UNIVERSITY OF CALIFORNIA

Los Angeles

Multi-Modal Robotic Learning, Reasoning and Planning

A dissertation submitted in partial satisfaction

of the requirements for the degree

Doctor of Philosophy in Statistics

by

Feng Gao

2022

ABSTRACT OF THE DISSERTATION

Multi-Modal Robotic Learning, Reasoning and Planning

by

Feng Gao
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2022
Professor Mark S. Handcock, Chair

Building an intelligent robot that is capable of collaborating with humans in daily tasks is a challenging problem. Although recent artificial intelligence research shows remarkable results in classical tasks, there is still a long way to achieve human-level intelligent robots. We need to start developing methods in terms of perception, learning, reasoning, and planning.

In this dissertation, we study multi-modal robotic learning, reasoning, and planning from three different perspectives: (i) robot imitation learning: we first introduce a series of works including hardware prototype, data collection, modeling human demonstration, and planning for robot imitation learning. (ii) multi-modal reasoning: we study multi-modal reasoning in two different tasks. We develop a dataset and models for visual abstraction reasoning with human IQ test. Additionally, we propose a visual language reasoning method for outside knowledge visual question answering. (iii) robot planning: we show our attempts in robot planning. We introduce a physically realistic virtual testbed where robots can interact with humans. In addition, we show a hierarchical reinforcement learning method for robot planning.

The dissertation of Feng Gao is approved.

Ying Nian Wu

Hongjing Lu

Chenfanfu Jiang

Mark S. Handcock, Committee Chair

University of California, Los Angeles

2022

To my parents and Yuxing

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I hereby thank my committee chair, Dr. Mark S. Handcock. He offered tremendous help to my Ph.D. journey, especially when I was in the most difficult period. He demonstrates how to be a humble leader for me.

I sincerely thank my co-advisor, Dr. Ying Nian Wu. He helped me with his wisdom and kindness when I was in the most difficult period of Ph.D. journey. He is the best example of being a pure scientist and a lifetime researcher. His pursuit of simplicity and elegance in formulating research problems always encourages me to keep improving my work. I will never forget those sunny afternoons when we had pleasant conversations on both research and life topics in his office.

I am very grateful for having Dr. Hongjing Lu, and Dr. Chenfanfu Jiang to be my committee members. Dr. Hongjing Lu opened the door to psychology research and offered hands-on opportunities for me to conduct psychology experiments. Dr. Chenfanfu Jiang, an unbelievably talented researcher, broadened my horizon in computer graphics and physics simulation.

I would also like to thank Dr. Song-Chun Zhu, who was my previous advisor, for introducing me to the field of AI research. It is my great honor to have him as my advisor in my early-stage Ph.D. study. His sharp opinions toward future trends of AI research always inspire my work.

Among all the senior students I had collaborated with, I want to specifically thank Dr. Yixin Zhu and Dr. Siyuan Qi, who were my student mentors in my early Ph.D. study. Yixin, the “big brother” of mine at UCLA, offered so much assistance in both research and daily life. He shared his experiences in research selflessly, helping me to adjust myself in my early Ph.D. life. Siyuan, a brilliant person, showed me how to be a qualified researcher and helped me to shape my own research taste.

Additionally, I want to thank all my peer collaborators. I really enjoy working with them,

and I will treasure all the shared experiences. Particularly, I want to mention the following collaborators. Baoxiong Jia, a very considerate, optimistic, and hardworking person, is one of my best friends. We spent lots of happy evenings together in the lab and in the gym, building up both our minds and body. Chi Zhang, once my roommate, inspires me with impressive self-discipline, smartness, and kindness. Xu Xie is the funniest peer I've met. Mark Edmonds is my best American friend, and I enjoy every single conversation with him. I thank Sirui Xie for being so considerate while working with me. Siyuan Huang is a great peer who is always calm and optimistic. Ping Qing, my mentor when I was a research intern at Amazon, is so patient and selfless, leading to a very pleasing collaboration. Yixin Chen, Qing Li, Shuwen Qiu, Zhenliang Zhang, Hangxin Liu, Zeyu Zhang, Ziyuan Jiao, Pan Lu, Muzhi Han are my best colleagues ever. It is my great pleasure to have you working in the same office. I also want to value my labmates whom I had worked with: Ruiqi Gao, Erik Nijkamp, Arjun Akula, Yuanlu Xu, Tianmin Shu, Ping Wei, Hanlin Zhu, Tao Yuan.

Lastly, I want to express my greatest gratitude to my parents for their selfless love and never-ending support in the past decades. The most sincere thanks go to my girlfriend, Yuxing Qiu. Her warmest support and companionship brighten my life, helping me to overcome all difficulties.

VITA

2020–2022	Teaching Assistant, Department of Statistics, UCLA.
2021	Applied Scientist Intern, Amazon Alexa AI
2017–2021	Graduate Student Researcher, Department of Statistics, UCLA.
2016–2017	Research Intern, Department of Statistics, UCLA.
2015–2017	M.S. in Computer Science, USC.
2011–2015	B.E. in Software Engineering, UESTC.

PUBLICATIONS

(* indicates Joint First Author)

Transform-Retrieve-Generate: Natural Language-Centric Outside-Knowledge Visual Question Answering. **F. Gao**, Q. Ping, G. Thattai, A. Reganti, Y.N. Wu, P. Natarajan. CVPR, 2022.

Dark, beyond Deep: A Paradigm Shift to Cognitive AI with Humanlike Common Sense. Y. Zhu, T. Gao, L. Fan, S. Huang, M. Edmonds, H. Liu, **F. Gao**, C. Zhang, S. Qi, Y.N. Wu, J.B. Tenenbaum, S.-C. Zhu. Engineering, 2020.

A Tale of Two Explanations: Enhancing Human Trust by Explaining Robot Behavior.

M. Edmonds*, **F. Gao***, H. Liu*, X. Xie*, S. Qi, B. Rothrock, Y. Zhu, Y.N. Wu, H. Lu, S.-C Zhu. Science Robotics, 2019.

VRGym: A Virtual Testbed for Physical and Interactive AI. X. Xie, H. Liu, Z. Zhang, Y. Qiu, **F. Gao**, S. Qi, Y. Zhu, S.-C. Zhu. ACM TURC, 2019.

Learning Perceptual Inference by Contrasting C. Zhang, B. Jia, **F. Gao**, Y. Zhu, H. Lu, S.-C Zhu. NeurIPS, 2019.

RAVEN: A Dataset for Relational and Analogical Visual Reasoning C. Zhang*, **F. Gao***, B. Jia, Y. Zhu, S.-C Zhu. CVPR, 2019.

Unsupervised Learning of Hierarchical Models for Hand-Object Interactions X. Xie, H. Liu, M. Edmonds, **F. Gao**, S. Qi, Y. Zhu, B. Rothrock, S.-C Zhu. ICRA 2018.

