparable to human-human team works. To understand how gestures influence human-robot communication, we could connect it with the importance of gestures in human-human communication as we have discussed in Section 2.1. Human understands gestures by acquiring information with vision first and then processing the information with our cognitive systems. Similar to how humans interpret gestures, we could also understand gestures in HRI from those parts according to [80, 81]:

- Collections of raw gesture data by sensors
- Identification and tracking of gestures
- Interpretation of recognized gestures
- Robot reaction to gestures

Collection of raw gesture data: Before the recognition part of gestures, the robot needs to "see" the gesture. There are image-based and non-images-based approaches for the perception of gestures [80]. The image-based method is inspired by how human beings naturally see gestures. Optical cameras like single camera [82] or stereo camera [83] are used for a robust construction of gestures. However, traditional optical systems suffer from computational complexity and calibration difficulties [84]. With depth-sensing technologies developing rapidly, depth sensors are favored in terms of gesture recognition. Compared with traditional stereo cameras, depth sensors neglect the color information and simplify the data to 3D depth information. It is a cheaper and easier solution for gesture recognition and less prone to errors. However, because most depth sensors are resolution limited [80], it is usually applied for body gesture recognition or hand gesture recognition in short distances. Non-image-based approaches have also gained a lot of attention with different sensor technologies. One branch is using wearable devices. Liu *et al*. [85, 86] has developed a glove-based system to study human-object manipulation in AR and VR environments. Another branch is using non-wearable devices. This type of technology does not need to directly contact the human body and has the capability to get higher-precision hand gesture data in a short distance. For example, Google has introduced Project Soli, which is a Radio Frequency (RF) signal-based hand tracking system [87]. In a recent work by Li *et al*. [88], they utilize two different sensors, Microsoft Kinect v2 and Leap Motion, to jointly integrate gesture data combining both image-based and non-image-based approaches. Kinect v2 uses Time-of-Flight (ToF) technology to capture body language, while Leap Motion is used to detect close-distance hand gestures with IR light. Our data collection process is similar to the work presented in [88], and we also want to use both of the sensors to provide holistic information on both body and hand gestures.

Identification and tracking of gestures: The next step in a gesture recognition process is how to detect gestural information in each frame and find the temporal difference between frames. For gesture identification from raw data, it is natural to also use visual features. We could use check the color [89], local features [90], or shape and contour [91] on a single image to identify a gesture. A recent trend of gesture identification also uses ML algorithms to spot gestures with raw sensor data. Those methods usually do not need background removal and only reply to the learning algorithms [80]. For example, Tang *et al*. [92] applies Random Decision Forest (RDF) in their 3D hand posture recognition systems. Another type of technology that has become extremely popular in recent years, especially with the development of depth sensors, is utilizing skeleton models. As we may notice, it is unnecessary to identify or track every part of the part in order to interpret gestural information. Point cloud information from depth sensors is largely redundant, and it can be simplified to a skeleton model that provides the most valuable information. For example, some novel works on recognizing sign languages adopt skeleton-based methods as their identification systems [93, 94].

Interpretation of recognized gestures: The last but the most important step in the gesture recognition process is how we interpret collected and tracked gestures. It reminds us of how humans view gestures in the context of interactive communication. From the robot's point of view in HRI, this boils down to *what* humans are doing and *why* humans they are doing it [54]. It is not practical to devise methodologies entirely based on the psychology of human-human communication, and we should give more insights on the spatial and temporal perspectives (*i.e.*, proxemics and kinesics [56]) in regards of HRI to help the robot infer the *intent* of human gestures.

To facilitate the understanding of gestures, there are a plethora of computational studies on gesture recognition to study how gestures are incorporated as a part of the communication system [58–62, 93–96]. In terms of understanding *what* humans are doing, we need to do gesture classification. We could utilize different ML models to either classify gestures into various categories. Different methods ranging from traditional learning algorithms like K-Nearest Neighbors (KNN) [97], Hidden Markov Model (HMM) [98], and Support Vector Machine (SVM) [99], to emerging deep learning techniques like CNN and Recurrent Neural

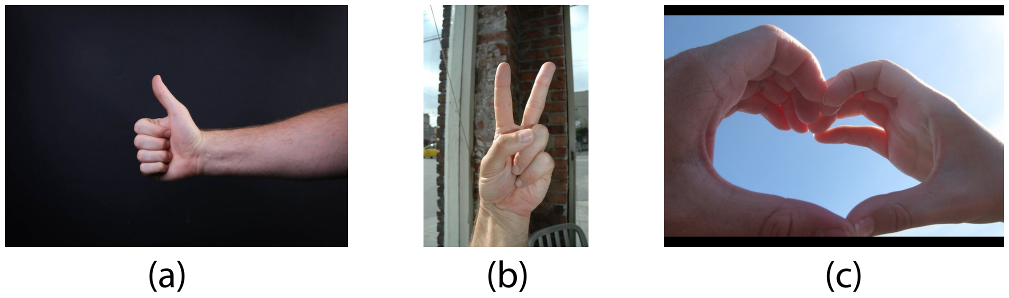

Figure 2.2: **Some typical predefined symbolic gestures.** Humans use some conventionalized signals in interactive communications [54]. Here are some examples: (a) We raise up our thumb to someone who does a great job. (b) We show index and middle finger to mean victory. (c) We use both hands to make a shape of heart to show love.

Network (RNN) [100] with increasing computation power. As to understand *why* humans show a specific gesture, this requires the robot to be endowed with humanlike cognitive intelligence. We have talked about gesture classification in Section 2.1 based on [53] with respect to human-human communication. According to the prior work by Nehaniv *et al*. [54], we could classify gestures in the context of HRI and human intents into those categories: 1) 'Irrelevant'/Manipulative Gestures, 2) Side Effect of Expressive Behaviour, 3) Symbolic Gestures, 4) Interactional Gestures, 5) Referential/Pointing Gestures. For symbolic gestures, we use our conventionalized signals to classify gestures into different categories in order to correctly interpret them (see examples in Fig. 2.2). For other types of gestures, we may need to predict the human intent. For example, referential or pointing gestures are used to refer to the target location or object of interest. Therefore, it is crucial for the robot to predict the attended object or location from hand/finger motion [101] or eye gaze [102]. It is worth noting that most of these approaches are based on a predefined gesture set with fixed meanings or focus on gesture type classification [58], pose estimation [59, 60], or both [61, 62]. In contrast, we let users use any natural gestures and demonstrate the agent can directly learn semantics and underlying intents of these gestures.

Robot reaction to gestures. Interacting with human-aware behaviors for robots is a challenging area in HRI [103]. This could be understood as *how a* robot should react to gestures. Expressing their own intents is as significant as interpreting human intents for a socially interactive human-robot system. We have seen recent work in human-robot communication seek to address this challenge in a variety of approaches, *e.g.*, the generation of intent-expressive motion [104] and explicable task plans [105]. Nevertheless, a majority of the work is focused on physical robots. Physical HRI affords high risk and cost, and simulation could address such problem [106]. AR and VR are emerging in HRI, and they are attractive to simulation human-robot behaviors. As a more affordable way to deploy and train robots, many researchers choose to do extensive experiments in a VR environment before moving to a physical device [85, 86, 106]. Our framework is also a VR-based HRI system using Unity3D and Oculus Rift, and we hope that more extensive research could be conducted under this framework. More information about simulation in HRI is covered in Section 2.3, which is known as Embodied AI.

2.3 Embodied AI Learning

Embodied AI, which means "AI for virtual robots", finds a tangible intelligent robot or system that humans could interact with. Different from internet AI that learns from static datasets of images, videos, and texts [107, 108], embodied AI is faithful that true intelligence can emerge by interacting with the environment [109] egocentrically just as how humans sense the world. Embodied AI is, therefore, at the intersection of vision, language, and reasoning for an intelligent embodiment in simulation [11].

Echoing the idea in [7], we intend to bring "common sense" to embodied AI, *i.e.*, we want AI to not only have embodied physical abstraction but also possess humanlike cognition. Such cognitive AI can have multimodal communication with the human in simulation, including non-verbal means of interaction. Additionally, it should think more on *why* instead of *how* in understanding human intent for HRI.

In this section, we build upon the most recent survey on embodied AI [11], which is the most comprehensive survey that investigates embodied AI to the best of our knowledge, and include some most recent works to show an overall picture in this field. In Section 2.3.1, we survey some most prevalent embodied AI simulators and benchmark them from different perspectives. In Section 2.3.2, we review different research tasks on those existing simulators. Finally in Section 2.3.3, we provide that how RL is applied for embodied AI learning. Additionally, we want to demonstrate how our work is related to the prior literature by reviewing all aspects of embodied AI.

2.3.1 Embodied AI Simulators

To help the research in embodied AI, various simulated environments have spurred for the community's benefit (see Table 2.1). Different from the earlier environments like DeepMind's Arcade Learning Environment [124] or OpenAI Gym [125], which are designed for general agents, they are of critical importance for research on the intelligence with an embodiment where the agents have humanlike sensibility and consciousness. Those environments are curated for different purposes in embodied AI, from benchmarking RL tasks in a simple 3D game platform [110] to evaluating robotic manipulation tasks with fine-grained physics [126], and we have seen a growing interest and effort in benchmarking advanced deep learning, reinforcement learning, and robotics in a general-purpose platform.

In Table 2.1, we summarize the provisions of some most recent simulators for different features in order to give a comprehensive look at the state-of-the-art embodied AI platforms. This is a more thorough summary compared with existing surveys on embodied AI simulator [11, 40, 118]. Despite BabyAI, which is designed uniquely for grounded language learning with a 2D setting, most simulators are constructed in a 3D environment. Those 3D environments are created from either synthetic scenes [40, 112–115, 118, 121, 127] or real photographs [26–28, 112, 113, 119]. For simulators with photorealism, they need real-world scans with software in portable devices or depth sensors. They are literally a replica of the real world and therefore retain higher fidelity, benefitting a Sim2Real transfer. There are existing datasets scanned from real-world like SUNCG [122], Matterport3D [29], Gibson [27], and

Replica [123]. In contrast, simulators without photorealism are generally built with game engines like Unity3D or Bullet. They use objects from 3D assets like PartNet [128] and Unity Asset Store. They need fewer resources and are more customizable compared with world-based simulators.

Another important feature of a simulator is its physics. To mimic a realistic environment, physical interactions between objects are also important for embodied AI to study interactive navigation or language-based tasks. Several simulators have enabled simple or advanced physics as a part of the environment using simulation engines like Unity3D or Bullet [40, 114, 115, 115, 118, 127]. Simple physics include rigid-body dynamics and kinematics. They can simulation behaviors like collisions between objects [112]. Some simulators, *e.g.*, AI2-THOR [40] and VRKitchen [116] provide both object interactions and object state changes on top of that. Advanced physics, *e.g.*, soft-body dynamics, are more intricate. Simulators like ThreeDWorld [118] are capable of emulating those complex physics in their environment. More advanced features can support more complex embodied AI tasks, facilitating research from more dimensions.

Despite the sprout of various simulators, we see the deficiencies in relating embodied AI simulators with HRI, especially with non-verbal cues. There are many simulators with robotic embodiment [40, 114, 120], but they are mostly concerned with visual navigation tasks and neglect human factors. Although VLN [23] and EQA [13] tasks frame HRI in a language-based setting, only verbal cues are considered. There are simulators with human interfaces that allow HCI with non-verbal features. For example, VRKitchen [116] and ThreeDWorld [118] both have VR-based human-centric interactions with the environment. Human operators can use VR devices to pick objects or change object states. However, both of them are human-centered. Even though ThreeDWorld has robotic embodiment, it does not take the significance of the robot in the interactions, *i.e.*, *how* robots understand human intents. For this reason, we build our own simulator based on AI2-THOR [40] and extend current embodied AI simulators with support for non-verbal HRI [41].

2.3.2 Embodied Visual Navigation

2.3.1 covers all kinds of simulators which are the basics for conducting embodied AI experiments. The motivation to build the aforementioned simulator is to raise awareness in cognitive AI research with an embodiment. Humans learn to know everything mostly by interacting with the surroundings with the sense organs to update our belief of the world. Likewise, robots, or intelligent agents in a general sense, also require perception, interaction, communication with the whole world as a result of sensorimotor actuations [129]. From an embodied AI viewpoint, this boils down to different tasks commonly seen in robotics like *embodied navigation*, *embodied interaction*, and *embodied learning* [11].

There is not a canonical taxonomy for embodied AI tasks yet from prior literature. We can group the tasks in four main types which are *visual exploration*, *visual navigation*, *embodied question answering*, and *visual interaction*, echoing a similar classification in [11]. *Visual exploration* is a primitive yet important task for embodied AI [130–133]. It is task-agnostic, and the agent needs to move around in an open environment to collect information for downstream tasks [134]. *Visual navigation.* The objective is often considered as the coverage of the unseen environment [130] or building the internal model of the environment through the exploration [133]. Contrary to visual exploration, *Visual navigation* is accompanied by a determined task, either a point, an area, or an object [135]. The agent is required to reach the proximity of the target by navigation in an unknown environment. This poses more challenges on the agent as it needs to understand the environment semantically via different modalities [12, 21, 28, 136–140]. *Embodied question answering (EQA)* is to perform Question Answering (QA) in an embodied setting [13]. The agent needs to harness a variety of its AI capabilities, such as visual perception, language grounding, and cognitive reasoning, to complete the QA along with some concurrent tasks like navigation. *Visual interaction* refers to tasks that require the agent to interact with the environment [30, 126]. The task is often manifold that the agent seeks to complete a series of tasks in order to reach the end goal through interactions with objects. Both *EQA* and *visual interaction* want the agent to

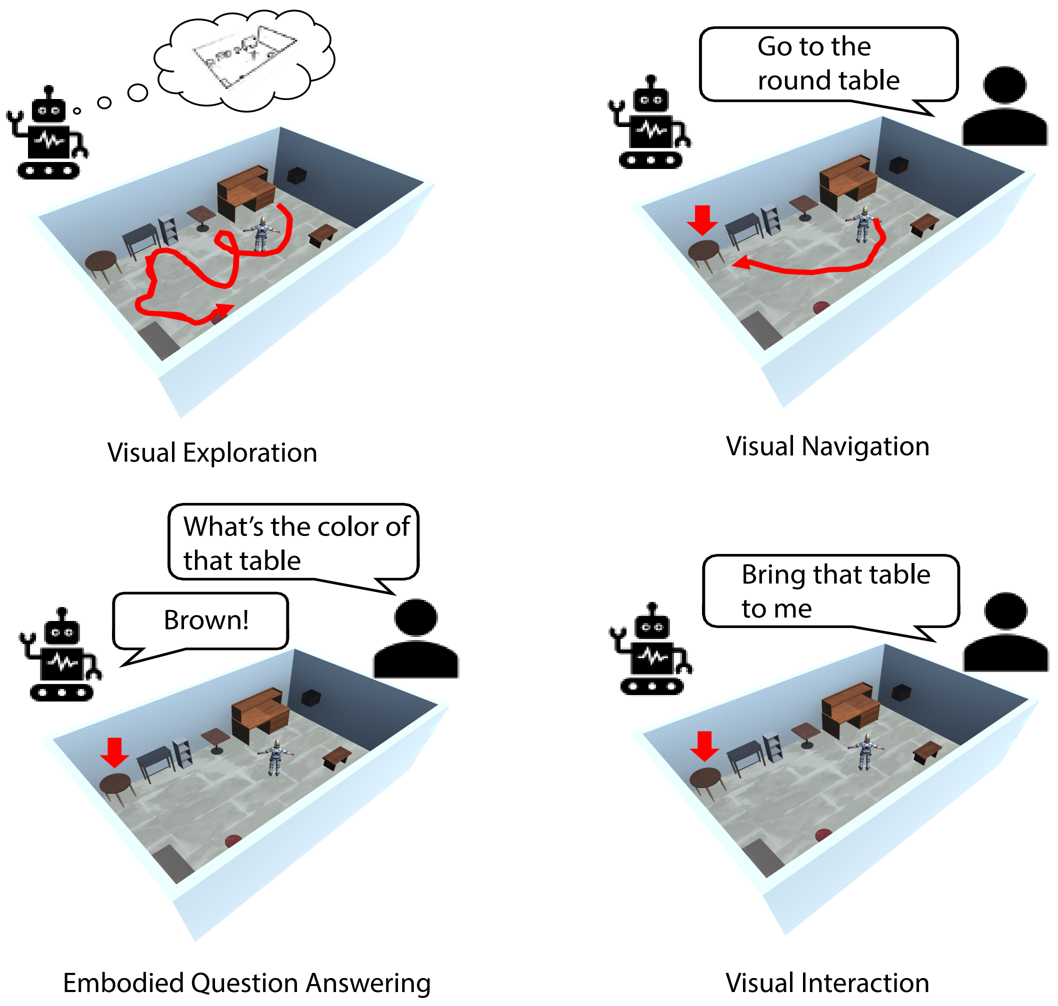

Figure 2.3: **Different research tasks in embodied AI.** This is a visualization for four main types of embodied AI research. In this example, visual exploration is target-agnostic, and the robot needs to build a representation of the room. For the other tasks, the robot is given a target table and asked to either navigate to the target, answer a question concerning the target, or interact with the target.

"interact" in general, either with objects or humans, making the learning task more onerous and challenging. See Fig. 2.3 for a visualization on different tasks.

In this dissertation, we will focus on the *visual navigation* task, which is the most fundamental task for embodied AI. As seen in Fig. 2.3, EQA and visual interaction are often downstream tasks of visual navigation, *i.e.*, the agent must find the target in order to complete the subsequent tasks. Animals are capable of exploring unseen environments and pursuing distant objects even in a cluttered environment. Navigation is also of central importance to mobile intelligent systems [135] to adapt to a new environment before proceeding with advanced behaviors [141]. Hence, constructing an intelligent agent in a simulator that can successfully and efficiently navigate to the target is pivotal for embodied AI as well.