A Glove-based System for Studying Hand-Object Manipulation via Joint Pose and Force Sensing. H. Liu*, X. Xie, M. Millar*, M. Edmonds, **F. Gao**, Y. Zhu, V. J. Santos, B. Rothrock, S.-C. Zhu. IROS 2017.

Feeling the Force, Integrating Force and Pose for F luent Discovery through Imitation Learning to Open Medicine Bottles. M. Edmonds*, **F. Gao***, X. Xie, H. Liu, S. Qi, Y. Zhu, B. Rothrock, S.-C. Zhu. IROS, 2017.

CHAPTER 1

Introduction

We, humans, show unparalleled intelligence compared to any existing Artificial Intelligence (AI) and robotics techniques. Human beings are able to (i) learn from a very small amount of examples, (ii) generalize knowledge to unseen cases, (iii) reason over tasks with multimodality and (iv) plan under unseen tasks.

Consider such a scenario that could happen in the future: You may have a robot mate to help you with your housework. You want it to be smart enough to collaborate with you so that you only need to teach it with as least examples as possible. Additionally, you want it to be your friend. You are willing to share your life with this "soul mate" meanwhile, it is able to have joyful conversations and share knowledge with you. Furthermore, it is also capable of handling new cases when without your further instruction. How could life be better with such robot mates?

In order to design and build robots that are able to collaborate with humans, understand our minds and improve our life, we need robots that are capable of learning from human demonstrations, understanding human knowledge, and generalizing in various scenarios without extra human instructions. Fortunately, the trend of current AI research shows a promising future for its applications in robotics. In recent years, with the renaissance of neural network (NN) study, Deep Learning (DL) techniques have helped Computer Vision (CV), Machine Learning (ML), and robotics research to achieve remarkable results.

Classical CV tasks such as object detection [DDS09, LMB14], had been well solved by both traditional ML-based [FGM10] and DL-based techniques [HZR16, RHG15, RDG16].

Multi-modal tasks have been the new challenges in CV community. Visual question answering (VQA) [AAL15] and embodied AI [SKM19] are two of the most representative tasks that involve multi-modal data. The agent is not only required to understand visual information but also others modalities such as natural language, force sensing, etc. Besides the great leap in CV research, we also witnessed the blooming of ML and task planning research in the past years. The iconic Go-playing agent [SHM16, SSS17a] had demonstrated that it is plausible to achieve super-human performance with well-defined domains and representations. In addition, RL researchers took advantages of DL techniques in CV to outperform human in certain arcade games. In the field of robotics research, it is popular to apply ML-based methods to improve the performance in various tasks such as grasping [ZSW18], imitation learning [EGX17] and navigation [FOR18].

According to the above trends in recent AI, CV and ML, and robotics research, we categorize them into three major domains: robotic learning, multi-modal reasoning, and robotics planning. We argue that these domains are crucial for future robotics research. Technically speaking, we hope that a smart robot is able to learn and reason over multi-modal input from various tasks. By correctly modeling the tasks, it can plan and interact with the world according. In this dissertation, we will unfold the details of our attempts in the following three specific domains for building smarter robots:

Robotic Imitation Learning Imitation learning is the fundamental function of an intelligent robot. Although robots with hand-coded rules seem to be widely used and more acceptable nowadays, it is very limited when we want to deploy them a wide spectrum of tasks. It is inevitable that robots need to learn from human demonstrations in the future so that they can better work with us. Therefore, in this dissertation, we study robot imitation learning with a humanoid robot. In chapter 2, we will introduce a series of works in this chapter, including hardware design, data collection, robot system deployment, imitation learning modeling and planning, and study of robot behavior explainability [LXM17, XLE18, EGX17, EGL19].

Multi-modal Reasoning Multi-modal reasoning is also a key feature of a smart robot. Humans are educated to understand the world using numerical/physical laws and logic. Can we also enable the robots to do so? Many daily tasks require simple reasoning over visual perception and natural language. If we want to have robot mates that can collaborate with us, they must be capable of understanding and reasoning over what they see (visual) and what we say (nature language). In chapter 3, we study visual reasoning in two different tasks, *i.e.* human IQ tasks and outside knowledge visual question answering [ZGJ19, ZJG19, GPT22].

Robotic Planning The ultimate goal of a smart robot is to act like a human, even outperform humans. Thus, planning in a world is one of the most important features of intelligent robots. In chapter 4, we will introduce two attempts in approaching better robot planning: (i) a physically realistic interactive environment for robot training [XLZ19] and (ii) a RL-based planning algorithm.

CHAPTER 2

Learning: Robot Imitation Learning

2.1 Building A Tactile Glove to Collect Hand-Object Interaction Data for Imitation Learning

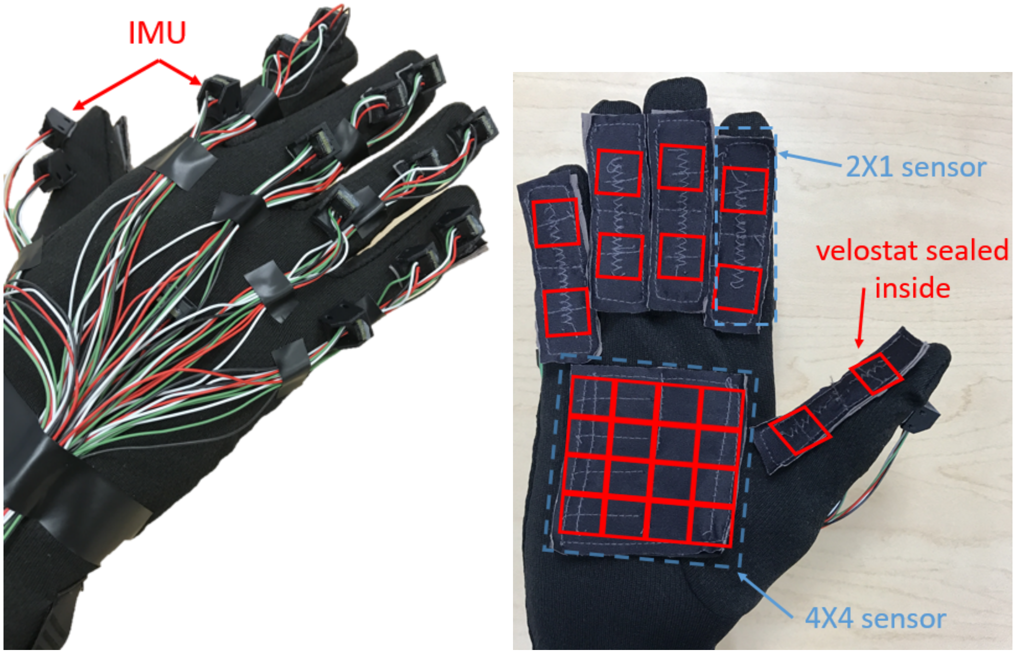
In this section, we present a design of an easy-to-replicate glove-based system that can reliably perform simultaneous hand pose and force sensing in real time, for the purpose of collecting human hand data during fine manipulative actions. The design consists of a sensory glove that is capable of jointly collecting data of finger poses, hand poses, as well as forces on palm and each phalanx. Specifically, the sensory glove employs a network of 15 IMUs to measure the rotations between individual phalanxes. Hand pose is then reconstructed using forward kinematics. Contact forces on the palm and each phalanx are measured by 6 customized force sensors made from Velostat, a piezoresistive material whose force-voltage relation is investigated. We further develop an open-source software pipeline consisting of drivers and processing code and a system for visualizing hand actions that is compatible with the popular Raspberry Pi architecture. In our experiment, we conduct a series of evaluations that quantitatively characterize both individual sensors and the overall system, proving the effectiveness of the proposed design.

2.1.1 Introduction

Robots that imitate the behaviors of humans may enable more natural and friendly interactions with humans in man-made environments, as with robotic handshaking [THG16]. Just as whole body sensing [KB12] is critical for the study of human movement, hand pose and force information is crucial to the investigation of manipulative tasks. While researchers can track hand pose based on perception [RA15], force estimation from vision using numerical differentiation methods [ZZZ15, PKQ15], or sophisticated physics-based soft-body simulation [WMZ13, ZZM13], glove-based devices still have their own advantages, presenting convenient, integrated solutions that can be natural and essential for collecting ground truth hand data during manipulations and interactions.

Designs of tactile gloves have long been proposed for a wide range of applications, and they remain an active research area. Dipietro *et al.* provided a comprehensive survey of glove-based system designs and their application from 1970s to 2008 [DSD08]. Since then, a number of novel designs have emerged to address existing limitations, including portability, reliability, and cost. As the main motivations of developing data/tactile gloves or other glove-based systems are obtaining the pose and force information during manipulative actions, we divide some notable recent designs since 2008 into two categories based on the types of data they can collect: gloves with i) only pose-sensing, and ii) joint pose- and force-sensing.

Pose sensing gloves generally utilized IMUs, flex sensors, or similar orientation systems to obtain finger joint angles. Taylor *et al.* [TKM13] tests a Zigbee network of IMUs using independent state estimation for feasibility of joint prediction. In the design by Kortier *et al.*, each of 15 phalanxes is fitted with a PCB populated with one 6 degree-of-freedom (DoF) accelerometer/gyroscope and one 3DoF magnetometer. In this way, a more comprehensive representation of the hand pose is captured [KSR14]. Ligorio *et al.* improves localization of the phalanx by combining IMUs with a camera-based localization system [LS13]. Efforts have been made to improve pose sensing accuracy using filtering [SLS15] and estimation techniques like the extended Kalman filter [KSR14, LS13, KAS15]. Using curvature/flex sensors to measure finger flexion is an approach that has been proven to be effective [KMS11, KSE08]. This approach, however, may bring discomfort to the user or sacrifice the user's dexterity. Another recent design, Wolverine [CHC16], adapts a DC motor and time-of-flight sensor into



(a) (b)

Figure 2.1: Prototype consisting of (a) 15 IMUs on the dorsum of the hand and (b) 6 integrated Velostat force sensor with 26 taxels on the palmar aspects of the hand.

an exoskeleton structure in order to obtain hand pose without using a glove directly.

Pose and force sensing glove-based systems represent efforts to combine both force sensing and pose sensing into an integrated system. Hammond *et al.* [HMW14] designs a liquid-metal embedded elastomer sensor that can measure force across the palm. The sensors measure skin strain in order to track joint motion, which may lead to less reliable measurement. In the design by Gu *et al.* [GSL15], a glove equipped with FlexiForce sensors is used to collect force information, and a Vicon motion capture system is employed to track wrist, index finger, and thumb angles for manipulative action recognition. For applications that focus more on the fingertip, a specific tactile sensor is available [BBA16]. Further related work involves estimating manipulative force from hand pose obtained via a network of 9DoF IMUs, *e.g.* [MBS16]. One potential drawback is that many of the commonly-used force sensors such as the FlexForce are built on plastic substrates that, while flexible compared to a standard PCB, can be too rigid to conform to the contours of the hand, resulting in limitations on natural hand motion during fine manipulative actions, particularly when a large force sensing area is desired. In recent years, Velostat, a piezoresistive conductive film, has become popular and is applied to pressure sensing in the fields of soft robotics [LKY15], robotic touch perception [PMT16], robotic tactile interaction [MSG15], and gesture recognition [JLK11].

The aforementioned efforts in the literature, particularly in joint pose/force sensing, indicates the needs for capturing the dynamics and not just the kinematics involved in fine manipulative actions. Such demand is especially important given that changes in hand-object interaction forces are not always accompanied by measurable changes in hand posture. The study of fine hand-object manipulative interactions requires finer spatial resolution force sensing, in combination with pose sensing, than has previously been demonstrated. The objective of this work is to create, characterize, and demonstrate an integrated system that extends pose sensing gloves with force sensing over large areas, with finer spatial resolution, and with materials that do not constrain natural hand motion.

Contributions The glove-based system presented in this work makes the following contributions:

1. The proposed design is an easy-to-replicate, cost-effective glove-based system that performs simultaneous hand pose and force sensing in real time for the purpose of studying fine manipulative actions. A configuration of IMUs similar to [KSR14] is adapted and inter-joint rotations are captured in order to reconstruct hand pose with a high degree of comprehensiveness, as shown in Figure 2.1a.

2. We design a customized force sensor using Velostat (Figure 2.1b), whose force-voltage relation is investigated, in order to capture distributions of forces over large areas of the hand rather than just at single contact points (e.g. fingertips only).