According to the recent surveys on embodied navigation [135, 142], visual navigation can be classified into PointGoal, ObjectGoal, and AreaGoal navigation based on the nature of the goal. PointGoal navigation requires the agent to navigate to a specific location. ObjectGoal and AreaGoal navigations task the agent to navigate to a semantically labeled target object or area. Notably, PointGoal and AreaGoal navigations provide spatial information, and both ObjectGoal and AreaGoal provide semantic information. Among them, PointGoal and ObjectGoal navigations are the most common tasks for embodied AI. Yet PointGoal navigation has drawn much attention by incorporating different methods [28, 136, 143, 144], *e.g.*, [136] utilizes a modular structure to update the confidence and belief of the world while moving toward the target. The task is rather simplified against other tasks. The target is a point in space, often represented by a coordinate, and the focus is on episodic memory construction for the unseen environment, similar to *visual exploration*. On the other side, ObjectGoal navigation (or ObjectNav [142]) is defined as navigating to a specified object. Beyond the skillsets possessed by PointGoal navigation, such as visual perception, episodic memory construction, mapping, and planning, ObjectNav also requires cognitive reasoning and semantic understanding, making it more challenging. Besides, since ObjectNav has a concrete object as the target, it is considered as the upstream task for many downstream tasks that are more complicated such VLN, EQA, and so on.

While ObjectNav has grown interested in the embodied AI community, it requires a consensus on the task definition. [142] and [9] has made great contributions and led to a convergence on ObjectNav. One ought to accurately define a high-level depiction of the task: *What* do we consider as successful navigation? *What* is the embodiment for the agent? *What* environments do we use? Those questions constitute a comprehensive understanding of the task, and we can test and evaluate our agents with the well-established definition. We will cover more details in Section 4.2 with respect to our own problem.

Visual navigation boils down to a motion planning task in an unseen 2D or 3D environment. From the point of view of traditional robotics, it can be decomposed into several sub-tasks as mapping, localization, planning, and locomotion. In terms of mapping and planning, there are a plethora of works Simultaneous Localization and Mapping (SLAM) focusing on building the metric map using geometric information [145]. Therefore, we could see some visual navigation tasks solved with this non-learning method. For example, [140] adopts advanced algorithms from SLAM field to build the local map and localize the agent from the depth sensor. Afterward, a D* Lite algorithm [146] is employed to accurately calculate an optimal path from the previously built obstacle map and then use a proportional-integral-derivative (PID) controller to generate discretized locomotions. The non-learning method often requires the geometric scanning of the environment to reconstruct the space, which means that extensive hand-engineering is involved and sub-tasks are necessary, making it susceptible to sensor noises. In stark contrast, a learning method could exploit regularities in the data and train a model in an end-to-end manner while preserving competitive performance [143], leading to a consistent behavior even in downstream tasks. Thanks to the rapid development of deep learning [147], a surge of interest in using learning-based methods is found in the embodied AI community [12, 21, 28, 130–134, 136–140]. Learning can be done either with an end-to-end model [122, 139], or with modular structures [136, 138]. Some works also add auxiliary supervised or unsupervised tasks to assist the entire learning process, *e.g.*, building an occupancy map and comparing it with the ground truth [133]. Besides, we see humanlike knowledge like scene prior added to endow the agent with human awareness [148]. It is worth noting that, among all kinds of learning methods, RL, or specifically Deep Reinforcement Learning (DRL), is widely used for the navigation task. Embodied navigation does not include clearly labeled or grouped features to be applied with supervised learning and unsupervised learning. However, RL only concerns with state-action transitions and it helps us to complete AI-driven navigation tasks (refer to Section 2.3.3 for deeper dive on RL in embodied navigation). Considering the pros and cons of learning and non-learning methods, some works contribute to a hybrid solution that combines both approaches [136, 138]. For example, Chaplot *et al.* [138] predict the high-level map with a Neural SLAM module and generate low-level navigation policies with a classic pipeline.

Visual navigation with a target can be further integrated with other modalities like perceptual inputs, language instructions, and audios to either facilitate finding the target or build toward more complex tasks [11], such VLN [12], EQA [13], IQA [14], Audio-Visual Navigation [149], and Visual Interaction [150]. From the most recent works, using verbal cues is a hotspot in visual navigation, which can formulate the task as grounding the language in sequence-to-sequence visual information. Language grounding is crucial for both parties involved in communication to understand each other. Natural language, the most common modality for human-human and human-robot communication, can realize the grounding in various ways. For communication with robots, language can be interpreted from instructional commands to actions [151, 152]. For static images or texts, it can be either visually grounded [153, 154] or text-based [155] Q&A. In our work, language grounding is replaced by "gesture grounding;" we provide gestures as the new communicative interface. The agent is tasked to learn by grounding human gestures into a series of actions and identifying target objects.

Image captioning with large datasets [156] and Visual Question Answering (VQA) [153, 157] has made significant progress in vision and language understanding, which enables visually-grounded language navigation agents to be trained. Many tasks following the VLN framework [12, 13, 15–24] have been addressed and solved using end-to-end learning models, either in 2D world [117, 158, 159], 3D world [16, 160], or even photo-realistic environments [12, 19, 20, 23, 127]. Our work is built on the existing VLN framework but extended by incorporating gestures as a new modality for communications.

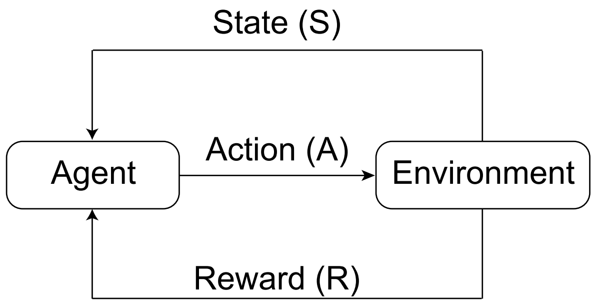

Figure 2.4: **Schematic diagram of reinforcement learning.** At each time step t, the agent has a state representation of the environment. It selects an action and gets an updated state with a numeric reward.

2.3.3 Reinforcement Learning for Embodied Learning

As we have discussed in Section 2.3.2, instead of using traditional path-planning approaches [161] to compute a route to the goal location, the embodied AI community has recently focused more on end-to-end learning for training navigation policies, especially with RL. In RL, The agent needs to learn a policy to maximize the cumulative reward by interacting with the environment. Compared with other machine learning methods, such as supervised learning [162], RL benefits from simple reward definitions and easy implementations. As a result, RL becomes the core of many learning frameworks [12, 27, 28, 136, 139, 148].

RL has become popular in recent years because of the human-level performance on Atari games [163] and the success of AlphaGo [164]. More and more researchers in embodied AI community are also devoted to applying RL in mapping, planning, and reasoning. Reinforcement learning is a Markov Decision Process (MDP) [165]. The agent interacts with the environment with discrete time steps, $t = 0, 1, 2, 3....$ For every time step t, we consider the setting of a finite-horizon discounted MDP:

$$\mathcal{M} := (\mathcal{S}, \mathcal{A}, p, r, \gamma), \tag{2.1}$$

where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is a finite action space, $p(s'|s, a)$ is the transition function, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the reward function, and $\gamma \in (0, 1)$ is a discount factor. A state is called Markovian iff it has the following characteristics:

$$p(s_{t+1}|s_t, a_t) = p(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}...s_1, a_1), \tag{2.2}$$

which indicates that the next state s_{t+1} is only dependant on the previous state s_t. At step t, the controller observes a state s_t and selects an action $a_t \in \mathcal{A}^{s_t}$ according to a *policy* π, which maps a state to an available action (see Fig. 2.4 for a schematic illustration). The environment then transits to a new state s_{t+1} with probability $p(s_{t+1}|s_t, a_t)$ and the controller receives an instant reward $r(s_t, a_t, s_{t+1})$. Given a policy $\pi : \mathcal{S} \to \mathcal{A}$, we define its value function $V^\pi : \mathcal{S} \to$ as:

$$V^\pi(s) := \mathbb{E}^\pi\bigg[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})|s_0 = s\bigg], \tag{2.3}$$

where the expectation is taken over the trajectory following π. The objective of RL is to learn an optimal policy π^* such that its value (on any state $s \in \mathcal{S}$) is maximized over all policies, i.e.,

$$\forall\ \pi, s \in \mathcal{S} : \quad V^* := V^{\pi^*}(s) \geqslant V^\pi(s). \tag{2.4}$$

A policy π is said to be ϵ-*optimal* if it achieves near-optimal value for every state, i.e. $\forall\ s \in \mathcal{S} : \quad V^\pi(s) \geqslant V^*(s) - \epsilon$. The action-value function (or Q-function) of a policy π is defined as

$$Q^\pi(s, a) := \sum_{s' \in \mathcal{S}} p(s'|s, a) \cdot \big(r(s, a, s') + \gamma V^\pi(s')\big). \tag{2.5}$$

The optimal Q-function is denoted by $Q^* := Q^{\pi^*}$. By Bellman Optimality Equation, we have:

$$V^*(s) = \max_{a \in \mathcal{A}^s} Q^*(s,a) = \max_{a \in \mathcal{A}^s} \sum_{s' \in \mathcal{S}} p(s'|s,a) \cdot \big(r(s,a,s') + \gamma V^*(s')\big). \tag{2.6}$$

RL algorithms can be grouped into three categories: value-based method, policy-based method, and actor-critic method [141]. The value-based method aims to estimate the value function at a given state according to Eq. (2.3), and the agent selects an action with the maximum state-action value. Q-learning [166] is widely used as a value-based RL algorithm. It updates Q through the Bellman formula:

$$Q_{i+1}(s,a) = \mathbb{E}^{\pi}[R_t + \gamma \max Q_i(s_{t+1}, a_{t+1})|s_t, a_t], \tag{2.7}$$

where Q is guaranteed to converge to its optimal value Q^* when $i \rightarrow \infty$ [167]. Traditional Q-learning updates the state-action value with the following equation [165]:

$$Q(s_t, a_t) \rightarrow Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]. \tag{2.8}$$

In Eq. (2.8), $r_t + \gamma \max_a Q(s_{t+1}, a)$ is the target value, and $r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$ is the Temporal Different (TD). α is the learning rate to have a stable update on the Q value based on the TD. This is a bootstrapping way to update the Q value. There are also methods to update Q with Monte Carlo methods or a combination of them (n-step bootstrapping) [165]. TD learning has higher bias and lower variance, while Monte Carlo methods incur higher variance and low bias.

However, Q-learning is a tabular RL solution. It is not feasible when the state and action space are continuous or large. Therefore, function approximation is used as a representation for the Q table, *e.g.*, a neural network. We can hence write $Q(s,a)$ as $Q(s,a|\theta)$, where θ is the parameter for the Q network. This leads to the transition in RL from table-based Q-learning to DRL. This idea is first introduced in [163], where they use a deep learning network with two convolutional layers as the function approximator of Q value.

Instead of indirectly computing the optimal policy, the policy-based method targets directly modeling and updating the policy. The parameterized policy can be written as $\pi(a|s,\theta)$, where θ is the policy network parameters. In order to maximize the expected return $\mathbb{E}[R_t]$, we need to perform the gradient ascent on the objective function:

$$J(\theta) = \mathbb{E}^{\pi}\bigg[\bigg(\sum_{t=1}^{T} G_t \nabla_\theta log \pi_\theta(a_t|s_t,\theta)\bigg)\bigg], \tag{2.9}$$

where T is the horizon and G_t denotes the return from time t. If the return is estimated by Monte Carlo methods, this policy gradient is called REINFORCE. In vanilla policy gradient, G_t is often replaced by $A^{\pi_\theta}(s_t, a_t)$, where A is the advantage function with $A(s,a) = Q(s,a) - V(s)$.

The actor-critic method is a hybrid version of value-based and policy-based methods, which combines the advantages of both of them. For the actor-critic architecture, the actor-network chooses the action while the critic network evaluates the action taken at the current time step. The following is one option to implement the actor-critic approach:

$$J(\theta) = \mathbb{E}^{\pi}\bigg[\bigg(\sum_{t=1}^{T} Q(s,a|w) \nabla_\theta log \pi_\theta(a_t|s_t,\theta)\bigg)\bigg], \tag{2.10}$$

where w is the parameter for the value network. We update both the policy parameters with Eq. (2.10) and the value parameters with the TD error as shown in Eq. (2.8).

RL can be either on-policy or off-policy. On-policy algorithms learn the optimal policy that is carried out by the agent that explores the environment. For instance, Deep Q-Learning (DQN) in [163] is off-policy. It has a replay buffer that collects previous trajectories of the agent to reduce the correlation between samples. In contrast, in the off-policy method, the agent explores a different policy than the learned one. Vanilla policy gradient is on-policy since the policy learned is directly used for the value iterations.

Of note, the aforementioned algorithms are reward-driven. The goal of them is to maximize the discounted accumulative rewards. However, this is challenging when the reward is sparse (*e.g.*, a reward is given only when winning or losing the game), and it can be astounding to shape the reward. One feasible solution to mitigate the problem is to use Imitation Learning (IL). IL, or Learning from Demonstration (LfD) [168], tries to encourage the agent to learn the optimal policy by following or imitating human demonstrations. In IL, the problem boils down to getting the optimal policy π^* with unknown reward functions $R(s,a)$ by the experts' demonstrations (also known as trajectories)

$$\tau = (s_0, a_0, s_1, a_1, ...). \tag{2.11}$$

There are multiple ways to implement IL. The simplest form is Behavior Cloning (BC), where the agent learns to imitate the expert behavior in a supervised manner, and the loss function is defined as the difference between learned and demonstrated policies. We can also use Direct Policy Learning (DPL) to iteratively update the policy while having access to an expert at training time. For example, Dataset Aggregation (DAgger) [169] trains the actual policy on all the previous training data. Another approach is called Inverse Reinforcement Learning (IRL). As the name suggests, this method estimates the parameterized reward function $_\theta(s,a)$ inversely from the expert policy.

In embodied AI learning with RL, the actor-critic method is more favored than the other methods in recent publications [12, 27, 28, 136, 139, 148]. Versatile experiments have testified the effectiveness of actor-critic RL algorithms, such as DQN [163], Asynchronous Advantage Actor-Critic (A3C) [170], PPO [171], and Soft Actor Critic (SAC) [172]. Although the navigation task in embodied AI is manifold, the basic RL formulation is very similar. In short, the agent needs to learn how to navigate with sparse rewards by interacting with the environment. The state S includes the egocentric visual perceptions, including color images, depth images, or even segmented images [26, 28], which constitutes the most important part of the agent observations. High-performance neural networks in CV can be used to preprocessor encode the visual information. There are also other states available to the agent, *e.g.*, target object semantic labels that are static, or language instructions that are either static [12] or dynamic [14]. As to the agent's actions, it depends on the embodiment of the agent. If the agent is considered as a point mass, the actions are as simple as *moving forward* and *turning left/right* [173]. If the agent has intricate robotic embodiment and fine-grained movements, the actions are more complex, *e.g.*, the agent needs to move its arm to interact with an object [126]. For the rewards, intuitively, a positive reward should be assigned when the agent successfully navigation to the target or complete a task. However, this reward is sparse, and only a very small portion of the possible states contribute to the success criteria. Therefore, reward shaping is often necessary to make the learning sample-efficient [174]. Also, some studies use IL to pretrain the policy network as a bootstrapping in RL using data aggregation methods [17, 136]. This helps the agent to warm up before learning and mitigate the scarcity of rewards. There are also datasets generated as expert demonstrations (*e.g.*, expert atomic actions to complete procedural tasks [30, 116]) to facilitate research on IL-boosted embodied learning.

In this dissertation, we choose PPO [171] as the RL learning algorithm. PPO is an off-policy actor-critic method. It uses a clipped objective function to ensure that the updated policy does not deviate from the old one too much. Embodied visual navigation suffers from sudden state changes, leading to rewarding fluctuations. PPO can help maintain stable improvements on the policy and resolve the fluctuation issue. For this reason, PPO is one of the most popular RL methods for embodied navigations [28, 41, 137, 139, 175].

Name	Year	3D	photo-realistic	Environment Configuration	Physics	Actionable	Simulation Engine	HRI
DeepMind Lab [110]	2016	Yes	No	Customized environment	No	No	Quake II Arena	No
VizDoom [111]	2017	Yes	No	Based on Doom	No	No	ZDoom	
MINOS [26]	2017	Yes	Yes	SUNCG, Matterport3D	No	No	-	No
AI2-THOR [40]	2017	Yes	Yes	120 synthetic rooms	Yes	Yes	Unity3D	No
HoME [112]	2017	Yes	Yes	SUNCG	Yes	No	Bullet	No
House3D [113]	2018	Yes	Yes	SUNCG	No	No	OpenGL	No
VirtualHome [114]	2018	Yes	No	6 hourseholds	Yes	Yes	Unity3D	No
Habitat [28]	2019	Yes	Yes	Matterort3D, Gibson, Replica	No	No	Magnum	No
CHALET [115]	2019	Yes	No	58 rooms, 10 hourses	Yes	Yes	Unity3D	No
VRKitchen [116]	2019	Yes	No	16 Kitchens	Yes	Yes	Unreal4	No
BabyAI [117]	2019	No	No	19 levels of complexity	No	No	-	No
ThreeDWorld [118]	2020	Yes	No	Customized environment	Yes	Yes	Unity3D	No
iGibson [119]	2020	Yes	No	Gibson	Yes	Yes	Bullet	
SAPIEN [120]	2020	Yes	No	Customized environment	Yes	Yes	PhysX	No
ActionNet [121]	2020	Yes	No	Based on AI2-THOR	Yes	Yes	Unity3D	No
GesTHOR [41]	2021	Yes	No	Based on AI2-THOR	Yes	Yes	Unity3D	**Yes**

Table 2.1: **Summary of embodied AI simulators.** We compare different simulators with several selected features. 3D: if the environment is built in 3D (otherwise 2D). photo-realistic: if it uses world-based scene (otherwise game-based scene). Physics: if it allows basic physics features (otherwise there is no physics). Actionable: if objects in the environment can be interacted (otherwise objects are non-interactable). In environment configuration, some simulators use existing world-based datasets like SUNCG [122], Matterport3D [29], Gibson [27], and Replica [123].