3. All software implementation of the proposed design, including the forward kinematics

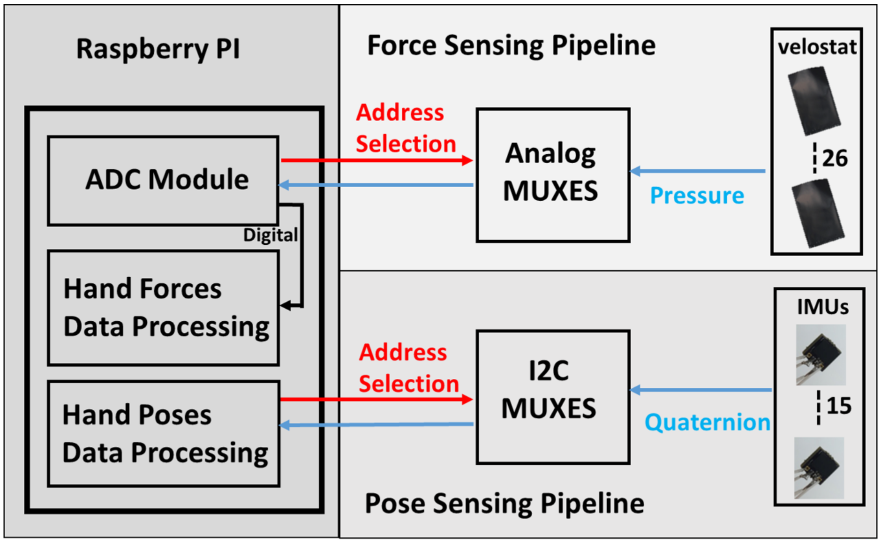


Figure 2.2: Overall system schematic

model for the hand, force vector derivation, and visualization of manipulative actions, are developed using the Robot Operating System (ROS) framework, and are publicly available at *GitHub*.

4. A prototype system is evaluated and characterized, showcasing its capability for reliably capturing dynamical information about manipulative actions. We further analyze the power consumption of the prototype system, indicating it can be powered by a small, portable power bank and be wireless to improve user's mobility.

Overview The remaining subsections under this section are organized as follows. The proposed overall design and hardware implementations of the prototype are described in subsection 2.1.2. This section also details the construction of the Velostat force sensors and its force-voltage relationships. The software implementation consisting of the forward kinematics model of the hand, force vector derivation, and visualization for manipulative actions are shown in subsection 2.1.3. The performance of the proposed design is evaluated in subsection 2.1.4 via a series of experiments.

2.1.2 Overall Design and Prototyping

This section presents the overall system schematic. In hardware implementation, a network of 15 IMUs are configured and deployed. For the force sensing pipeline, we utilized Velostat, a piezoresistive conductive film whose resistance changes in response to applied forces, to construct a force sensor that is capable of measuring contact force over a large area via an array of individual taxels. A prototype of the design is built and presented as well.

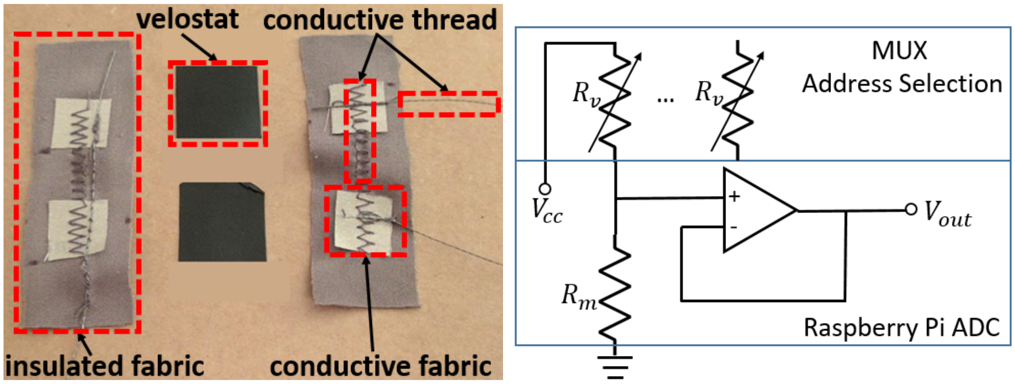
2.1.2.1 Overall Design

An integrated system consisting of a glove and a processing unit for hand pose and force acquisition is developed. Figure 2.2 shows a schematic of the integrated system deploying two sensing networks. A network of 15 IMUs is equipped to measure the orientation of the palm and each phalanx for comprehensive hand pose reconstruction. A network of 6 customized force sensors constructed from a piezoresistive conductive film—Velostat—are attached to the palm and each finger, and contact forces are measured.

2.1.2.2 Hardware Implementation

Pose sensing pipeline The pose estimation module is built from 15 Bosch BNO055 9DoF IMUs. One IMU is mounted to the palm of the glove, twelve are mounted to the three phalanxes on each of the four fingers, and one IMU each is mounted to the distal and intermediate phalanges of the thumb. Each IMU contains a 12-bit triaxial accelerometer, a 16-bit triaxial gyroscope, and a triaxial geomagnetometer. Sensor fusion is performed via a proprietary algorithm on a 32-bit microcontroller, yielding a global-frame orientation quaternion for each phalanx of hand.

The BNO055 footprint is 5×4.5 mm^2 and is mounted on a customized 6.35×6.35 mm^2 breakout PCB, making it easier to attach to the glove fabric with minimum constraints on the user's natural hand motion. These sensors are networked over a pair of I^{2}C buses in star



(a) Velostat sensor construction

(b) Velostat sensor circuit

Figure 2.3: Velostat force sensor construction and circuit layout

configuration, each of which is multiplexed using one TCA9548A I^{2}C multiplexer. Each of these two multiplexers is connected to one of two I^{2}C bus interfaces available on a single Raspberry Pi 2 Model B, which acts as the master controller for the entire glove system. We base the layout for our pose-sensing pipeline largely on work by Kortier *et al.* [KSR14], whose experiments quantify the characteristics of such an arrangement.

Physical connections use a high-flexibility, silicone-coated 29-gauge stranded-core wire. The IMUs are fixed with neutral cure silicone rubber into small 3D-printed housings, which are sewn into the glove's Lycra fabric over the top of their corresponding phalanxes.

Force sensing pipeline The force sensing pipeline uses a network of force sensors deploying Velostat. Figure 2.3a shows the multi-layer structure of this sensor. A single-point-sensing version of these sensor is constructed by layering small strips of Velostat (2×2 cm^2) between two outer shells of conductive fabric with conductive thread stitched into it. Lead wires to the pad are braided into the conductive thread fibers. The braided wire is then soldered to itself to form loops that hold the braid in place.

Time division of the channels is done for the palm grid via a pair of 74HC4051 analog multiplexers, and for the pads on the fingers via a single CD74HC4067 analog multiplexer. The multiplexers are controlled via the Raspberry Pi 2's GPIO, and their values are read into the Raspberry Pi via an SPI-enabled ADS1256 ADC at 40 whole-hand sps.

Force sensor characterization In order to characterize the force-voltage relation of the sensor, an experiment is conducted using a similar setup to that mentioned in [LS15]. Weights are applied to a 2×2 cm^2 Velostat sensing taxel ranging in value from 0.1 kg to 1.0 kg in 0.1 kg increments, and additionally at values of 1.2, 1.5, and 2.0 kg. All Velostat sensors utilized in prototyping are made of the same 2×2 cm^2 size taxel to ensure a single force-voltage relation can be applied. The calibration circuit is the same as Figure 2.3b except that only the Velostat taxel of interest is connected. A voltage divider to allow tuning of the taxel's sensing range was proposed by Lee *et al.* [LS15], in which the force-voltage relation follows a power law with different coefficients, yielding the force voltage relation $F = -1.067V^{-0.4798} + 3.244$ with $R^2 = 0.9704$, where F is the applied force in terms of weight and V is the output voltage. In this work, however, we approximate the force-voltage relation with a logarithmic law instead due to its better R^2 value under our experimental setup, which yields the relation $F = 0.569 \log(44.98V)$ with $R^2 = 0.9902$. The comparisons between power law and logmarithmic law are shown in Figure 2.4.