CHAPTER 3

GesTHOR: Simulation Framework for Interactive Embodied Agent

In this chapter, we introduce our simulation framework for the interactive embodied agent, GesTHOR, and present its novelty compared with other simulators. Fig. 3.1 is an overview of the framework. It is organized in three sections: In Section 3.1, we briefly talk about our initiatives creating this framework on top of Chapter 2. In Section 3.2, we discuss the key components in GesTHOR. Lastly, in Section 3.3, we touch on our GOND in regards to why and how we generate it.

3.1 Introduction

As the research in multimodal communication and embodied AI proceeds, the intersection of two fields emerges and raises interest in both communities. Zhu *et al*. [7] call for a paradigm shift to cognitive AI systems with humanlike common sense, and this shift also takes place with embodied AI. The most important modality for the embodied agent is vision, which is the basis for almost all embodied AI systems, just as how imperative vision is for humans. However, human uses different senses to perceive the world, and a humanlike agent should also be capable of fusing various sensory information. To this end, multimodal inputs are injected into the learning process of the embodied agent. Language instructions are most commonly used as one of the multimodal cues in numerous works [12, 13, 16, 19, 20, 22, 23, 117, 158–160, 176–178] with NPL technologies to ground them as machine-readable features.

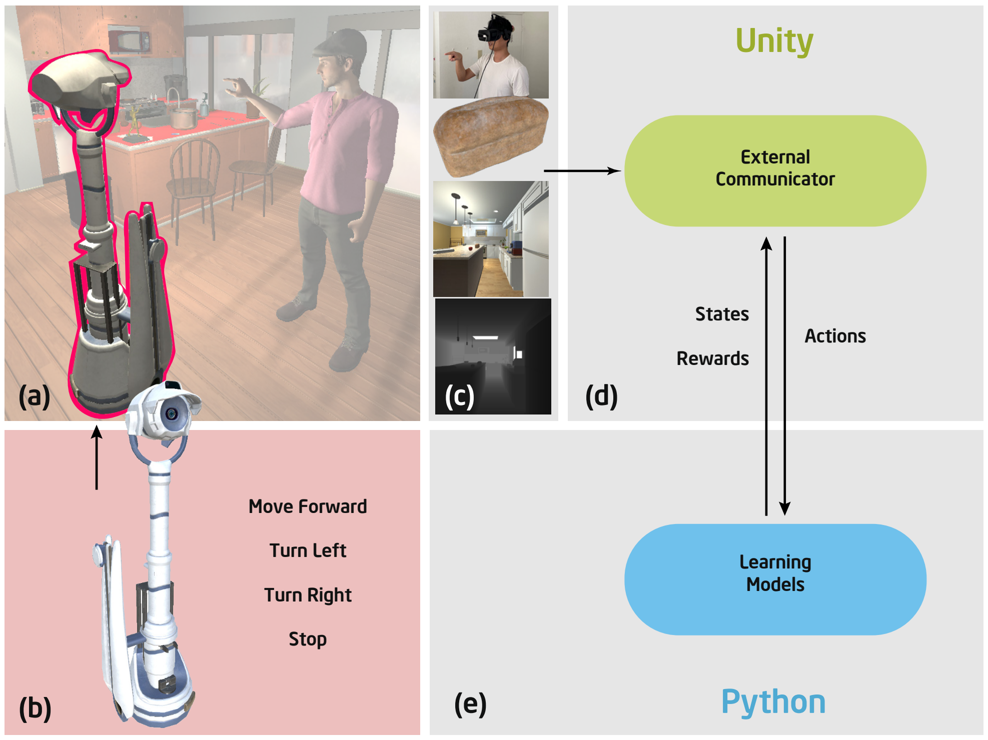

Figure 3.1: **Overview of GesTHOR.** (a) Scenes and the (b) learning agent are built in Unity. The agent can perform four actions: *move forward*, *turn left*, *turn right*, and *stop*. (c) It receives several sensory inputs, including RGBD images, target semantic labels, and human gestures. Unity contains (d) an external communicator that can communicate with the (e) RL learning model in PyTorch. The learning model receives states and rewards from the communicator and sends back chosen actions.

Other modalities such as semantic scene priors [148], Object Relation Graph (OGR) [179], and audio signals [175, 180] are used to equip the agent with human-level knowledge to help explore in seen and unseen environments. Notwithstanding the breakthrough in multimodal embodied AI, non-verbal communications are scarce. We claim that HRI with non-verbal cues is also important for embodied AI in both physical and simulation environments. In view of that, we propose a new framework, GesTHOR, for both simulations with embodied agent and interactions with learning objectives. We hope GesTHOR can help researchers in embodied AI in a new horizon.

3.2 Simulation Environment

3.2.1 Environment Overview

To build our simulation environment that supports multimodal HRI, we need several characteristics: (1) **Photo-realistic:** Sim-to-real (or sim2real) transfer is the bridge that connects embodied AI and robotics, and photorealism is a necessity for successful sim2real transfer [181, 182] by reducing the extra workload on domain adaptation. Besides, the development of computer graphics can render high-quality 3D assets that are built upon real-world objects, and game environments like Deep Mind Lab [110] and VizDoom [111] are no longer popular. (2) **Interactive:** This feature distinguishes our work from other frameworks. Despite the number of simulators for embodied AI, most of them focus only on visual richness and object physics. According to Table 2.1, there are many simulators that support object interactivity [40, 114–116, 118–120] so that either the user or the agent can manipulate the object. However, none of them pays attention to the interactivity between humans and agents. Our environment is human-robot-interactive. (3) **Customizable:** One of the problems in current embodied AI research is the reproducibility of the result. The simulator should be customizable and compatible with different methods.

Taking those features into account, we choose AI2-THOR [40] as the starting point to build our simulation environment. AI2-THOR is a simulator built on the Unity3D game engine. It contains a diverse collection of 3D objects crowdsourced or created from scratch. Those objects have photo-realistic textures and are actionable. From Table 2.1, AI2-THOR has the most diverse environment configuration. It has 120 room-sized 3D scenes from 4 categories (kitchen, living room, bedroom, and bathroom) and over 100 categories with thousands of unique object instances. It has been widely used for different visual navigation tasks [30, 139, 140, 148, 150]. Currently, AI2-THOR has three different environments for different tasks: 1) iTHOR [40] for object interaction; 2) RobotTHOR [173] for sim2real transfer; 3) ManiuplaTHOR [126] for atomic robotic manipulation. We name our simula-

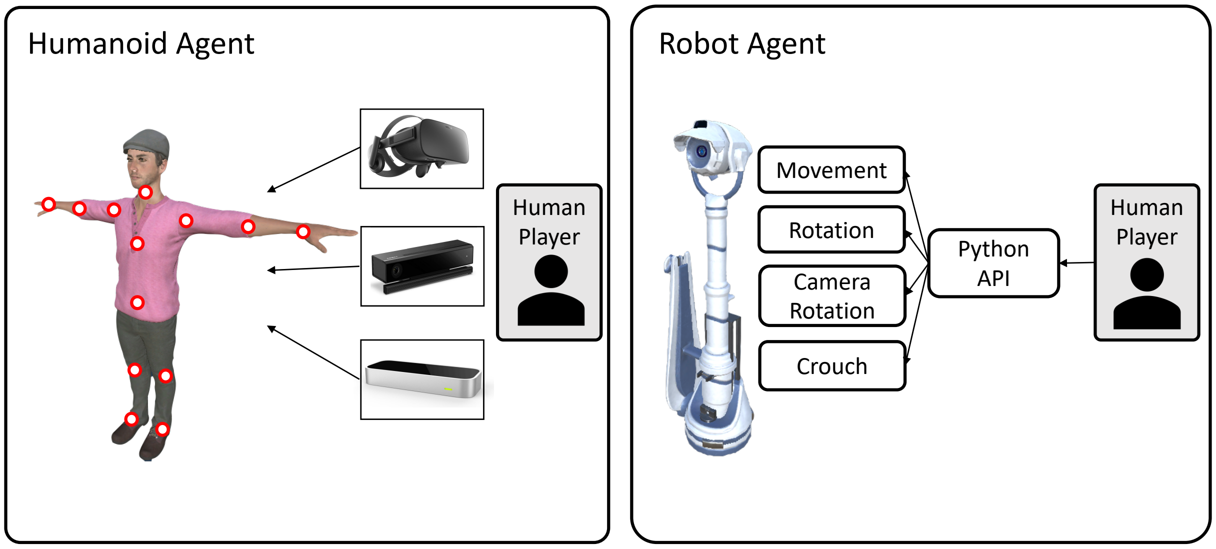

Figure 3.2: **The humanoid agent and robot agent.** The humanoid agent is manipulated via sensory devices and has the whole body movement. The robot agent is controlled by users via a Python API and has some basic robotic embodied motions.

tion environment GesTHOR to mean Gesture-based THOR and extend current AI2-THOR environments. Because of its popularity, we think making AI2-THOR as the basis of our simulaiton environment should help others using the same simulator adapt their work to our environment.

3.2.2 Agent Embodiment

There are two types of agents in our environment: humanoid agent and robot agent (see Fig. 3.2). Their details are explained below:

Humanoid agent. The humanoid agent has human appearances and embodied representations (see left of Fig. 3.2), similar to the avatars used in VRKitchen [116] and VirtualHome [114]. It is created using a humanoid avatar in Unity and can generate realistic human actions. Users can either control the agent by creating C# scripts in Unity or using Oculus Rift, Kinect v2 and Leap Motion Controller (LMC) jointly. The humanoid agent can therefore exhibit interactive gestures, such as gazing, body posture, and pointing, to communicate with the robot agent, and the user could experience this VR-based interaction by controlling the agent. This agent can be used to simulate a human-in-the-scene [41] learning environment to study HRI. We will elaborate on the details of the VR interface in Section 3.2.3.

Robot agent. The robot agent has the robotic look and actuation (see right of Fig. 3.2). The physical robot used is LoCoBot, the same as the one in [173]. The physical robot is equipped with an Intel RealSense RGB-D camera, and we are able to get the same physical properties for the robot agent. Unity's built-in camera component allows a 2D view of the virtual space. It is attached to the eyesight of the robot agent at 1.5m from the ground with a 90-degree field-of-view and provides real-time RGB images in the first-person view. The resolution of the RGB images is $3 \times 224 \times 224$, and each pixel contains scaled RGB values from 0 to 1. The depth image is extracted from the depth buffer of Unity's camera view. It has a size of $1 \times 224 \times 224$, and each pixel value is a floating-point between 0 and 1, representing the distance from the rendered viewpoint to the agent, linearly interpolated from 0m to 10m. The robot agent has a rigid capsule body with $0.3m$ radius and $1.5m$ height so that it can detect collisions with environmental objects. The robot agent can execute simple robotic actuation, including movements in 3D space, egocentric rotation, camera rotation (*i.e.*, pitch, yaw, roll), and crouch/stand. Users can control its behavior via a Python API as described in [173]. This robot agent is able to exhibit fine-grained actions with the robot arm as shown in [126]. Readers who are interested in the details can refer to that paper. The robot agent

3.2.3 VR Interaction Interface

3.2.3.1 Sensory Devices

VR is an essential technology for modern AI, where human handling is required in simulation. The non-contact interactivity makes VR widely used in areas like education, military, surgery, gaming, and so on. Extensive studies in VR have been conducted in HRI [106, 183]. For embodied AI, VR is also used as the human interface to interact with the objects [114, 116]. Unity3D game engine provides the ability to deploy across platforms and integrate third-party resources, compatible with different sensory devices [88, 184]. It allows us to build an interactive VR environment with gesture tracking functionality.

VR Headset: In our work, we use Oculus Rift (hereinafter referred to as Oculus) in conjunction with the Unity3D engine to bring users a realistic experience in simulation. The user can wear the Oculus HMD, and the two Oculus constellation sensors detect the rotations of the headset with user height offset. After that, sensor data can be processed and sent to Unity to update the head rotations of the humanoid avatar, thus changing the egocentric view about the scene. The updated scene view is reflected back on the HMD so that the user can be immersed in the simulation in real-time.

Gesture Tracking: After making a synchronous view connection between the user and the humanoid agent, the next question is how to control the humanoid agent motions. We follow the work of Li *et al.* [88] by incorporating two motion-sensitive devices, Microsoft Kinect v2 (hereinafter referred to as Kinect) and LMC, for gesture recognition and data collection. We didn't adopt the method used by [116], where they use a pair of Oculus controllers to track hand motions and calculate the body movements by Inverse Kinematics (IK) solver. Oculus controllers only have a coarse capture of the hand motions, and the IK solver is complex and prune to sensor noise. In contrast, Kinect and LMC can provide a more accurate and stable recognition of human gestures. Many gesture recognition-based studies have been conducted with the application of Kinect [185, 186] and LMC [187, 188]. Kinect works in mid and far ranges with body precision. Kinect is used to track overall body movements but is incapable of capturing fine in-hand motions. LMC offsets the drawbacks of Kinect providing fine-grained hand movements in short distances.

Device Arrangement: Fig. 3.3 illustrates the device arrangement. Users should stand away from Kinect by 0.5m-2m for the best tracking quality. LMC is mounted on Oculus HMD.

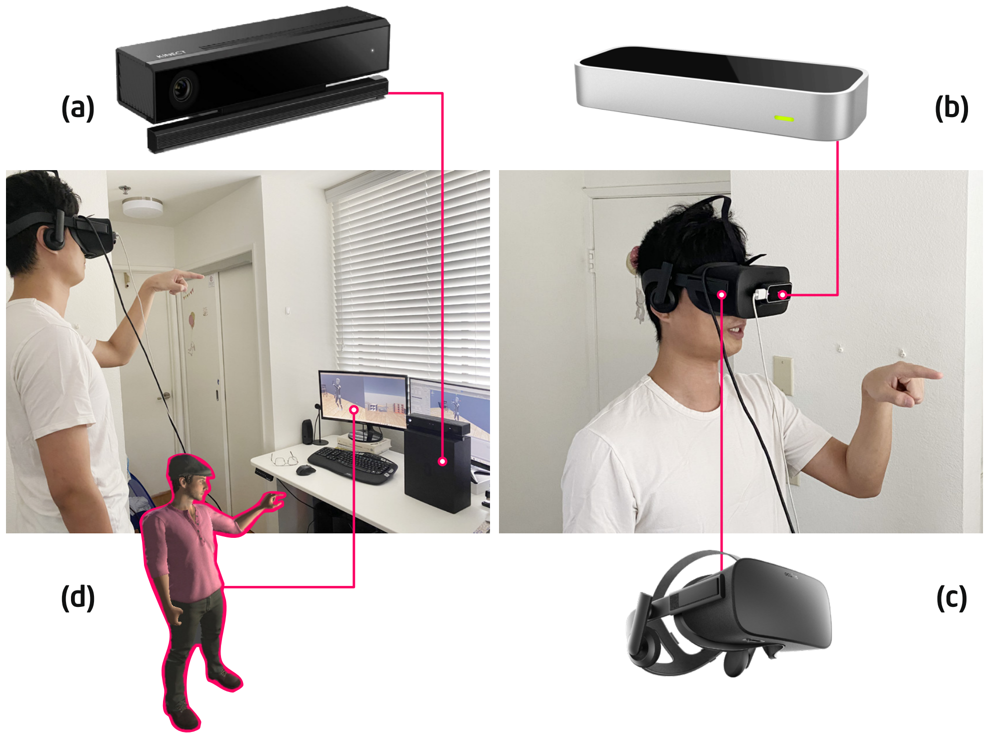

Figure 3.3: **Device arrangement for VR interaction in GesTHOR.** The human player wears an (c) Oculus headset with (b) LMC. (a) Kinect is placed 1.5m from the human player and 1.5m above the ground. The screen displays (d) A humanoid agent that are mirroring the body and hand movements.

During data acquisition, the human player is asked to wear the Oculus headset and face the Kinect sensor within eligible tracking distance. It is worth mentioning that in order to track the hand actions, especially finger actions, users need to move hands in front of LMC at a distance between 30cm and 60cm for the best tracking accuracy. In Unity, a humanoid agent (see Fig. 3.3d) mirrors players' movements in real-time, including body composure and hand motions.

3.2.3.2 User Interface

We provide a convenient toolkit for users to record and replay each program for their own datasets. A program involves four elements: *environment*, *human*, *robot*, and *instruction*. *Environment* encapsulates scene layouts and objects available. *Human* specifies the user who perceives the environment and interacts with the robot. *Robot* tells all the information

Attribute	Value
Scene	Kitchen_14
Target Object	Chair
Target Position	$(x, y, z) = (1.5, 1.1, 2.3)$
Human Position	$(x, y, z) = (2.4, 0.0, 4.6)$
Human Rotation	$90°$
Instruction	"Go to this chair"
Audio	audio.wav
Image	image.jpg
Motion	motion.csv

Table 3.1: **Metadata of an example program.** Each recorded program is connected with its own metadata. The metadata can be used to retrieved the program details or replay the program in Unity.

Body (tracked by Kinect)	Hips, Upper Leg, Lower Leg, Foot, Spine, Chest, Upper Chest, Neck, Head, Shoulder, Upper Arm, Lower Arm, Jaw
Hand (tracked by LMC)	Hand, Thumb Proximal, Thumb Intermediate, Thumb Distal, Index Proximal, Index Intermediate, Index Distal, Middle Proximal, Middle Intermediate, Middle Distal, Ring Proximal, Ring Intermediate, Ring Distal, Little Proximal, Little Intermediate, Little Distal

Table 3.2: **Tracked human body bones.** Human body bones tracked by Kinect and LMC.

of the robot agent in the same scene with the humanoid agent. Lastly, *instruction* should include what the human wants the robot to complete in terms of its observations, actions, and objectives. This interface should helper uses simulate real human-robot communication in multimodal ways.

Preparing the Scene: Since the user interface is intended to record embodied HRI programs, it is important to provide plausible scenes and objects. All 120 rooms and objects in iTHOR environment [40] can be used in our interface, and we allow users to choose their own definition of the program. However, we do have restrictions for some tasks. For example, for the object navigation task, target objects should be not be enclosed in a container. The agent requires to "see" the target to make successful navigation. One can use the user's discretion to design the rule for generating the programs.

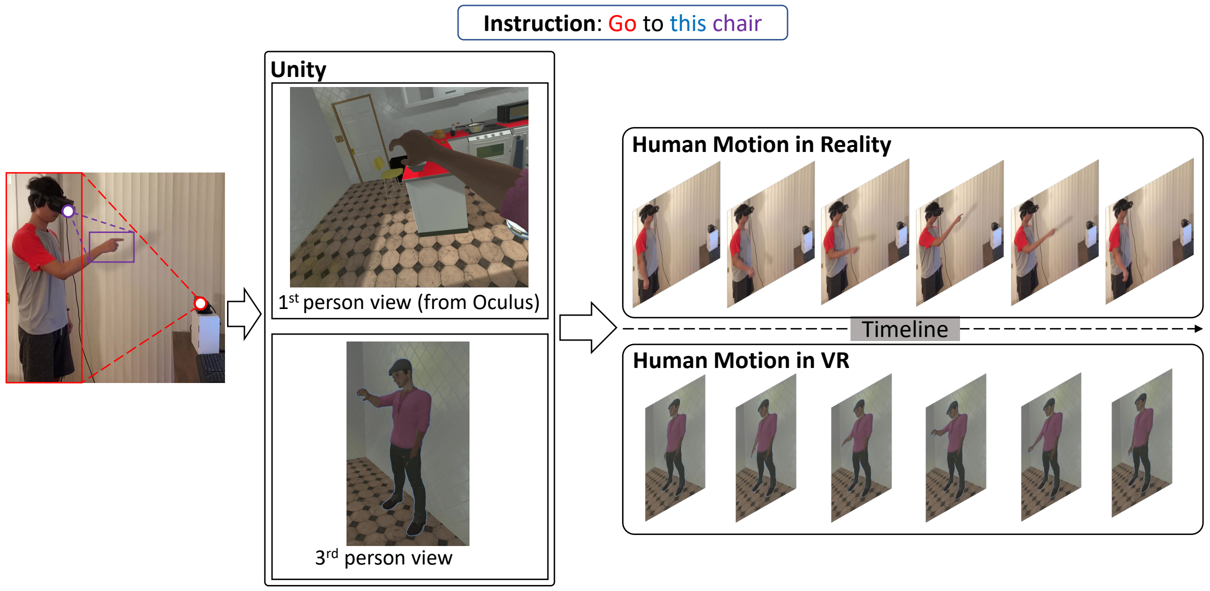

Figure 3.4: **Record human motions in a program..** This figure illustrates the process to record human motions in a program with metadata saved in Table 3.1. The user follows the instruction and gives a gesture to the robot agent (*e.g.*, pointing with a finger by the instruction“go to this chair”). Human motions are synchronized in reality and VR.

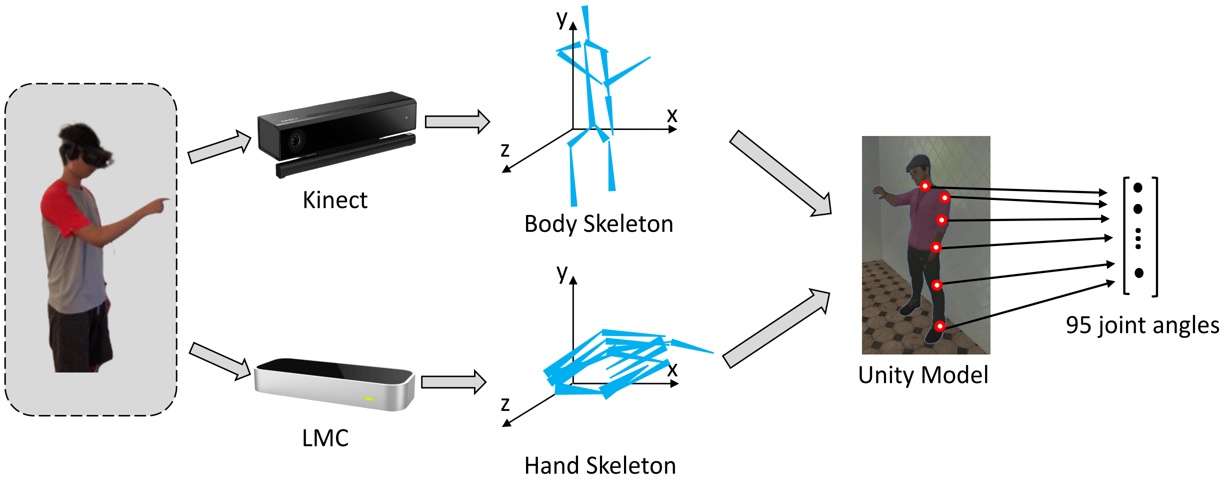

Figure 3.5: **Human motion data processing.** This is the workflow to process human motions. Kinect and LMC have the raw data for body and hand skeletons, which store the coordinates of tracked joints. They are fused by Unity SDKs and applied to the humanoid model. Unity keeps track of 95 joint angles as the final output features.

Record a Program: Following the device setup as shown in Fig. 3.3, the user can record a program with the sensory devices. The content of the program depends on the user’s intent. In lieu of the multimodal features in GesTHOR, we allow information from multiple channels to be recorded, including images, audios, and motions. These channels are saved as separate files, and their locations, as well as other environmental messages, are recorded in a metadata file. Table 3.1 is the metadata of an example problem. In this example, one needs to instruct the robot to navigate to a target chair. He might point with a finger with a short speech. He can choose to record the first-person view at each frame, the audio of the speech, and the gesture motion. In terms of recording of a gesture motion, Fig. 3.4 demonstrates the process. The user wearing a VR HMD is situated in the virtual environment and shows a gesture. Kinect and LMC track the body and hand motions and get the raw coordinates of tracked joints (see Fig. 3.5). Unity SDKs for Kinect and LMC handle raw data by fusing body and hand collections and converting it to a vector of joint angles that can be applied to the humanoid model in Unity. Unity humanoid avatar simplifies each connection point as a hinge joint since each body part only has limited Degree of Freedom (DoF). This humanoid character in Unity, therefore, is able to replicate the player's motions in real-time.

Tracked human body bones are listed in Table 3.2. Their names align with names defined by Unity *HumanBodyBone* enumeration. Notably, LMC tracks much more bones than Kinect because of its higher precision. Our gesture collection differs from other works [184] by including not only coarse body movement but also fine-grained finger motions. We argue that body postures and finger configuration are both paramount in understanding gestures in HRI, which has been testified for human-human communication [43]. Body conveys high-level messages, *e.g.*, in which direction the target is, while hand transmits detailed information, *e.g.*, where the target is in that direction.

Replay a Program: Our interface not only allows users to record and save programs but also provides tools to replay any program. As shown in Fig. 3.6. When a program is recorded, its metadata is dumped into a single file with all the information. The metadata is the key to reconstructing the recording. For example, we can know the scene, objects, and human/robot poses by referring to the metadata. We can even animate the humanoid agent to reproduce the recorded human motion by scripting in Unity. The replay feature for a program allows studies to emulate a human-in-the-scene case with the virtual world without

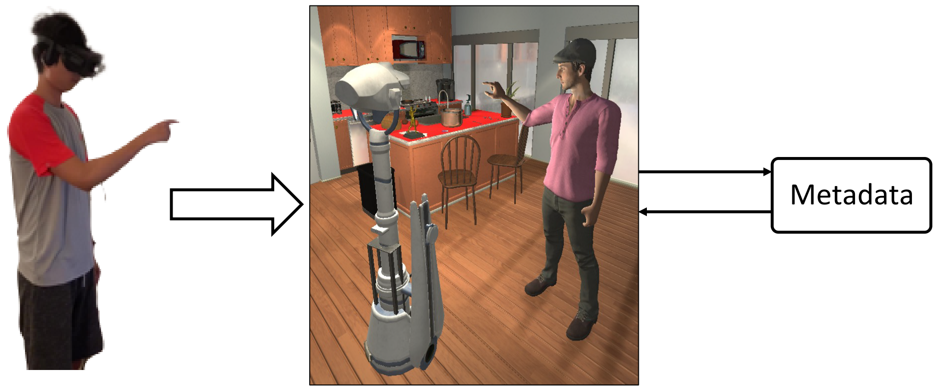

Figure 3.6: **Replay a recorded program.** Users can always replay the recorded program in Unity, including scenes, objects, and human motions.

building the physical environments.

3.3 Gesture ObjectNav Dataset

In the previous section, we introduce our simulation environment and the **VR!** (**VR!**) interactive interface. In this section, we talk about collecting data specifically for the object navigation task, which is the most straightforward yet challenging task for embodied navigation.

3.3.1 Implementation Details

Ideally, object navigation learning with gestures would take place in real-time, where a human player continuously observes an agent's behavior and interacts with it such that the agent can respond to the feedback immediately. Unfortunately, this "human-in-the-scene" idealization is infeasible because the entire training process may take hundreds of thousands of episodes. Therefore, we opt for using pre-recorded gestures to simulate real-time interactions between the human player and the agent as closely and efficiently as possible, which constitute our episodic Gesture ObjectNav Dataset—a dataset used for ObjectNav task with natural gestures.

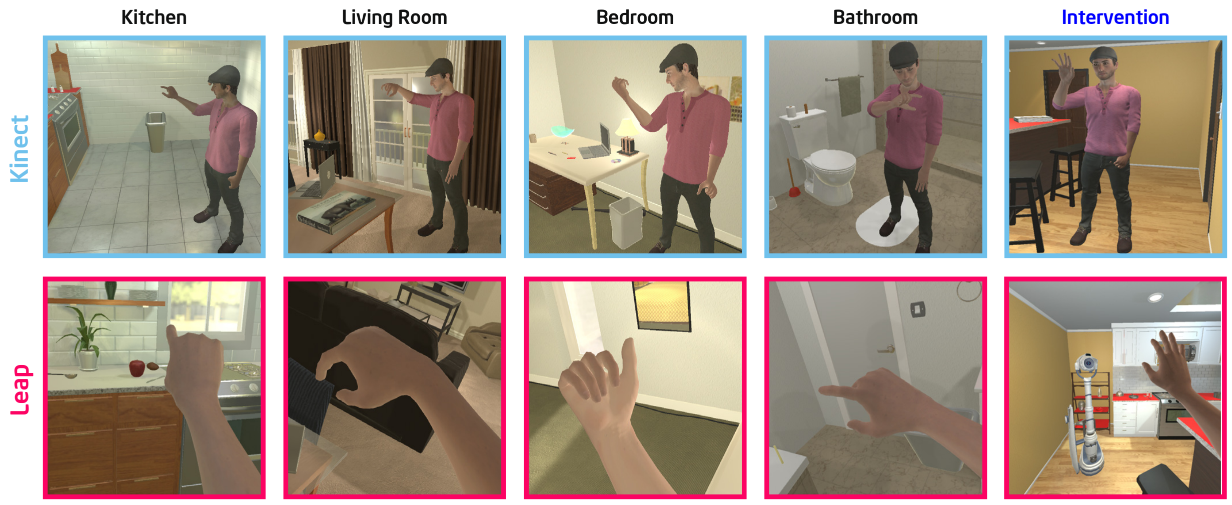

Figure 3.7: **Examples of referencing and intervention gestures.** The first four columns are referencing gestures, while the last one is an intervention gesture. Users perform different gesture styles while pointing at various target objects in the scene. The top row shows the body movements captured by Kinect, and the bottom row shows the hand configurations recorded by LMC.

For each episode, the users are provided with the environmental information, including room type and room ID, target object type, target position, humanoid agent position, humanoid agent rotation, robot agent start position, robot agent start rotation. There are two types of gestures that humans can use: one is for referencing, and the other one is for intervening. To record referencing gestures, a user is asked to communicate with the agent to guide the direction to the target with a gesture. The user is given enough time to familiarize the environment and come up with a natural gesture style. If ready, the user can show the referencing gesture to the robot agent. We do not ask participants to use any specific gestures such as pointing with a finger but encourage them to use any gestures as if they are talking to another person. To record intervention gestures, the user is given a situation when the robot agent needs a warning for his current movements. Therefore, the user can think of a gesture in a rejective manner and show this gesture while facing the robot agent (*i.e.*, waving hands or crossing forearms). Note that intervention gestures are not recorded episodically. They are saved as extra components in the dataset since they don't rely on environmental information. See Fig. 3.7 for examples of referencing the intervention gestures.

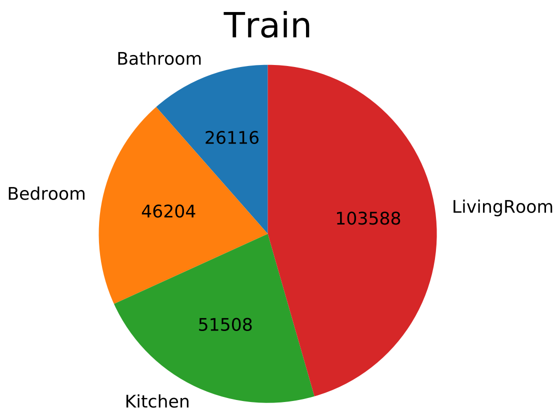

Figure 3.8: **Distribution of scene types in the train fold.** This figure shows the number of programs for different scene types for training.

Kinect and LMC recognize the gesture at 30 frames per second, and the real-time animations are displayed with the humanoid agent. At each frame, the transforms of all detected body and hand joints are captured and converted to joint angles of the humanoid agent in Unity as depicted in Fig. 3.5. Each user is given 100 frames to complete one gesture motion. The gesture sequence, as well as the environmental information, is recorded as one episode in the dataset. Each gesture sequence has exactly 100 time steps, although different sequences have different idle leading or trailing times. We add this specification for easier processing of the gesture features during model updates.

3.3.2 Dataset Statistics

AI2-THOR provides 120 scenes covering four different room types: kitchen, living room, bedroom, and bathroom. We randomly split 30 scenes for each scene type into 20 training rooms, five validation rooms, and five testing rooms. We have over 230,000 unique programs for training (see Fig. 3.8) and 5,000 programs for validation and testing (250 programs for each room). Every episode has a distinct referencing gesture and task definition. We also have 20 different intervention gestures that can be used during the agent learning process. Each program is defined by $(s, o, p_o, p_h, r_h, p_r, r_r, m_r)$, where

- $s =$ the scene in GesTHOR,
- $o =$ target object type,
- $p_o =$ target position,
- $p_h =$ human position,
- $r_h =$ human rotation,
- $p_r =$ robot position,
- $r_r =$ robot rotation,
- $m_r =$ motion of referencing gesture.

One important aspect in GOND is that all gestures are natural. Naturalism is reflected in several parts in the dataset: 1) Diversity. As shown in Fig. 3.7. Although the referencing gestures need to provide directional cues, we can use different fingers or even the whole palm to give the referential message. 2) Continuity. Each gesture should be a continuous motion. A user usually starts from a natural standing pose, then proceeds to a leading pose, and ends with a trailing pose. We claim that instead of treating gestures as static skeletons [93], leading and trailing poses are also significant for a holistic understanding of gestures. The learning agent should be able to uncover the semantics from each motion. 3) Imperfection. Although referencing gestures are directional cues, it is impossible to know the exact position of the target directly. We allow noises in gesture motions and think this is part of the agent learning process.

Table 3.3 shows all the object categories and Fig. 3.9 is the distribution of GOND w.r.t. object types. We choose 38 object types from AI2-THOR from different scene types. Note that some objects come from a specific scene type (*e.g.*, ToiletPaper can only appear in Bathroom), and some objects can be found in multiple layouts (see Fig. 3.10(b) where chairs appear in the kitchen, living room, and bedroom). Multiple object instances with the same categories can also exist in the scene (see Fig. 3.10(a) where seven chair instances show in the same living room), posing challenges for the object navigation task. We show the distribution of object instances with the target category in Fig. 3.11. Of note, there are

Scene Type	Object Categories
Kitchen	Apple, Break, ButterKnife, Chair, Cup, DiningTable, Fork, Knife, PaperTowelRoll, Plate, Potato, RemoteControl, SideTable, Spoon, Tomato, Window
Living Room	ArmChair, Box, Chair, CoffeeTable, DeskLamp, DiningTable, FloorLamp, Laptop, Newspaper, Plate, SideTable, Sofa, Television, TissueBox, Window
Bedroom	AlarmClock, Box, Chair, DeskLamp, Laptop, Poster, RemoteControl, SideTable, Sofa, Television, TissueBox, Window
Bathroom	HandTowel, HandTowelHolder, PaperTowelRoll, Plunger, ShowerCurtain, ShowerDoor, SideTable, TissueBox, ToiletPaper, ToiletPaperHanger, Towel, TowelHolder, Window

Table 3.3: **Object categories in GOND for different scene types.** The object categories are selected from AI2-THOR as our navigation targets. Some objects are unique for a scene type, while others can appear in different scene types.

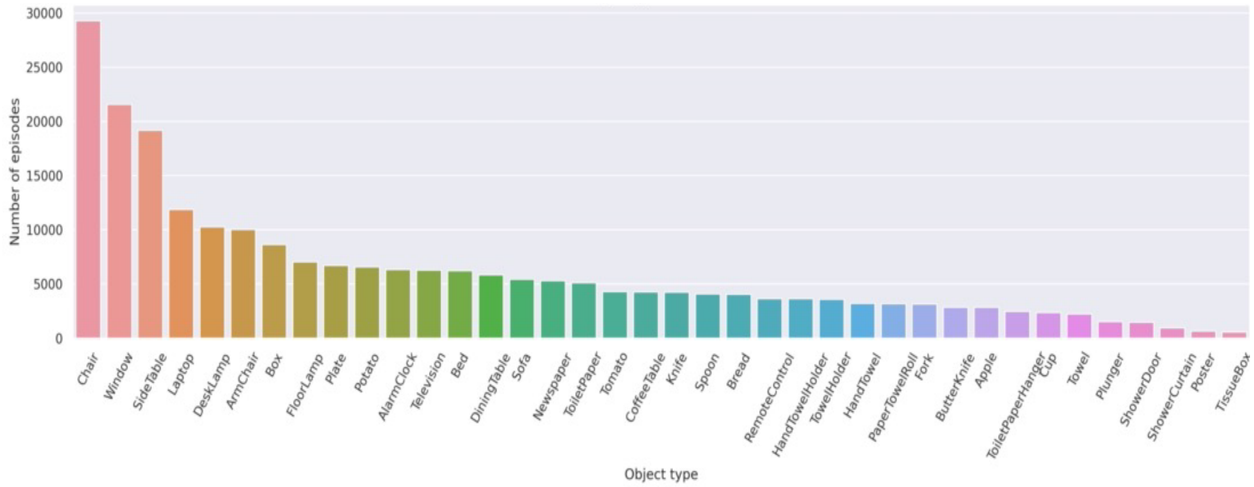

Figure 3.9: **Distribution of object types in the train fold.** This chart shows the number of programs for different object types for training.

around 30% programs in train, validation, and test sets that have multiple instances, which is not a trivial object navigation task. Besides, we make sure that all the objects are visible in the scene, not occluded by objects from the humanoid agent's field of view and reachable from the starting location of the robot agent. From Table 3.3, we know that furniture like chairs and tables appear more frequently in the dataset because they exist in different scene types and are easier to be recognized. This should align with human common sense since we also interact with those objects more often in our daily communication.