Parameter	Value
BNO055 IMU	N = 15
Sampling Frequency	20 [Hz]
Velostat sensor	N = 26
Sampling Frequency	40 [Hz]
Raspberry Pi 2	N = 1
Quad-core CPU	900 [MHz]
RAM	1 [GB]

Table 2.1: Prototyping hardware parameters

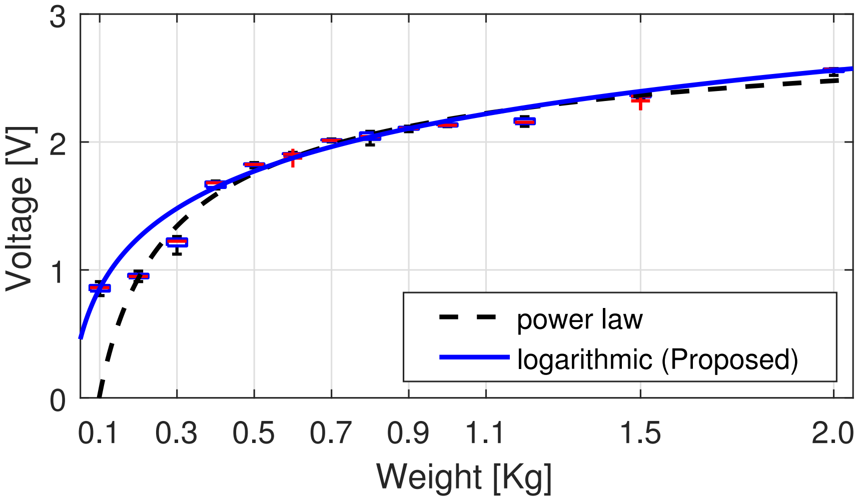


Figure 2.4: Force-voltage relation of one constructed Velostat sensing unit. A logarithmic law fit performs better than a power law fit.

2.1.2.3 Prototyping

Figure 2.1 shows a prototype of the proposed design and Table 2.1 lists the equipment we utilized in the prototype and their parameters. The force sensing functionality is achieved by deploying five 2 × 1 customized Velostat force sensors on each finger / thumb that each detects pressure in two regions (proximal and distal), and a single 4 × 4 sensor spreads over the glove's palm. The sensors placements and sensing regions are shown in Figure 2.1b. By constructing a voltage divider circuit as shown in Figure 2.3b, where multiple Velostat sensors are connected in parallel via a multiplexer that accesses a single sensor at a time. The Analog-to-Digital converter (ADC) extended from the Raspberry Pi integrated with a 200Ω resistor serves as the voltage divider. The resistance of the corresponding cell can be measured to capture the force in that region. This arrangement enables the capability of measuring the force distributed on the hand.

The 15 IMUs on each phalanx and the palm (Figure 2.1a) provide pose sensing. These IMUs are connected to the Raspberry Pi 2, a single-board computer that is well suited for wearable devices, via proper multiplexers. With the merit of remote accessing in Raspberry

Pi and ROS, one can access the processed data in Raspberry Pi remotely and visualize in workstations.

Compared to the existing expensive commercially available glove-based systems, which are only capable of transmitting raw data collected by the sensors to a workstation, our proposed design can enable on-board processing (see subsection 2.1.3) of the captured information.

2.1.2.4 System Power Analysis

In order to make the entire glove-based system more portable, including the processing unit, we investigate the power consumption of the major components and the system as a whole. The power is calculated by the product of the voltage and current across the components of interest. The results reported in Table 2.2 are the peak values over 10 minutes of continuous operation. The proposed system has the merit of low power consumption by having a peak of 2.72W in total. Thus, a normal cellphone power bank (5V output, 3.5Ah, and 75g) could power the system for a reasonable amount of operation time. The proposed system can be operated in a fully wireless manner after adding a wireless adapter, improving user's mobility.

Component	Power (W)
IMU (×15) Network with MUX	0.60
Velostat Sensor with MUX	0.02
Raspberry Pi with ADC	2.15
Total	2.72

Table 2.2: Power consumption of the system

2.1.3 Software Implementation

In the subsequent subsections, we introduce three core software implementations: i) hand pose calculation, ii) force vector derivation, and iii) manipulative action visualization of both hand pose and hand-object interaction forces. In an effort to maximize compatibility with different usages, the software, including processing and visualization, is built on top of the ROS environment.

2.1.3.1 Hand Pose Reconstruction using Forward Kinematics

Hand forward kinematics The human hand has approximately 20 degrees-of-freedom (DoF): 2 DoF for metacarpophalangeal (MCP) joints, 1 DoF for proximal interphalangeal (PIP) joints, and 1 DoF for distal interphalangeal (DIP) joints. Using such structure, each finger can be modeled as a 4 DoF kinematic chain where the palm is the base frame and the distal phalanx is the end-effector frame. For simplicity, we model the thumb as a 3 DoF kinematic chain consisting nominally of its interphalangeal and carpometacarpal joints.

Given the rotations measured by two consecutive IMUs, joint angles are obtained and the position and orientation of each phalanx can be computed by forward-kinematics. Figure 2.5 shows the frame attachment and the kinematic chain of the index finger as an example. The palm is assigned as Frame 1, the proximal, middle, and distal phalanx are Frame 2 to Frame 4, respectively. l_1, l_2, and l_3 denote the length for proximal, middle, and distal phalanx, respectively. β and θ_1 denote the abduction/adduction and flexion/extension angles of the MCP joint while θ_2 and θ_3 denote the flexion/extension angles of the PIP and DIP joints. d_x and d_y are the offset between palm's center to the MCP joint in the x and y directions. Given these notations, the standard Denavit-Hartenberg (D-H) parameters are derived for each reference frame and tabulated in Table 2.3. A general homogeneous transformation matrix T from frame $i-1$ to i is

$$
{}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.1}
$$

where $c\theta_i$ denotes $\cos(\theta_i)$ and $s\theta_i$ denotes $\sin(\theta_i)$.

The pose of each phalanx in Cartesian space can be expressed in the palm reference frame by concatenating the homogeneous transformation matrix as shown in Table 2.4.