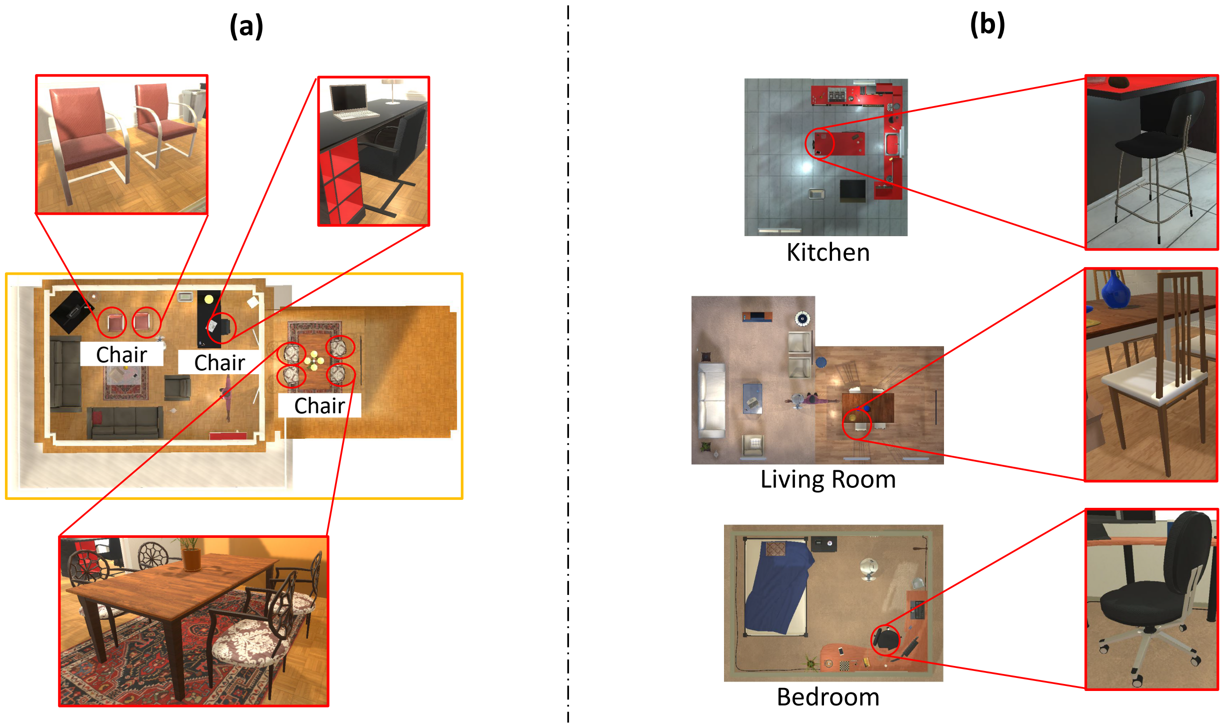

Figure 3.10: **Object diversity in our gesture dataset.** We show the diversity of objects in GOND from two aspects: (a) Multiple instances of the same categories in the same scene. (b) The same object category in different scene types.

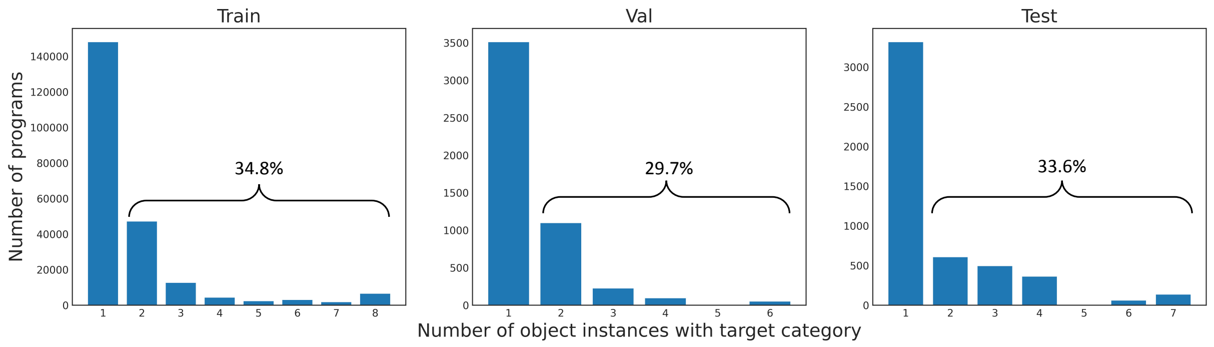

Figure 3.11: **Number of instances with the target category.** We show the number of programs for the different number of object instances with the target category in train, validation, and test folds. The percentage in each fold with multiple object instances is also displayed on the plot.

Fig. 3.12 demonstrates the spatial diversity in GOND. Fig. 3.12(a) shows the geodesic distance distribution between the robot's initial position and the target position. Geodesic distance measures the shortest-path distance in the environment [135]. This metric takes the structural information into account and depicts the reachability to the target from the robot's perspective. We see that the geodesic distances distributes in different ranges with an average

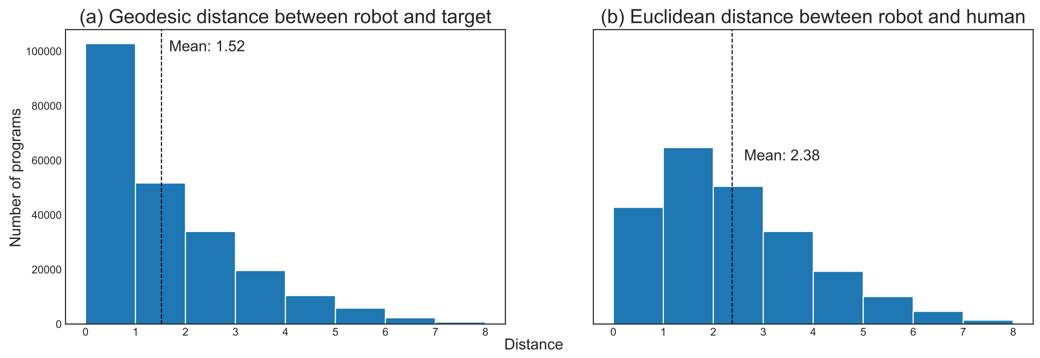

Figure 3.12: **Spatial relationship in the scene.** To illustrate the diverse spatial distribution in our dataset, we show the histogram of (a) geodesic distance between robot and target and (b) Euclidean distance between robot and human.

of 1.52m (1 unity in Unity represents 1.5). Fig. 3.12(b) shows the Euclidean distance between the robot's initial position and the human position. This tells us the distance between human and robot when human shows the gestures. Human intends to behave differently when the proximity to the robot changes, and our dataset captures a diverse collection of the distances to ensure the variety of gestures.

CHAPTER 4

Embodied Navigation with Natural Gestures

In this chapter, we will talk about the task we work on with the dataset introduced in Section 3.3—an ObjectNav task with natural gesture instructions. After brief introduction on the choice of our task in Section 4.1, we will start with the problem formulation (Section 4.2) followed by the RL learning network (Section 4.3). Finally, we present our experimental results both quantitatively and qualitatively (Section 4.4).

4.1 Introduction

According to our literature review in Section 2.3.2, there is a diverse collection of tasks for embodied intelligent agent (see Fig. 2.3 for some examples). They utilize various learning approaches and provide different benchmarks to evaluate the agent performance. By interacting with the virtual world egocentrically and integrating numerous modalities like image, depth and GPS, the agent acquires different skillsets like exploring [130–133], object-finding [12, 21, 28, 136–140], and question-answering [13, 14, 157, 178, 189].

Notwithstanding the versatility and diversity of embodied AI tasks, there is a lack in the exploration of non-verbal interactions. From a human perspective, our natural way of communication is interwound with verbal and body language. For example, in Fig. 1.1, we prefer to express our intent with a short sentence and a natural pointing gesture. From an AI perspective, more communicative channels can help its generalization to unseen environments. For example, when there are multiple instances of the target object type in the scene, it is cumbersome to describe the context verbally and ambiguous for the agent to understand. What's more, when adapted to a new environment, it is hard for the agent to process new contextual information. To this end, we plan to conduct a thorough study with the object navigation task and use natural gestures from GOND. We contemplate on prior arts and quest for the following questions: Instead of using natural language, can we replace the language grounding by gestures in a similar setting? Can we improve the performance of navigation with gestures incorporated? Can the learning agent acquire the underlying semantics of gestures, even when they are not predefined?

4.2 Problem Formulation

Our task can be grouped as an ObjectNav task [142]. In each training or evaluation episode, we randomly select one program from the dataset. The robot agent and humanoid agent are initiated at specified locations. Other environmental layouts should also be included in each program of GOND. Note that there can be more than one instance of the target object type in the same environment (see Fig. 3.11). Different from the criteria set up in [135] where the agent is encouraged to navigate to the object instance that is closest to its starting position, we have a specified target if there are objects with the same type in the scene. To complete the task, the agent must navigate to the specified target instance. The embodied robot agent has four available actions: *turn left*, *turn right*, *move forward*, and *stop*. Each turning action results in a rotation of 15°, and each forward action results in a forward displacement of 0.25m. An episode is deemed successful iff the following conditions are satisfied [142]:

- **Proximity**: The robot agent, which could be abstracted as a point mass, must stay with a 2D Euclidean distance less than 1m from the target. For computation efficiency, we also abstract the target as a point mass in its center. This value is chosen empirically to ensure the navigation task is not trivial to the robot agent.
- **Visibility**: The object must also be within the agent's field of view in order to succeed.

We consider the robot agent has a field of view of 90°. We enforce lenient criteria that at least 1 pixel of the target should be within the observation frame.

- **Awareness**: It is important to know that the agent "understands" the human intents. Therefore, the agent then needs to issue a termination (*i.e.*, *stop*) action in the proximity of the goal. We allow the agent to issue multiple stops in an episode but measure success rates using different numbers of maximum stops (1-3). We allow an unlimited number of stops in training; the agent needs to explore and learn after issuing incorrect stops in earlier episodes.

An episode is terminated if the above success criteria are met or the maximum allowed time step (which is 100 in our setup) is reached. The limitation on the number of steps encourages the agent to complete the task efficiently and also leaves a margin for the agent to explore at the start of exploration, considering that the average number of steps required to complete an episode is around 20.

4.3 RL Learning Architecture

4.3.1 Learning Objective

We formulate our visual gesture navigation using DRL, specifically PPO. Our learning process can be viewed as a Markov Decision Process (MDP). At each time step t, the agent perceives a state s_t (*i.e.*, a combination of the sensory inputs), receives a reward r_t from the environment, and chooses an action a_t according to the current policy π:

$$a_t \sim \pi_\theta(a_t|s_t), \tag{4.1}$$

where θ represents parameters for the function approximator of the policy π. PPO is an actor-critic model. It uses the following objective to update the policy:

$$L^{CLIP}(\theta) = \mathbb{E}^{\pi}\left[\sum_{t=1}^{T} min(r_t A_t^{\pi}, clip(r_t, 1-\epsilon, 1+\epsilon)A_t^{\pi})\right], \tag{4.2}$$

where $r_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$. r_t measures the difference between the new policy and old policy. ϵ is a hyperparameter that ensures that new policy will not deviate too much from the old policy, leading to a rather stable policy update.

In our work, the advantage value in Eq. (4.2) is represented by the General Advantage Estimation (GAE) [190]. Advantage can be estimated using either TD as shown in Eq. (2.8), or Monte Carlo runs. However, the first one has a high bias, and the second one has a high variance in value estimation. One solution as described in [190] is annealing between those two by taking the exponential average of them. The advantage function is:

$$A_t^{GAE(\gamma,\lambda)} := \sum_{l=0}^{\infty}(\gamma\lambda)^l \delta_{t+1}^V. \tag{4.3}$$

Here, γ is the exponential weight discount, which trades off the bias variance. λ is the discount factor for the future rewards. δ_{t+1}^V is the TD advantage estimate:

$$\delta_{t+1}^V = r_t + V(s_{t+1}|w) - V(s_t|w), \tag{4.4}$$

where w is the parameters for the value function approximator. To get A_t^{GAE}, we have the following relationship given a trajectory:

$$A_{t+1}^{GAE} = \delta_t^V + (\lambda\gamma)A_t^{GAE}. \tag{4.5}$$

Since our objective to maximize Eq. (4.2), it is identical to perform Stochastic Gradient Descent (SGD) on the objective function to update the actor and critic network (*i.e.*, θ and w). See Algorithm 1 for the entire algorithm.

Algorithm 1 PPO learning for the navigation agent

```
1: Parameters: $\epsilon$, $\gamma$, $\delta$.
2: Initialize policy network parameter $\theta$ and value network parameter $w$.
3: for $k = 1, 2, \ldots$ do
4:     for $t = 1, 2, \ldots$ do
5:         Collect a rollout $\tau_k$ on current policy $\pi(\theta)$ until N time steps are reached.
6:         Compute $\delta_t^V$ with Eq. (4.4).
7:         Compute $A_t^{GAE}$ with Eq. (4.5).
8:         Do mini-batch SGD on Eq. (4.2) to udpate $\theta$ and $w$.
9:     end for
10: end for
```

Hyperparameter	Value
Rollout length	128
Update repeats	4
Learning rate (r)	0.0003
Clipping parameter (ϵ)	0.2
GAE [190] parameter λ	0.99
Discount factor (γ)	0.99

Table 4.1: **Hyperparameters for training PPO agent.** All hyperparameters are fine-tuned empirically.

4.3.2 PPO Implementation

Our implementation is built on AllenAct [191], which is a flexible learning framework designed uniquely for embodied AI research. It is supportive of different simulators and research tasks. In our work, we make some modifications to the ObjectNav task in the AI2-THOR environment from AllenAct. The learning network is implemented in PyTorch [192], and the model is trained with four NVIDIA GeForce GTX 2080Ti GPUs. Each model is trained with 20 threads at 200 frames per second (*i.e.*, 20 Unity instances are invoked, and the frame rate takes all environments into account). The number of threads is selected empirically, with a trade-off between the single instance frame rate and the total frame rate.

We implement our PPO algorithm with 128 steps in a single rollout and repeat the mini-batch gradient update four times. The buffer size is 1280 to save rollouts for the policy upate. We use Adam [193] as the optimizer for SGD with a learning rate of 0.0003 that linearly decays to zero, and the maximum gradient norm is 0.5. We choose a discount factor of 0.99, a GAE coefficient of 0.99, and a clipping ratio of 0.2. The detailed hyperparameters for training PPO agent is listed in Table 4.1.

In terms of rewards, the agent receives a positive reward of $+1$ if it completes the navigation successfully. Since we encourage the agent to reach the target object with a minimal amount of steps, the agent receives a small time penalty of -0.001 for each step. We further add a collision penalty of -0.005 for each collision detected; the collision penalty is added to mitigate "sliding" behavior stated in [12]. If the agent stops in the wrong location, a penalty of -0.01 is added. This reward scheme should provide incentives to the navigation agent so that it prefers exploration to exploitation at the beginning of the training (*i.e.*, maximize success rate), and allow the agent to gradually learn the optimal navigation policy (*i.e.*, maximize efficiency). Although the reward is sparse, we allow multiple *stop* actions to make a smooth learning curve.

4.3.3 Learning Network

We equip the embodied agent with different sensory modalities, and each of them feeds into a part of the input network for the RL model. All layers in our network are followed by a Rectified Linear Unity (ReLU) non-linearity function. Below we introduce these components of the architecture.

Visual network: The backbone of the visual network is ResNet-18 [194] pre-trained on ImageNet. It takes $224 \times 224 \times 3$ RGB and $224 \times 224 \times 1$ depth images as inputs. The weights of all layers in the network except the last fully connected layer are frozen during training to extract $512 \times 7 \times 7$ features from the *conv*5 layer. These features are fed to 2 CNN layers with 1×1 convolutions to reduce the number of channels from 512 to 64. The output features are flattened and a fully-connected layer produces 512-dimensional visual feature v_t^{rgb} and v_t^{depth} for RGB and depth perceptions respectively.

Gesture network: The raw input of the gesture network is a sequence with 100 time steps and 95 features. Each feature represents the joint angles from the Unity humanoid model, which are processed from the coordinates for tracked body and hand joints by Kinect and LMC (see Fig. 3.5). The gesture input is flattened and encoded into a 512-dimensional gesture feature g_t.

Semantic network: In addition, we provide the target object category from a selected set and pass it to an embedding layer of dimension 512 denoted as semantic feature m_t. This is equivalent to speech or text instructions from a user in prior work on interactive embodied agent learning. Since our focus in this paper is the gesture, we simplify this part of the input as a categorical variable (*e.g.*, a single word in a fixed vocabulary). Note that this vector alone does not specify the target object location: There can be multiple instances of the same object category in the scene, and the agent needs to infer which instance the human player is referring to.

Actor and critic networks: Both the actor and critic networks are 2-layer MultiLayer Perceptron (MLP). Each layer has 512 features. Their parameters are aforementioned θ and w. The critic network outputs a single value as the estimation of the value function $V(s_t|w)$. The actor-network outputs four logits as the log probabilities for the robot agent actions. A logistic regression function is applied to get the probability distribution, and the robot selects an action according to this distribution.

Note that v_t^{rgb}, v_t^{depth}, g_t, and m_t are concatenated on the last dimension to get a 2048-dimensional feature vector. There is a memory unit using Gated Recurrent Unit (GRU) [195] after this combiner to encode and memorize the history of navigation. The GRU has two layers with a hidden dimension size of 512. The 512-d feature produced by the GRU is encoded by the actor and critic network separately to get the value and action. The overview of our RL learning model is depicted in Fig. 4.1.

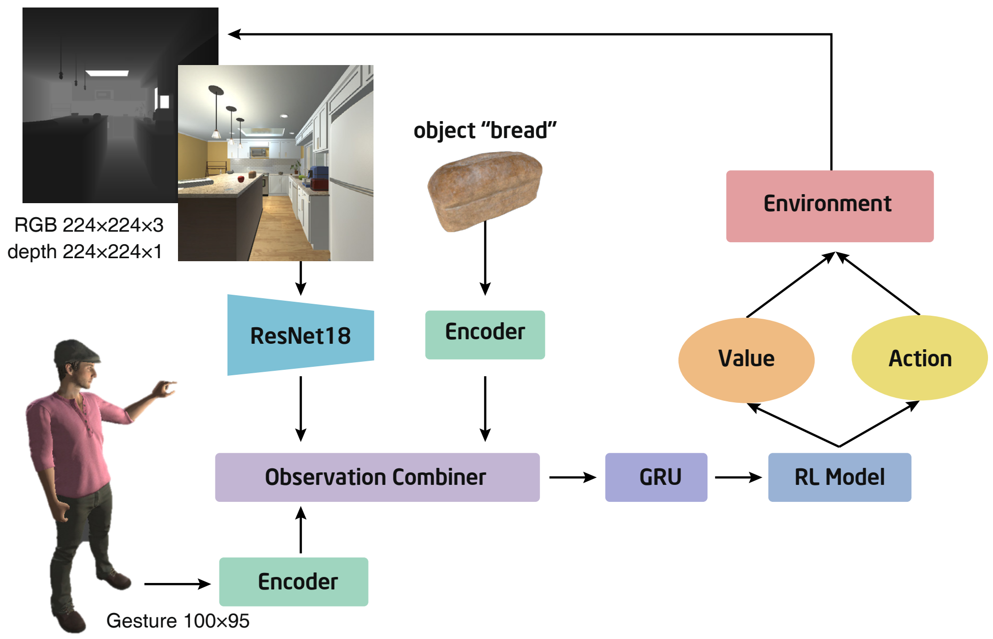

Figure 4.1: **The RL learning model.** Our model fuses perceptions from different sensory modalities, and the actor-critic model samples an action in each step according to the updated policy and send it back to the environment.

4.4 Experiments

We set up benchmarks to define object navigation tasks in GesTHOR with natural human gestures and use the RL learning architecture introduced in Section 4.3 to evaluate the agent navigation performance.

4.4.1 Evaluation Criteria

We evaluate our methods in GesTHOR environment with data points from GOND. AI2-THOR provides 120 scenes covering four different room types: kitchen, living room, bedroom, and bathroom. Each room has its own unique appearance and arrangements. We randomly split 30 scenes for each scene type into 20 training rooms, five validation rooms, and five testing rooms. We train each model for 5 million steps and evaluate them on each scene in validation and test sets for 250 episodes and report the average results with three runs. For evaluation on the test fold, we choose models with the best performance on the validation fold.

We use 2 metrics to evaluate different methods:

- Success Rate (SR): for the i-th episode, the success can be marked by a binary indicator S_i. The success rate is the ratio of successful episodes over completed episodes N:

$$SR = \frac{1}{N}\sum_{i=1}^{N} S_i. \tag{4.6}$$

- Success weighted by Path Length (SPL): this metric is proposed by Anderson *et al.* [135]. It measures the efficacy of navigation. SPL is calculated as follows:

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \left(\frac{l_i}{\max(p_i, l_i)}\right), \tag{4.7}$$

  where l_i is the shortest path distance from the agent's starting position to the goal in episode i, and p_i is the actual path length taken by the agent.
- Distance to Success (DTS): this metric has been used for recently embodied navigation work [175]. It has the following formulation:

$$DTS = \max(0, |x_a - x_t| - d), \tag{4.8}$$

  where x_a is the agent location on evaluation and x_t is the target location. d is the navigation threshold (1.5m). Different from SPL, it evaluates the agent performance regardless of success. It measures how close the agent is to the target at the end of the episode. DTS works together with SPL to measure the efficiency of navigation.

It is worth noting that we allow multiple *stop* actions during training, so we evaluate SR, SPL, and DTS under the different number of allowed stops as well. However, due to the nature of object navigation, the agent performance at the first *stop* action is the most important to us.

We measure the performance of the following baseline models:

- **Random:** in addition to (1), the agent receives referencing gesture inputs.
- **Vision Only :** The agent only has the visual observations (*i.e.*, RGB, and depth images) at each frame. It is target-agnostic and therefore sets another baseline with only *visual exploration* capability.
- **Visual Semantic (VisSem):** The agent uses both the visual networks and the semantic network. The target category is provided as a semantic cue for the navigation agent to identify the target.

We compare the baseline models with our own methods:

- **VisSem+Referencing:** In addition to the VisSem model, the agent receives referencing gesture inputs. Referencing gesture is provided as a sequential semantic cue in the entire navigation episode. We evaluate if referencing gestures can replace natural language instructions to improve navigation performance.
- **VisSem+Intervention :** In addition to the VisSem, the agent receives rejective gesture inputs when the forward direction forms an angle larger than 90 degrees between the agent and the target. This also gives sequential semantic information to the agent.
- **VisSem+Both:** The agent uses both the referencing gestures and intervention gestures. Intervention gesture is fed to the gesture network only if the agent moves incorrectly according to the VisSem+Intervention model, and the referencing gesture is provided to the agent for the rest of the time.

In our comparative setting, VisSem model does not use any gestures. While one may expect that it should always underperform, this is only true if the agent has learned and inferred the semantics of human gestures and incorporated the signals during navigation, which is the focus of our evaluation. Again, this is not trivial because we do not predefine the meaning of any gestures. Similarly, the intervention gestures are strong directive feedback from the human user, but we evaluate how well the agent can infer its meaning and adopt it in navigation.

Methods	SR (%)	SPL (%)	DTS (m)
Random	8.7	7.1	2.908
Vision Only	0.36	0.098	2.719
VisSem	15.2	9.1	2.419
+Referencing	19.2	11.5	2.313
+Intervention	41.0	24.8	2.004
+Both	**49.2**	**28.7**	**1.976**
Scene Prior [148]	35.1	15.5	-
SAVN [140]	40.9	16.8	-

Table 4.2: **Navigation performance for baseline and our methods.** All the metrics are the results of the evaluations on the test set at the first stop. We note that referencing and intervention gestures both boost the performance, of which intervention gesture is more effective. We also compare our result with state-of-the-art works reported from [148] and [140].

4.4.2 Navigation Performance

Scene Types	Methods	SR (%)			SPL (%)		
		Train	Validation	Test	Train	Validation	Test
Kitchen	VisSem	12.3	10.2	11.5	7.6	7.1	7.9
	+Referencing	21.1	18.7	19.2	13.3	11.6	11.2
	+Intervention	**44.9**	**31.5**	**40.3**	**27.0**	**20.0**	**24.0**
Living Room	VisSem	6.3	3.6	3.5	4.1	2.3	2.1
	+Referencing	4.9	3.2	2.7	3.2	1.7	1.6
	+Intervention	**13.0**	**9.0**	**9.5**	**7.8**	**5.3**	**5.4**
Bedroom	VisSem	15.2	9.1	8.7	9.1	5.3	5.4
	+Referencing	**43.5**	10.7	15.4	**28.3**	6.6	10.5
	+Intervention	42.4	**22.4**	**20.4**	27.5	**13.8**	**11.9**
Bathroom	VisSem	16.3	15.5	11.9	11.4	9.1	8.8
	+Referencing	33.0	19.4	19.9	20.5	11.1	11.7
	+Intervention	**40.5**	**32.2**	**35.0**	**29.1**	**21.0**	**23.0**

Table 4.3: **Evaluation results for train/validation/test split with different scene types.** SR and SPL at the first stop are reported in this table. We compare models trained with referencing gestures and intervention gestures against a baseline model.
* Reported from [148]. This method uses additional scene prior knowledge but not gestures.

Table 4.2 evaluates all the baseline models and our methods on the test set. Note that all the scenes in testing are "new" to the navigation agent who is trained on a completely different set. The generalization to unseen environments is the key ability for learning a good model. Table 4.2 also puts a hard constraint on the number of stops to one to match the state-of-art benchmarks [135]. We can see that Vision Only model performs poorly given that

Scene Types	Methods	SR (%)				SPL (%)			
		1 Stop	2 Stop	3 Stop	∞	1 Stop	2 Stop	3 Stop	∞
Kitchen	VisSem	11.5	18.0	23.1	49.3	7.9	12.1	14.6	25.1
	+Referencing	19.2	26.3	29.7	47.9	11.2	15.0	16.6	24.1
	+Intervention	**40.3**	**55.1**	**62.8**	**89.0**	**24.0**	**32.7**	**37.2**	**52.8**
Living Room	VisSem	3.5	6.4	9.5	23.3	2.1	3.7	5.1	9.8
	+Referencing	2.7	3.9	4.7	7.8	1.6	2.4	2.9	4.7
	+Intervention	**9.5**	**16.9**	**21.7**	**58.0**	**5.4**	**9.8**	**12.6**	**30.8**
Bedroom	VisSem	8.7	15.3	18.4	31.0	5.4	9.5	11.2	17.0
	+Referencing	15.4	18.7	20.2	31.9	10.5	12.4	13.3	19.2
	+Intervention	**20.4**	**27.9**	**33.5**	**51.2**	**11.9**	**16.3**	**19.5**	**29.7**
Bathroom	VisSem	11.9	18.9	23.7	57.8	8.8	13.8	16.9	33.1
	+Referencing	19.9	30.5	35.9	64.3	11.7	18.1	21.3	32.4
	+Intervention	**35.0**	**44.6**	**51.7**	**76.5**	**23.0**	**29.6**	**33.9**	**48.3**

Table 4.4: **Evaluation results for test scenes on four scene types with different number of allowed stops (∞ denotes infinite allowed stops).** SR and SPL are presented. We compare models trained with referencing gestures and intervention gestures against a baseline model.

no semantics are not provided, which aligns with our expectations. Compared to the VisSem baseline model, we confirm that adding gestures can significantly improve the navigation success rate as well as the efficiency over the baseline model. Of note, models trained with intervention gestures outperform models trained with referencing gestures, both in SR and SPL, demonstrating that intervention gesture is a more effective kind of gesture to communicate with the agent. It can be explained by the fact that the intervention gesture instructs the agent in an *online* manner, which is a stronger signal compared with the referencing gesture, which is presented in an *offline* way at the start of each episode.

Our model achieves the best performance when combining both the referencing and intervention gestures. It indicates that similar to human communication, HRI in embodied AI also benefits from more diversity in communication channels. We also compare our work with some state-of-the-art methods reported in [148] and [140]. Yang *et al.* [148] uses oracle scene prior knowledge graph to assist navigation, while Wortsman*et al.* [140] learns to adaptively update the model during both training and inference stages. Our best model outperforms both methods.

Table 4.3 and Table 4.4 measure the navigation performance of models trained separately for each scene type. They give a more thorough review of our model performance adapted

Methods	SR (%)			SPL (%)		
	1	2	3+	1	2	3+
VisSem	20.2	18.9	8.0	6.1	7.0	4.3
+Referencing	21.1	19.8	13.5	7.9	8.3	6.4
+Intervention	49.2	41.1	32.2	17.7	16.7	11.1
+Both	**55.2**	**47.5**	**36.5**	**19.4**	**20.3**	**11.4**

Table 4.5: **Navigation performance on multi-instance scenes.** All the metrics are the evaluations results on the test set at the first stop. All methods are evaluated with the different number of target category instances in the scene.

specifically to different scene types. Table 4.3 shows the performance of different methods when evaluated at the first stop. The result is similar to what we show in Table 4.2, further asserting the effectiveness of two different gesture types in navigation. Table 4.4 shows the performance at test scenes evaluated at a different number of stops. We should see that both SR and SPL increase with the number of allowed stops, and the improvement of SR and SPL with gestures is more evident in a lower number of allowed stops.

4.4.3 Ablation Studies

The previous section explains the agent performance from the navigation perspective, *i.e.*, how the agent *communicates to learn*. It demonstrates that using additional gestural observations in the end-to-end RL model could greatly enhance its performance. Nonetheless, another issue coupled with it is how the agent *learns to communicate*. It can be viewed as if the agent understands the underlined semantics of gestures. To this end, we conduct a few ablation studies from different perspectives to give further insights of our methods.

Multi-instance scene: As shown in Fig. 3.10(a), one of the main challenges for our object navigation task is to find the object when there are *multiple instances* of the target category in the same scene (we call it a *multi-instance* scene). According to a recent summary about object navigation in embodied AI [135], almost all the navigation tasks select the nearest object with the target category to the starting position as the target object. This asks for the generalization ability on common objects for the agent. Instead, we ponder on

Methods	SR (%)	SPL (%)	DTS (m)
Random	8.7	7.1	2.908
VisSem	6.5	5.8	2.772
+Referencing	11.2	9.4	2.769
+Intervention	21.26	16.1	2.614
+Both	**22.64**	**16.3**	**2.578**

Table 4.6: **Navigation performance in blind mode.** All the metrics are the evaluations results on the test set at the first stop. All methods are evaluated with visual information turned off.

working with *instances* instead of *classes* [142], and require the agent to learn to navigate to a specified object from given modalities. An ablation study on multi-instance scenes is shown inTable 4.5. We display the evaluation results for different models on the test set at the first stop, with SR and SPL calculated with the different number of instances of the target category. Intuitively, navigation performance should drop with a higher number of instances due to the ambiguity on the target, and it is testified on the table regardless of the methods. However, we observe that methods with gestures are less vulnerable to the number of object instances, and the effectiveness of gestures is more noticeable with a higher number of object instances. This demonstrates that the learning agent is able to locate the target instance with the gestural and contextual information.

Blind mode: In this ablation study, we turn off all the visual perceptions for methods during evaluation. We aim to find out if gestures can still give instructional cues even without any visual contextual observations. We call this type of inference as *blind mode.* As shown in Table 4.6, VisSem model performance drops inferior to the random baseline method, indicating that the semantic label alone does not provide any useful guidance. Nevertheless, agents with gestures can still navigate with performance better than the random model, especially with the intervention gestures. It demonstrates that the semantic cues from gestural inputs are still useful for navigation without visual observations. Especially, referencing gestures provide directional cues that can instruct the agent to at least move to a correct area of interest. For intervention gestures, it provides online guidance that also leads the agent to the proximity of the target object.

Methods	SR (%)	SPL (%)	DTS (m)
RGB+Semantic+Both	41.4	23.2	1.988
Depth+Semantic+Both	41.0	24.8	2.012
RGBD+Semantic+Both	**49.2**	**28.7**	**1.976**

Table 4.7: **Ablation study on visual perceptions.** We choose the VisSem+Both model and change only the vision part. All the metrics are the evaluations results on the test set at the first stop.

Visual ablation: We have been considering the significance of gestures for the learning process, but the agent navigates in a multimodal communicative manner. This visual ablation study shows the importance of different visual modalities (see Table 4.7). We can see that RGB or depth images alone render significantly lower navigation performance, indicating that multiple modalities can help with the learning process.

4.4.4 Qualitative Results

To visualize the effectiveness of our methods, we show some qualitative results in Figs. 4.2 and 4.3. Fig. 4.2 compares our referencing gesture model against the baseline model with visualized trajectories in different scenes and targets. It could be observed that in all scenes, our referencing gesture model enables the agent to navigate to the target more intelligently, while the baseline model often struggles to find the target and stop or takes a longer path to find the target. Fig. 4.3 demonstrates how our intervention gesture model works to improve navigation significantly. In this example, the agent rotates at the place where it faces back to the target and is instructed with interventions gestures until the target is in its field of view before making any movements. This indicates that our agent is able to understand and react to the intervention gestures, resulting in much better navigation performance.

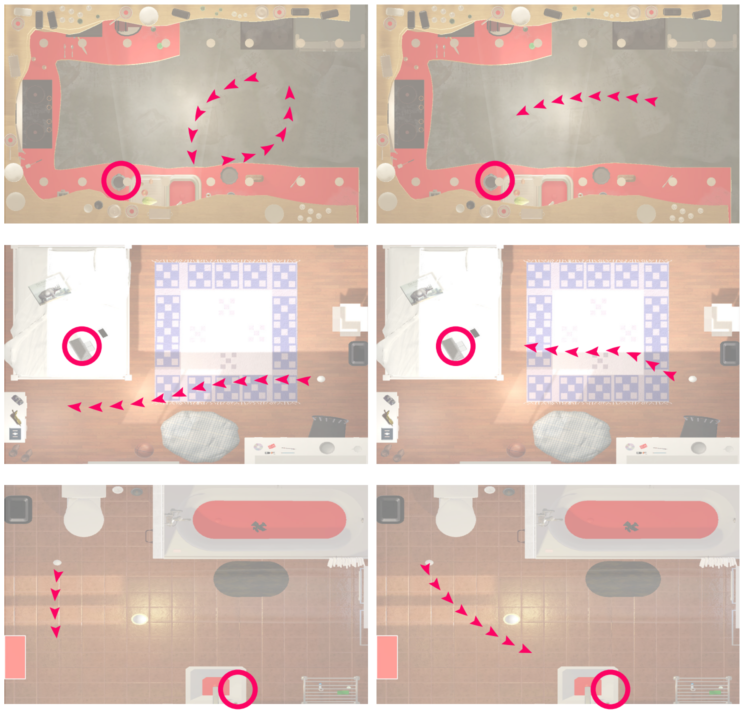
Figure 4.2: **Qualitative results with visualizations of trajectories for baseline (left) and referencing gesture (right) models.** Our agent can efficiently navigate to the target with the help of gestures.