Joint limits A commonly used closed form representation of the finger joints motion constraints [LWH00] is adapted.

$$
\begin{aligned}
&0^\circ \leqslant \theta_1 \leqslant 90^\circ \\
&0^\circ \leqslant \theta_2 \leqslant 110^\circ \\
&0^\circ \leqslant \theta_3 \leqslant 90^\circ
\end{aligned} \tag{2.2}
$$

$$
-15^\circ \leqslant \beta \leqslant 15^\circ \tag{2.3}
$$

The imposed joint limits define the upper and lower bounds of the joint motions and, thus, eliminate unnatural hand gestures due to sensor noise.

The forward kinematics models also keep track of the potential rotational offset between each fabric-mounted sensor and the underlying bone (skin-motion artifact), which account for two sources of error: i) the process of mounting and sewing the IMUs into the fabric of the glove introduces inconsistencies in the alignment of the sensors with respect to the actual phalanxes, and ii) anatomical differences between users result in IMU mounts naturally falling into places in different configurations dependent on the anthropometry of the user's hand.

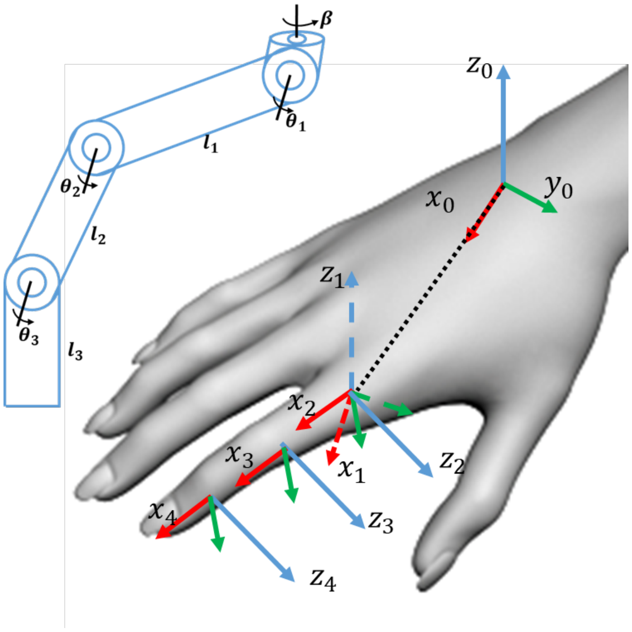


Figure 2.5: Frame attachment and the kinematic chain of the index finger, as an example

Pose calibration A compensatory calibration routine is performed to further eliminate the aforementioned inconsistencies. First, the hand is held flat on a table in a canonical pose. A glove-local reference frame is defined with x-y in the plane of the table and the x-axis parallel to the user's middle finger. The orientation of the single IMU on the palm is measured by hand with respect to this glove-local frame, $q_{\text{glove}\rightarrow\text{sensor}_{\text{palm}}} \in \mathbb{H}$. Then, a calibration event signal is called, triggering the forward kinematics code to update via direct measurement its internal representation of the rotation $q_{\text{sensor}_{\text{palm}}\rightarrow\text{sensor}_i} \in \mathbb{H}$ between the sensor on the palm and each of the remaining 14 sensors. Since the rotation $q_{\text{glove}\rightarrow\text{sensor}_{\text{palm}}}$ is already measured, it becomes trivial to compute the rotational errors $q_{\text{glove}\rightarrow\text{sensor}_i}$, which can then be cancelled out of the measured orientations.

2.1.3.2 Force Vector Derivation

We further combine force scalar data obtained from the force sensors with our estimated hand pose into the form of force vectors, enabling heterogeneous forces and poses in manipulative actions to share a shared representation. Specifically, each force vector is defined with the magnitude equal to the force reading from the corresponding force sensor. Vector direction is then set to be perpendicular to the finger phalanx that encoded the pose information. Due to the construction of our force sensors, the force reading obtained measures only the pressure but not the stress component of the surface force over the sensing fabric. In general, the force vector to one frame could be expressed as follow:

$$(F_{X_{ref}}, F_{Y_{ref}}, F_{Z_{ref}})^T, \tag{2.4}$$

where the *ref* denotes the frame we are referring to.

By applying the chain homogeneous transformations, we could derive the force vector

Link ID	α_{i-1}	a_{i-1}	θ_i	d_i
1	0	0	β	0
2	$\pi/2$	l_1	θ_1	0
3	0	l_2	θ_2	0
4	0	l_3	θ_3	0

Table 2.3: General standard Denavit-Hartenberg parameters of a finger

Phalanx	Transformation
Proximal	${}^0_1T\,{}^1_2T$
Middle/Distal for thumb	${}^0_1T\,{}^1_2T\,{}^2_3T$
Distal	${}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T$

Table 2.4: Concatenation of transformation matrices

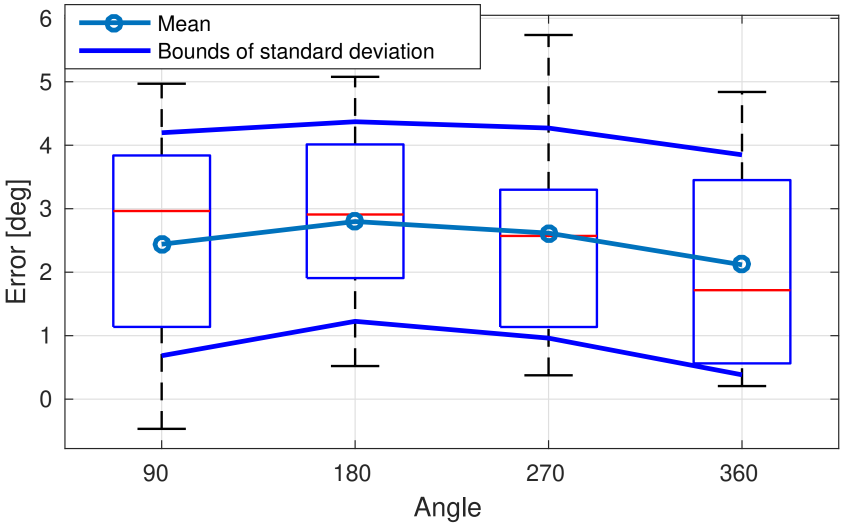


Figure 2.6: Bias and standard deviation of an individual IMU with up to 360° rotation. Red horizontal lines indicate median error, and the bottom and top edges of the blue boxes indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers.

with respect to any hand frame:

$$V = \prod_{i=1}^{n} {}_{i}^{i-1}T \cdot V_0, \tag{2.5}$$

where V and V_0 are homogeneous representation of 3-d vectors. In practice, we generate force vectors on each phalanx regarding wrist frame.

2.1.3.3 Visualization

To visualize the reconstructed hand motion, we create a hand model in ROS Unified Robot Description Format (URDF). In this model, we define the structure and connected joints of the human hand, as well as parameters such as the lengths of each phalanx and dimensions of the palm, which are measured in advance.

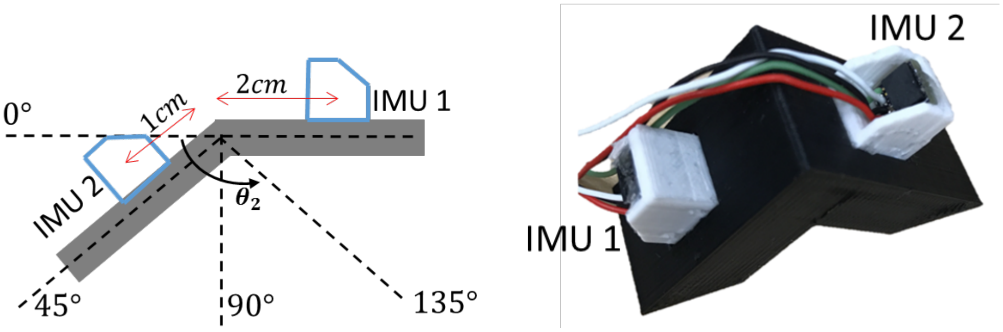
The orientation of each joint, as calculated in the forward kinematics, is then assigned to each linkage of the hand model to visualize the hand pose. We further create a set of force markers to indicate both the magnitude and direction of the forces being exerted by the palm and fingers over each of the 26 force sensing taxels, providing a visualization of the distribution of forces over the palm. Each finger proximal and distal link corresponds with one force marker while the palm link includes a 16-marker array which accords with the distribution of sensor grids on palm. The color of the marker remains green if the sensor is inactive, and turns red if force is applied.

2.1.4 Performance Evaluation

We evaluate the performance of individual components as well as of the system as a whole. Three experiments are conducted. The bias and variance of an individual IMU are firstly obtained. We further examine the accuracy of reconstructing a static angle with two articulated IMUs, indicating the performance of basic element using the pose sensing approach in the actual setup. The Velostat force sensor is evaluated by the quality of the force response in grasping a bottle having different weights. Captured pose and force information are also jointly evaluated via force vector visualization. Lastly, we perform the tasks of opening three types of medicine bottles that require different sets of manipulative actions.

2.1.4.1 IMU Evaluation

Single IMU evaluation As the reliability of the pose sensing critically relies upon the IMU performance, it is crucial to take the IMU's bias and variance into account, thus an experiment is conducted to model those quantities. An IMU is rotated driven by a precise stepper motor controlled by an Arduino microcontroller at a constant angular velocity of 60 RPM. Four rotation angles, 90°, 180°, 270°, and 360° are executed twenty times each. No rotating angles of greater than 360° are necessary as it is far beyond fingers' rotation limits. Figure 2.6 illustrates the mean and the standard deviation of the error of such rotating angles. The IMU displays consistent error characteristics, that is having a bias of 2° to 3° with a standard deviation of ±1.7°, with small variations for all 4 rotation angles. Such



(a) Schematic of the setup

(b) Actual exemplary setup

Figure 2.7: Experimental setup for evaluating the angle reconstruction with two articulated IMUs

results indicate that the selected IMU is generally reliably within the applications of the proposed design.

Articulated IMU reconstruction of fixed angles Based on the data of two adjacent IMUs that span a joint of interest, assuming revolute joint, we test the accuracy of the estimated joint angle. Four rigid bends with fixed angles of 0° 45°, 90°, and 135° are manufactured to simulate an rotation angle of a revolute joint. These four angles are selected since they evenly divide the reachable area of a finger joint with small exceeding based on Equation 2.2. The experimental schematic is shown in Figure 2.7a. Figure 2.7b is an exemplary setup using 90° joint angle: IMU 1 is placed 2cm away behind the bend, simulating the IMU attached to proximal phalanx, while IMU 2 is placed 1cm ahead of the bend, corresponding to the IMU on the middle phalanx. The IMU placement is identical to that in the prototype glove. For each bending angle, the test is repeated twenty times, and the joint angle estimates are shown in Figure 2.8. As bending angles increase, the reconstructed angle errors increase from 4° to approximately 6° while the confidence intervals increase. We can see that articulated IMUs under-perform as the rotated angle increases, but the error range is still reasonable and the designed IMUs configuration can reliably fulfill the task.

2.1.4.2 Grasping Bottles

After establishing the force-voltage relation of the proposed Velostat force sensors, we evaluate the performance of the entire force sensor network as a whole by performing grasping action. The reason that we choose grasping is becuase it is one of the most common actions in manipulation. An experiment in grasping an *empty*, *half-full*, and *full* water bottle, whose weight is 0.13kg, 0.46kg, and 0.75kg, respectively, is conducted to demonstrate the capability in differentiating low, medium, and high grasping forces.

The grasping hand pose is shown in the Figure 2.10a. The pose is natural and no artificial force is applied other than the force just sufficient to grasp and hold the bottle stably. For each bottle condition, ten grasps are performed. The force in the palm is treated as the average of the sixteen force readings from the 4-by-4 force sensor on the palm to simplify the analysis. Similarly, the force in each finger is the average of the 2-by-1 force sensor. More careful inspections of force exerted in grasping can also be proceeded by analyzing the response of every force sensing unit. The results shown in Figure 2.9 indicate the force increments is correlated with the weight increments of the bottle.

Figure 2.10b shows the visualization of the grasping pose and the force vector, which reliably captures the actual manipulative action. This experiment qualitatively indicates the sensitivity and reliability of the force sensing using proposed Velostat force sensors.

2.1.4.3 Capturing Fine Manipulative Actions

Using the prototype system, a series of manipulative actions in opening three types of medicine bottles are studied. Each bottle equips different lock mechanisms and requires particular actions for removing the bottle lid. *Bottle 1* has no safety lock and can be opened by simply twisting the lid. *Bottle 2* requires the lid to be simultaneously pressed down and twisted to open. *Bottle 3* requires pinching the lid's safety lock in order to open it. For *Bottle 2* and *Bottle 3*, some of the actions in the sequence (*i.e.* pressing and pinching)