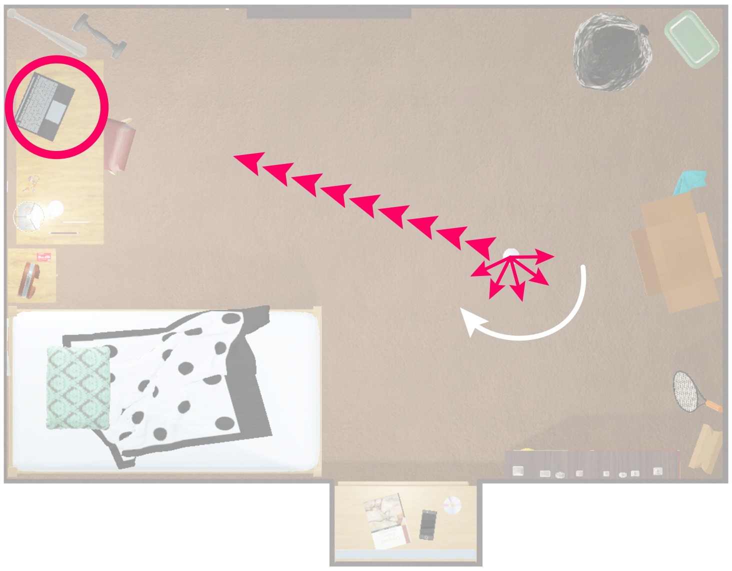
Figure 4.3: **Qualitative results for the intervention gesture model.** When the agent is back to the target, it receives interventional gestures and could first rotate until it faces the target before making any forward movements, thus increasing the navigation success rate and efficiency.

CHAPTER 5

Conclusions & Discussions

5.1 Conclusions

In this dissertation, we propose to incorporate a non-verbal communicative interface, *gesture*, into the embodied AI domain. We rethink how humans and robots could interact in simulated environments and build a new framework for embodied visual navigation where human users can give instructions to the autonomous agent using gestures.

In Chapter 2, we review studies about gestures in human-human communication and human-robot communication. From the prior work, gestures play a crucial role in information conveyance by assisting or substituting verbal interfaces in different ways. Notwithstanding the advancement in the application of gestures in various fields of HRI, we should notice that there is still scarcity in the *embodiment* of gestures. Previous work mostly focuses on learning the semantics of gestures from static datasets like images or videos. However, we realize that gestures should naturally occur with contextual information. It is astounding for robots to understand gestures in a huge space with traditional machine learning. Therefore, we propose to contextualize gestures with the *embodiment*.

In Chapter 3, we introduce our VR-based interactive learning environment, GesTHOR, based on AI2-THOR in Unity3D game engine in order to bring gestures to embodied HRI. We adopt the richful environment configurations of AI2-THOR and add interactive features to this environment by including a humanoid agent and a robot agent. We have implemented toolkits for users to act as the humanoid agent and collect gestural instructions in the VR environment using Oculus, Kinect, and LMC. Users can record programs with environmental and gestural information. We also allow the recorded programs to be replayed in Unity to emulate human-in-the-scene. This interface addresses the lack of *embodiment* for gestures in HRI because gestures are defined with contextual clues. To demonstrate the effectiveness of the VR interactive interface, we provide our gesture-based dataset, GOND, for studies on object navigation tasks with our pre-collected referencing and intervention gestures. The dataset contains a diverse collection of natural human gestures which users could utilize as a new modality in the embodied navigation task.

In Chapter 4, we have defined object navigation tasks with gestural instructions with GOND and designed an end-to-end deep reinforcement learning model for training and testing. The agent is tasked to navigate to the proximity of the target object instance with the specified category within the allowed steps. We develop our own methods with referencing and intervention gestures and compare their navigation performance to a few baseline methods. Our experimental results and ablation studies show that the agent is able to interpret human instructions with gestures and improve its visual navigation by a significant boost on the evaluation metrics. We also conclude that interactive activities during agent task execution can improve performance by showing quantitative and qualitative results.

While the main setting and experimental design of our study have been used in prior works, to the best of our knowledge, our work is the first incorporating human gestures for embodied agent learning and showing that the agent can learn the semantics of gestures without supervision. Such agents and gesture-based interface will be very useful for collaborative robots or virtual agents, and our simulator and dataset can benefit future research for HRI via gestures or other means of non-verbal communication. Our GesTHOR environment and GOND are released and publicly available for use in active studies in embodied AI.

5.2 Limitations

Notwithstanding the promising work reported in this dissertation, there are a few limitations in the following aspects:

i) We shall incorporate finer robotic actions and task planning in our framework. Currently, our robot agent is able to complete only navigation tasks, and a *mobility*-to-*manipulation* transition is yet unexplored. Human activities are intricate and involve not only motion planning but also fine-grained body control. It is, therefore, natural to devise an intelligent robot to respond to gestures by employing more body parts. For instance, when we point to a *cup* while uttering "*pick up the cup*", the robot agent could use an embodied robotic arm and hand to grab and hold the cup in the scene. Robot manipulation has been studied in recent work [118, 126]. In view of their findings, we can see that it is not easy to have the transition from mobility to manipulation. This is based on the advanced physics in the simulator and well-defined atomic actions. Our simulator has only low-level physics, *e.g.*, rigid body collisions. To support higher-level interactions, simulation on the surfaces and forces upon contact needs to be established. Besides, the robotic embodiment does not have hand configuration, which means that grasping actions can only be abstracted by an end effector, as what has been demonstrated in [126]. We should be aware that object grasping is a challenging area in the robotics community, and this functionality could be included in future research.

ii) In our gestures-based dataset, GOND, all referencing gestures and interventions gestures are single continuous motions as uncomplicated demonstrations of natural human gestures. We excluded the multistage gesture motions, which could occur naturally in daily activities. For example, for referencing gestures, instead of pointing to the target object directly, we can point to another location first as a sub-goal for the agent and then point to the target. This gestural information indicates hierarchical path planning and can help the agent in extremely difficult navigation tasks, *e.g.*, in an outdoor scene or clustered room.

We have conducted experiments and testified that the simplistic natural gestures we defined can be beneficial to the embodied navigation task, and we hope this can lay a solid ground for studies on more complicated gestures.

iii) The referencing gestures and intervention gestures are designed as contextual-dependent non-verbal cues to help the agent navigate. Although we include a diverse collection of gesture styles in GOND, we believe deeper semantic messages can be discovered by the agent. Let us take the intervention gestures as an example. The robot agent in our framework correlates the intervention gestures with the environment and navigation policy by recognizing a correct path. We expect the agent to turn or navigate in other directions rather than stay on the current path to get closer to the target. However, human shows the intensity of the intervention by changing, *e.g.*, the frequency of waving hands. In this case, the agent needs to find out the manifold semantic information. When we wave at a fast pace, the agent could interpret this as a strong signal to immediately stop and rotate. The case mentioned here may sound astounding in the first place, but we believe as the robotic cognition system evolves progressively, this challenge is worth exploring in later work.

iv) As shown in [41], the interactions between the humanoid agent and the robot agent at the training stage are *pre-recorded*. Although they come from real embodied HRI of human users, the training is not strictly conducted with *human-in-the-scene*. This allows us to deploy the training processes in scale since real-time interactions in training are infeasible. There is a trade-off between the speed of training and the authenticity of interactive gestures. Apparently, all gestures are evaluated in the same dataset, and the generalization and compatibility to online interactive gestures are still open to solutions. That being said, we allow inference with the actual interactions between a human user and the robot agent. Studies can start by evaluating the effectiveness of the learned models during the testing stage.

v) We show the generalization of our navigation agent to unseen scenes in selected rooms from AI2-THOR. Note that rooms used for training and evaluations share a similar synthetic style, although the layouts are different. Generalization to scenes with different styles (*e.g.*, from indoor scenes to outdoor scenes) is not considered within the current framework. In fact, the whole embodied AI community has been suffering from the transferability of methods evaluated on different simulators due to the unstandardized simulation environments. We aim to simplify the setting of training and testing with the same set of indoor scenes to effectively demonstrate the functionalities of gestures, but the ability to generalize should also be considered as an important feature. In addition, the sim2real transfer poses a more challenging task that we have not addressed. It is the key component in the cross-section of robotics and embodied AI and bridges the gap between simulation and real scenes. We encourage future research to be conducted to extend the generalization ability of our framework.

The aforementioned limitations are the missing components in our current work. Yet, there are other factors to be considered in the future, and we list the most important items to inspire thoughts on the improvement of our framework.

5.3 Future Directions

Our work initiates the study on embodied HRI with gestures and can be outreached to many different aspects. There are several directions to consider to extend the current research. We provide visions specifically to the aforementioned limitations in Section 5.2.

i) One could extend our current object navigation task to multi-task object navigation and manipulation. Aside from navigating to the target, the robot agent can be asked to complete sub-tasks, *e.g.*, bring the object back and pick up the object. Different from the most recent work in [30], which maps natural language actions to sequential actions, we think non-verbal instructions can also be included in the process. It is worth noting that our robot agent supports fine-grained robotic actions with a six DoF robotic arm so that a more complex task can be implemented without additional efforts. We could extend the current action space to include the atomic actions from the robot arm as illustrated in [173] to complete tasks like picking up and rearranging objects using the end effector. If hand motions are required for grasping simulation, an embodied hand model can be integrated with the current robot agent. Besides, our navigation methods could be integrated with any complex tasks that are based on the object navigation task. We can have a modular learning network that utilizes our current network as the first building block to give the object semantic clues. We believe that deeper network works could also benefit learning with sub-tasks. Grounding verbal and gestural instructions to low-level tasks are challenging and thus may require a larger network to capture the huge input space. In addition, the intelligent agent can also learn to communicate with gestures which in turn will help humans to utilize even more diverse gestures to communicate with agents. We encourage future work to include a humanlike robot agent with arms and hands to visualize the full-body gestures.

ii) As to the diversity and complexity of gestures, human gestures can be manifold that contain procedural information. Our VR interactive interface supports tracking gestures with unlimited styles, and users can easily define their own type of gestures. We can record piecewise gestures and define tasks that correspond to the segmented information. Besides, we could also inject more intricate gestural semantics. For example, as mentioned in Section 5.2, users can express their willingness and emotions through gestures. Considering the increasing variety of gestures, we need to endow the agent with more intelligent cognition to interpret and react to gestures. In terms of the RL learning process, we can do reward shaping to regularize the behavior of the robot agent. For example, in a scene with multiple tasks, completing each individual task will receive partial positive rewards. This boils down to carefully devising the learning objectives, which could be much different from the current setup.

iii) There are two main limitations discussed in regards to the generalization of our methods: generalization to novel gestures and scenes. As to unseen gestures, we understand that evaluation with real-time interactive gestures is astounding. We can design a system that generates automated gestures given contextual information. For instance, given the target location, room layouts, and the human position, a generative model can be used to synthesize a humanlike gesture motion. This requires a large amount of gesture data to be collected in order to get the most vivid reproductions. The crowd-sourcing technique is also a good way to massively acquire the gesture data to improve the generalization ability. As to unseen scenes, we are concerned with how to transfer scenes from one type to another or from simulation to reality. Note that GesTHOR is highly customizable and compatible with many different simulation environments, allowing our methods to be transferred to other scenes. Multiple passes of training can be serialized on different scene types to consolidate the effectiveness of gestures. Also, the sim2real transfer with non-verbal communicative interfaces can be testified with physical robots. A Robot Operating System (ROS) interface can be easily added in Unity to deploy the whole setup on the robot agent to the real world. The photo-realistic feature of GesTHOR can help mitigate the difficulties in this transferring due to the discrepancy from simulators to the real world. It is more convincing and promising to see our methods work in both simulation and reality.

We provide our perspectives on the current limitations and provide guidance to improve the framework. We hope our work would inspire and encourage more future work bridging the gap for current limitations in embodied HRI.

REFERENCES

[1] L. Wittgenstein, *The big typescript: TS 213.* John Wiley & Sons, 2012.

[2] J. P. Higham and E. A. Hebets, "An introduction to multimodal communication," *Behavioral Ecology and Sociobiology*, vol. 67, no. 9, pp. 1381–1388, 2013.

[3] J. Joo, E. P. Bucy, and C. Seidel, "Automated coding of televised leader displays: Detecting nonverbal political behavior with computer vision and deep learning.," *International Journal of Communication*, 2019.

[4] Z. Kang, C. Indudhara, K. Mahorker, E. P. Bucy, and J. Joo, "Understanding political communication styles in televised debates via body movements," in *European Conference on Computer Vision*, 2020.

[5] M. Tomasello, *Origins of human communication.* MIT press, 2010.

[6] C.-M. Huang and B. Mutlu, "Modeling and evaluating narrative gestures for humanlike robots.," in *Robotics: Science and Systems*, 2013.

[7] Y. Zhu, T. Gao, L. Fan, S. Huang, M. Edmonds, H. Liu, F. Gao, C. Zhang, S. Qi, Y. N. Wu, J. Tenenbaum, and S.-C. Zhu, "Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense," *Engineering*, vol. 6, no. 3, pp. 310–345, 2020.

[8] I. Maurtua, I. Fernandez, A. Tellaeche, J. Kildal, L. Susperregi, A. Ibarguren, and B. Sierra, "Natural multimodal communication for human–robot collaboration," *International Journal of Advanced Robotic Systems*, vol. 14, no. 4, 2017.

[9] E. Torta, J. van Heumen, F. Piunti, L. Romeo, and R. Cuijpers, "Evaluation of unimodal and multimodal communication cues for attracting attention in human–robot interaction," *International Journal of Social Robotics*, vol. 7, no. 1, pp. 89–96, 2015.

[10] B. Gleeson, K. MacLean, A. Haddadi, E. Croft, and J. Alcazar, "Gestures for industry intuitive human-robot communication from human observation," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2013.

[11] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *arXiv:2103.04918*, 2021.

[12] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[13] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *CVPR Workshops*, 2018.

[14] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "Iqa: Visual question answering in interactive environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[15] D. L. Chen and R. J. Mooney, "Learning to interpret natural language navigation instructions from observations," in *AAAI Conference on Artificial Intelligence*, 2011.

[16] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, "Gated-attention architectures for task-oriented language grounding," *arXiv:1706.07230*, 2017.

[17] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Neural modular control for embodied question answering," *arXiv:1810.11181*, 2018.

[18] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[19] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[20] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation," in *Conference on Neural Information Processing Systems*, 2018.

[21] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning*, 2020.

[22] W. Hao, C. Li, X. Li, L. Carin, and J. Gao, "Towards learning a generic agent for vision-and-language navigation via pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[23] F. Zhu, Y. Zhu, X. Chang, and X. Liang, "Vision-language navigation with self-supervised auxiliary reasoning tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[24] M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht, "Alfworld: Aligning text and embodied environments for interactive learning," *arXiv:2010.03768*, 2020.

[25] H. Joo, T. Simon, M. Cikara, and Y. Sheikh, "Towards social artificial intelligence: Nonverbal social signal prediction in a triadic interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[26] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "Minos: Multimodal indoor simulator for navigation in complex environments," *arXiv:1712.03931*, 2017.

[27] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[28] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, *et al.*, "Habitat: A platform for embodied ai research," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[29] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *arXiv:1709.06158*, 2017.

[30] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[31] M. Tomasello and M. J. Farrar, "Joint attention and early language," *Child development*, pp. 1454–1463, 1986.

[32] J. Joo, F. F. Steen, and M. Turner, "Red hen lab: Dataset and tools for multimodal human communication research," *KI-Künstliche Intelligenz*, vol. 31, no. 4, pp. 357–361, 2017.

[33] L. Fan, S. Qiu, Z. Zheng, T. Gao, S.-C. Zhu, and Y. Zhu, "Learning triadic belief dynamics in nonverbal communication from videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[34] M. Hasanuzzaman, V. Ampornaramveth, T. Zhang, M. Bhuiyan, Y. Shirai, and H. Ueno, "Real-time vision-based gesture recognition for human robot interaction," in *International Conference on Robotics and Biomimetics*, 2004.

[35] K. Nickel and R. Stiefelhagen, "Visual recognition of pointing gestures for human–robot interaction," *Image and Vision Computing*, vol. 25, no. 12, pp. 1875–1884, 2006.

[36] B. S. Ertuğrul, C. Gurpinar, H. Kivrak, and H. Kose, "Gesture recognition for humanoid assisted interactive sign language tutoring," in *Signal Processing and Communications Applications Conference*, 2013.

[37] C. Nuzzi, S. Pasinetti, M. Lancini, F. Docchio, and G. Sansoni, "Deep learning-based hand gesture recognition for collaborative robots," *IEEE Instrumentation & Measurement Magazine*, vol. 22, no. 2, pp. 44–51, 2019.

[38] J. Chang, J. Xiao, J. Chai, and Z. Zhou, "An improved faster r-cnn algorithm for gesture recognition in human-robot interaction," in *Chinese Automation Congress*, 2019.

[39] K. Jiang, S. Stacy, C. Wei, A. Chan, F. Rossano, Y. Zhu, and T. Gao, "Individual vs. joint perception: a pragmatic model of pointing as communicative smithian helping," in *Cognitive Science*, 2021.

[40] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv:1712.05474*, 2017.

[41] ©2021 IEEE Reprinted with permission from, Q. Wu, C.-J. Wu, Y. Zhu, and J. Joo, "Communicative learning with natural gestures for embodied navigation agents with human-in-the-scene," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[42] M. Turk, "Multimodal interaction: A review," *Pattern recognition letters*, vol. 36, pp. 189–195, 2014.

[43] M. Fröhlich, C. Sievers, S. W. Townsend, T. Gruber, and C. P. van Schaik, "Multimodal communication and language origins: integrating gestures and vocalizations," *Biological Reviews*, vol. 94, no. 5, pp. 1809–1829, 2019.

[44] F. F. Steen, A. Hougaard, J. Joo, I. Olza, C. P. Cánovas, *et al.*, "Toward an infrastructure for data-driven multimodal communication research," *Linguistics Vanguard*, vol. 4, no. 1, 2018.

[45] F. Steen and M. B. Turner, "Multimodal construction grammar," *Language and the Creative Mind*, 2013.

[46] M. Tomasello *et al.*, "Joint attention as social cognition," *Joint attention: Its origins and role in development*, vol. 103130, pp. 103–130, 1995.

[47] V. L. Tobin, *Literary joint attention: social cognition and the puzzles of modernism.* University of Maryland, College Park, 2008.

[48] A. Kendon, *Conducting interaction: Patterns of behavior in focused encounters*, vol. 7. CUP Archive, 1990.

[49] Y. Chen, Q. Li, D. Kong, Y. L. Kei, S.-C. Zhu, T. Gao, Y. Zhu, and S. Huang, "Yourefit: Embodied reference understanding with language and gesture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[50] J. M. Iverson and S. Goldin-Meadow, "Gesture paves the way for language development," *Psychological science*, vol. 16, no. 5, pp. 367–371, 2005.

[51] C. Colonnesi, G. J. J. Stams, I. Koster, and M. J. Noom, "The relation between pointing and language development: A meta-analysis," *Developmental Review*, vol. 30, no. 4, pp. 352–366, 2010.

[52] S. Goldin-Meadow, "Beyond words: The importance of gesture to researchers and learners," *Child development*, vol. 71, no. 1, pp. 231–239, 2000.

[53] D. McNeill, *Hand and mind*. De Gruyter Mouton, 2011.

[54] C. L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami, "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction," in *International Conference on Robot and Human Interactive Communication*, 2005.

[55] E. T. Hall and E. T. Hall, *The dance of life: The other dimension of time*. Anchor, 1984.

[56] W. S. Condon and W. D. Ogston, "A segmentation of behavior.," *Journal of psychiatric research*, 1967.

[57] H. St and N. Moore, "Nonverbal communication: studies and applications," 2004.

[58] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[59] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand pointnet: 3d hand pose estimation using point sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[60] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[61] S. Yang, J. Liu, S. Lu, M. H. Er, and A. C. Kot, "Collaborative learning of gesture recognition and 3d hand pose estimation with multi-order feature analysis," in *European Conference on Computer Vision*, 2020.

[62] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, “First-person hand action benchmark with rgb-d videos and 3d hand pose annotations,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[63] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, “Recurrent network models for human dynamics,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[64] V. Mnih, H. Larochelle, and G. E. Hinton, “Conditional restricted boltzmann machines for structured output prediction,” *arXiv:1202.3748*, 2012.

[65] R. Gross and J. Shi, “The cmu motion of body (mobo) database,” 2001.

[66] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[67] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, *et al.*, “Panoptic studio: A massively multiview system for social interaction capture,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 1, pp. 190–204, 2017.

[68] T. B. Sheridan, “Human–robot interaction: status and challenges,” *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.

[69] J. Y. Chen and M. J. Barnes, “Human–robot interaction,” *Handbook of Human Factors and Ergonomics*, pp. 1121–1142, 2021.

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

[71] D. Manzey, J. Reichenbach, and L. Onnasch, “Human performance consequences of automated decision aids: The impact of degree of automation and system experience,” *Journal of Cognitive Engineering and Decision Making*, vol. 6, no. 1, pp. 57–87, 2012.

[72] M. Lewis, “Human interaction with multiple remote robots,” *Reviews of Human Factors and Ergonomics*, vol. 9, no. 1, pp. 131–174, 2013.

[73] A. Henschel, R. Hortensius, and E. S. Cross, “Social cognition in the age of human–robot interaction,” *Trends in Neurosciences*, vol. 43, no. 6, pp. 373–384, 2020.

[74] S. N. Young and J. M. Peschel, “Review of human–machine interfaces for small unmanned systems with robotic manipulators,” *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 2, pp. 131–143, 2020.

[75] N. Mavridis, "A review of verbal and non-verbal human–robot interactive communication," *Robotics and Autonomous Systems*, vol. 63, pp. 22–35, 2015.

[76] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition."

[77] P. Ekman and W. V. Friesen, *The repertoire of nonverbal behavior: Categories, origins, usage, and coding.* De Gruyter Mouton, 2010.

[78] Y. Che, H. Culbertson, C.-W. Tang, S. Aich, and A. M. Okamura, "Facilitating human-mobile robot communication via haptic feedback and gesture teleoperation," *ACM Transactions on Human-Robot Interaction (THRI)*, vol. 7, no. 3, pp. 1–23, 2018.

[79] M. E. Walker, H. Hedayati, and D. Szafir, "Robot teleoperation with augmented reality virtual surrogates," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2019.

[80] H. Liu and L. Wang, "Gesture recognition for human-robot collaboration: A review," *International Journal of Industrial Ergonomics*, vol. 68, pp. 355–367, 2018.

[81] S. Saunderson and G. Nejat, "How robots influence humans: A survey of nonverbal communication in social human–robot interaction," *International Journal of Social Robotics*, vol. 11, no. 4, pp. 575–608, 2019.

[82] Y. Katsuki, Y. Yamakawa, and M. Ishikawa, "High-speed human/robot hand interaction system," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, 2015.

[83] M. Elmezain, A. Al-Hamadi, J. Appenrodt, and B. Michaelis, "A hidden markov model-based continuous gesture recognition system for hand motion trajectory," in *International Conference on Pattern Recognition*, 2008.

[84] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Communications of the ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[85] H. Liu, X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S.-C. Zhu, "A glove-based system for studying hand-object manipulation via joint pose and force sensing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[86] H. Liu, Z. Zhang, X. Xie, Y. Zhu, Y. Liu, Y. Wang, and S.-C. Zhu, "High-fidelity grasping in virtual reality using a glove-based system," in *IEEE International Conference on Robotics and Automation*, 2019.

[87] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Symposium on User Interface Software and Technology*, 2016.

[88] B. Li, C. Zhang, C. Han, and B. Bai, "Gesture recognition based on kinect v2 and leap motion data fusion," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 05, p. 1955005, 2019.

[89] J. Letessier and F. Bérard, "Visual tracking of bare fingers for interactive surfaces," in *Symposium on User Interface Software and Technology*, 2004.

[90] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[91] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect.," in *British Machine Vision Conference*, 2011.

[92] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

[93] Q. Xiao, M. Qin, and Y. Yin, "Skeleton-based chinese sign language recognition and generation for bidirectional communication between deaf and hearing people," *Neural networks*, vol. 125, pp. 41–55, 2020.

[94] S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, "Skeleton aware multi-modal sign language recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[95] S. Chen, H. Ma, C. Yang, and M. Fu, "Hand gesture based robot control system using leap motion," in *International Conference on Intelligent Robotics and Applications*, 2015.

[96] Q. Gao, J. Liu, Z. Ju, Y. Li, T. Zhang, and L. Zhang, "Static hand gesture recognition with parallel cnns for space human-robot interaction," in *International Conference on Intelligent Robotics and Applications*, 2017.

[97] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[98] J. McCormick, K. Vincs, S. Nahavandi, D. Creighton, and S. Hutchison, "Teaching a digital performing agent: Artificial neural network and hidden markov model for recognising and performing dance movement," in *International Conference on Movement and Computing*, 2014.

[99] A. Cenedese, G. A. Susto, G. Belgioioso, G. I. Cirillo, and F. Fraccaroli, "Home automation oriented gesture classification from inertial measurements," *IEEE Transactions on automation science and engineering*, vol. 12, no. 4, pp. 1200–1210, 2015.

[100] A. Jain, J. Tompson, Y. LeCun, and C. Bregler, "Modeep: A deep learning framework using motion features for human pose estimation," in *Asian conference on computer vision*, pp. 302–315, Springer, 2014.

[101] B. Mutlu, N. Roy, and S. Šabanović, "Cognitive human–robot interaction," *Springer Handbook of Robotics*, pp. 1907–1934, 2016.

[102] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[103] G. A. Kaminka, "Curing robot autism: A challenge," in *International Conference on Autonomous Agents and Multiagent Systems*, 2013.

[104] D. Szafir, B. Mutlu, and T. Fong, "Communication of intent in assistive free flyers," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2014.

[105] Y. Zhang, S. Sreedharan, A. Kulkarni, T. Chakraborti, H. H. Zhuo, and S. Kambhampati, "Plan explicability and predictability for robot task planning," in *IEEE International Conference on Robotics and Automation*, 2017.

[106] O. Liu, D. Rakita, B. Mutlu, and M. Gleicher, "Understanding human-robot interaction in virtual reality," in *International Conference on Robot and Human Interactive Communication*, 2017.

[107] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[108] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*, 2014.

[109] L. Smith and M. Gasser, "The development of embodied cognition: Six lessons from babies," *Artificial life*, vol. 11, no. 1-2, pp. 13–29, 2005.

[110] C. Beattie, J. Z. Leibo, D. Teplyashin, T. Ward, M. Wainwright, H. Küttler, A. Lefrancq, S. Green, V. Valdés, A. Sadik, *et al.*, "Deepmind lab," *arXiv:1612.03801*, 2016.

[111] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," in *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 2016.

[112] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville, "Home: A household multimodal environment," *arXiv:1711.11017*, 2017.

[113] Y. Wu, Y. Wu, G. Gkioxari, and Y. Tian, “Building generalizable agents with a realistic and rich 3d environment,” *arXiv:1801.02209*, 2018.

[114] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, “Virtualhome: Simulating household activities via programs,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[115] C. Yan, D. Misra, A. Bennnett, A. Walsman, Y. Bisk, and Y. Artzi, “Chalet: Cornell house agent learning environment,” *arXiv:1801.07357*, 2018.

[116] X. Gao, R. Gong, T. Shu, X. Xie, S. Wang, and S.-C. Zhu, “Vrkitchen: an interactive 3d virtual environment for task-oriented learning,” *arXiv:1903.05757*, 2019.

[117] M. Chevalier-Boisvert, D. Bahdanau, S. Lahlou, L. Willems, C. Saharia, T. H. Nguyen, and Y. Bengio, “Babyai: A platform to study the sample efficiency of grounded language learning,” in *International Conference on Learning Representations*, 2018.

[118] C. Gan, J. Schwartz, S. Alter, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, M. Sano, *et al.*, “Threedworld: A platform for interactive multi-modal physical simulation,” *arXiv:2007.04954*, 2020.

[119] B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, S. Buch, C. D’Arpino, S. Srivastava, L. P. Tchapmi, *et al.*, “igibson, a simulation environment for interactive tasks in large realistic scenes,” *arXiv:2012.02924*, 2020.

[120] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, *et al.*, “Sapien: A simulated part-based interactive environment,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[121] J. Duan, S. Yu, H. L. Tan, and C. Tan, “Actionet: An interactive end-to-end platform for task-based data collection and augmentation in 3d environment,” in *IEEE International Conference on Image Processing*, 2020.

[122] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, “Semantic scene completion from a single depth image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[123] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, *et al.*, “The replica dataset: A digital replica of indoor spaces,” *arXiv:1906.05797*, 2019.

[124] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The arcade learning environment: An evaluation platform for general agents,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[125] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv:1606.01540*, 2016.

[126] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi, "Manipulathor: A framework for visual object manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[127] X. Xie, H. Liu, Z. Zhang, Y. Qiu, F. Gao, S. Qi, Y. Zhu, and S.-C. Zhu, "Vrgym: A virtual testbed for physical and interactive ai," in *Proceedings of the ACM Turing Celebration Conference*, 2019.

[128] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su, "Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[129] P. D. Nguyen, Y. K. Georgie, E. Kayhan, M. Eppe, V. V. Hafner, and S. Wermter, "Sensorimotor representation learning for an "active self" in robots: a model survey," *KI-Künstliche Intelligenz*, vol. 35, no. 1, pp. 9–35, 2021.

[130] T. Chen, S. Gupta, and A. Gupta, "Learning exploration policies for navigation," *arXiv:1903.01959*, 2019.

[131] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *CVPR Workshops*, 2017.

[132] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," *arXiv:2004.05155*, 2020.

[133] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, "Occupancy anticipation for efficient exploration and navigation," in *European Conference on Computer Vision*, 2020.

[134] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman, "An exploration of embodied visual exploration," *International Journal of Computer Vision*, vol. 129, no. 5, pp. 1616–1649, 2021.

[135] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv:1807.06757*, 2018.

[136] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[137] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv:1911.00357*, 2019.

[138] D. S. Chaplot, D. Gandhi, A. Gupta, and R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *arXiv:2007.00643*, 2020.

[139] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *IEEE International Conference on Robotics and Automation*, 2017.

[140] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[141] F. Zeng, C. Wang, and S. S. Ge, "A survey on visual navigation for artificial agents with deep reinforcement learning," *IEEE Access*, vol. 8, pp. 135426–135442, 2020.

[142] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," *arXiv:2006.13171*, 2020.

[143] D. Mishkin, A. Dosovitskiy, and V. Koltun, "Benchmarking classic and learned navigation in complex 3d environments," *arXiv:1901.10915*, 2019.

[144] J. Ye, D. Batra, E. Wijmans, and A. Das, "Auxiliary tasks speed up learning pointgoal navigation," *arXiv:2007.04561*, 2020.

[145] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[146] S. Koenig and M. Likhachev, "D* lite," *Aaai/iaai*, vol. 15, 2002.

[147] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[148] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *International Conference on Learning Representations*, 2019.

[149] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *IEEE International Conference on Robotics and Automation*, 2020.

[150] M. Lohmann, J. Salvador, A. Kembhavi, and R. Mottaghi, "Learning about objects by learning to interact with them," *arXiv:2006.09306*, 2020.

[151] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental Robotics*, 2013.

[152] D. Misra, A. Bennett, V. Blukis, E. Niklasson, M. Shatkhin, and Y. Artzi, "Mapping instructions to actions in 3d environments with visual goal prediction," *arXiv:1809.00786*, 2018.

[153] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015.

[154] Y. Niu, H. Zhang, M. Zhang, J. Zhang, Z. Lu, and J.-R. Wen, "Recursive visual attention in visual dialog," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[155] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, "Towards ai-complete question answering: A set of prerequisite toy tasks," *arXiv:1502.05698*, 2015.

[156] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *TPAMI*, vol. 39, no. 4, pp. 652–663, 2016.

[157] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, "Ok-vqa: A visual question answering benchmark requiring external knowledge," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[158] H. Yu, H. Zhang, and W. Xu, "Interactive grounded language acquisition and generalization in a 2d world," *arXiv:1802.01433*, 2018.

[159] T. Cao, J. Wang, Y. Zhang, and S. Manivasagam, "Babyai++: Towards grounded-language learning beyond memorization," *arXiv:2004.07200*, 2020.

[160] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, *et al.*, "Grounded language learning in a simulated 3d world," *arXiv:1706.06551*, 2017.

[161] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[162] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *IEEE International Conference on Robotics and Automation*, 2019.

[163] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[164] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[165] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[166] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[167] C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan, "Is q-learning provably efficient?," *arXiv:1807.03765*, 2018.

[168] S. Schaal *et al.*, "Learning from demonstration," *Advances in neural information processing systems*, pp. 1040–1046, 1997.

[169] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *International Conference on Artificial Intelligence and Statistics*, 2011.

[170] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016.

[171] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[172] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International Conference on Machine Learning*, 2018.

[173] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, *et al.*, "Robothor: An open simulation-to-real embodied ai platform," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[174] N. Botteghi, B. Sirmacek, K. A. Mustafa, M. Poel, and S. Stramigioli, "On reward shaping for mobile robot navigation: A reinforcement learning and slam based approach," *arXiv:2002.04109*, 2020.

[175] C. Chen, Z. Al-Halah, and K. Grauman, "Semantic audio-visual navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[176] J. Y. Chai, Q. Gao, L. She, S. Yang, S. Saba-Sadiya, and G. Xu, "Language to action: Towards interactive task learning with physical agents.," in *International Joint Conference on Artificial Intelligence*, 2018.

[177] T.-C. Chi, M. Eric, S. Kim, M. Shen, and D. Hakkani-tur, "Just ask: An interactive learning framework for vision and language navigation," *arXiv:1912.00915*, 2019.

[178] F. Landi, L. Baraldi, M. Corsini, and R. Cucchiara, "Embodied vision-and-language navigation with dynamic convolutional filters," *arXiv:1907.02985*, 2019.

[179] H. Du, X. Yu, and L. Zheng, "Learning object relation graph and tentative policy for visual navigation," in *European Conference on Computer Vision*, 2020.

[180] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman, "Soundspaces: Audio-visual navigation in 3d environments," in *European Conference on Computer Vision*, 2020.

[181] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in *Conference on Robot Learning*, 2021.

[182] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.

[183] A. De Giorgio, M. Romero, M. Onori, and L. Wang, "Human-machine collaboration in virtual reality for adaptive production engineering," *Procedia manufacturing*, vol. 11, pp. 1279–1287, 2017.

[184] T. Guzsvinecz, V. Szucs, and C. Sik-Lanyi, "Suitability of the kinect sensor and leap motion controller—a literature review," *Sensors*, vol. 19, no. 5, p. 1072, 2019.

[185] A. A. Kadethankar and A. D. Joshi, "Dynamic hand gesture recognition using kinect," in *Innovations in Power and Advanced Computing Technologies*.

[186] X.-L. Guo and T.-T. Yang, "Gesture recognition based on hmm-fnn model using a kinect," *Journal on Multimodal User Interfaces*, vol. 11, no. 1, pp. 1–7, 2017.

[187] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188–1192, 2016.

[188] R. B. Mapari and G. Kharat, “Real time human pose recognition using leap motion sensor,” in *IEEE International Conference on Research in Computational Intelligence and Communication Networks*, 2015.

[189] N. Garcia, M. Otani, C. Chu, and Y. Nakashima, “Knowit vqa: Answering knowledge-based questions about videos,” in *AAAI Conference on Artificial Intelligence*, 2020.

[190] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” *arXiv:1506.02438*, 2015.

[191] L. Weihs, J. Salvador, K. Kotar, U. Jain, K.-H. Zeng, R. Mottaghi, and A. Kembhavi, “Allenact: A framework for embodied ai research,” *arXiv:2008.12760*, 2020.

[192] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *arXiv:1912.01703*, 2019.

[193] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv:1412.6980*, 2014.

[194] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[195] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv:1406.1078*, 2014.