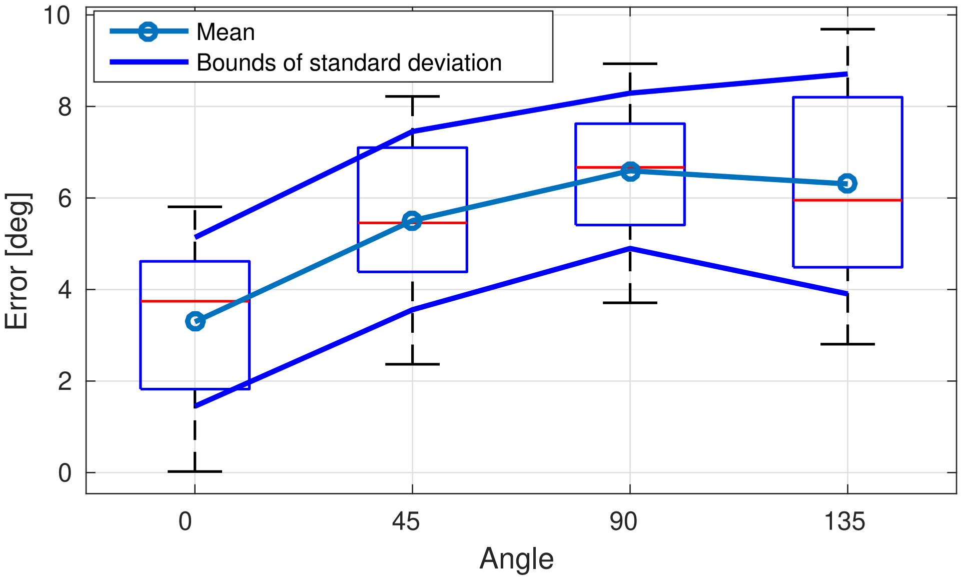


Figure 2.8: Mean and standard deviation of the reconstructed angles using articulated IMUs under different angles including the boxplot of the collected data. Red horizontal lines indicate median error, and the bottom and top edges of the blue boxes indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers.

are hard to perceive without recovering the force exerted by the hand. In the first row of Figure 2.11a,Figure 2.11b,Figure 2.11c, we visualize the manipulative action sequences cap-

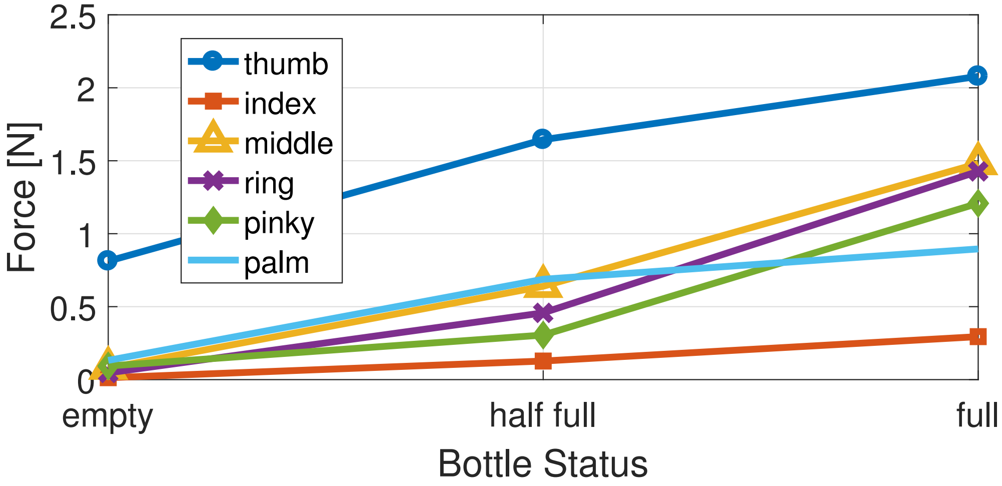
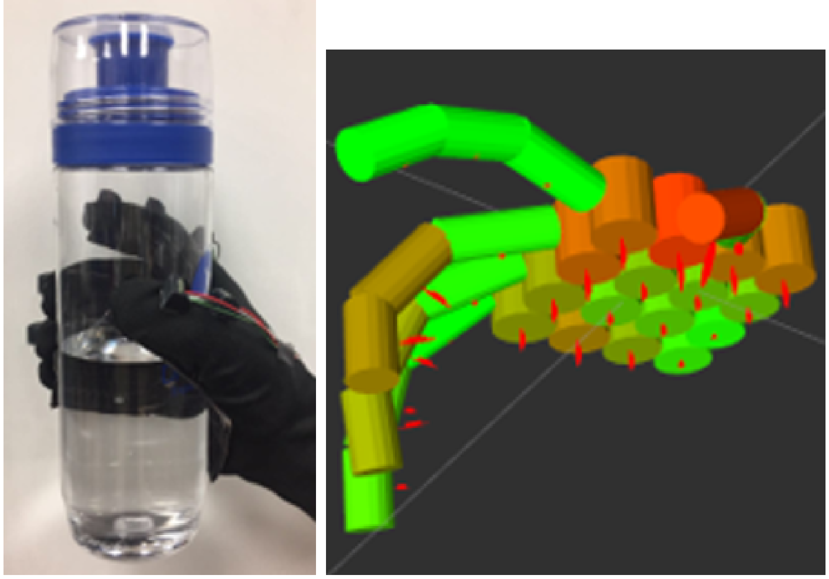


Figure 2.9: Force response of grasping empty, half-full, and full bottles, respectively.



(a) Actual grasping pose (b) Force vector

Figure 2.10: Grasp of half-full bottle and captured pose and force vector

tured for opening *Bottle 1*, *Bottle 2*, and *Bottle 3*, respectively. The second row of each corresponding figure illustrates the actual action sequences captured by a RGB camera.

Visualization In contrast to the action sequences captured by a RGB camera, the visualization results provide additional force information regarding the different fine manipulative actions involved in the opening the bottles. For instance, the fingers in Figure 2.11b are flat and parallel to the bottle lid, while the one in Figure 2.11c is similar to gripping pose. The responses of force markers are also different due to varying contact points between the human hand and the lid: high responses in Figure 2.11b are concentrated on palm area, while in Figure 2.11c, there are only two evident responses on distal thumb and index finger. Without taking force responses into account, the actions sequences of opening *Bottle 1* and *Bottle 3* are very similar to each other (see Figure 2.11a and Figure 2.11c).

The capability of detecting the visually unobservable forces has been shown as one of the advantages of the proposed design in studying the fine manipulative actions. By analyzing the spatio-temporal signals of force and pose in terms of joint angles, we can also evaluate the performance of the design as a whole.

Data interpretation Due to the distinct safety lock mechanisms equipped, the manipulative actions required for opening these three types of medicine bottles are different as shown in Fig. 2.11. The proposed glove-based system successfully captures the differences, and Figure 2.12 illustrates the force collected at one taxel on palm and at the fingertip of thumb, as well as the flexion angle of the MCP joint of the index finger.

As opening the *Bottle 2* requires pressing the lid, the proposed system captures high force response on the palm area. In contrast, the other two bottles yields very low force response in the same region. If we look at the force exerted at the fingertip of the thumb, opening *Bottle 3* with pinch-to-open lock has larger force in magnitude and longer duration compared to opening *Bottle 1* as it comes with no safety lock and it only involves twisting the lid with mild force. The thumb does not contact with the lid in opening *Bottle 2*, yielding no force response.

For joint angle measurements, since opening both *Bottle 1* and *Bottle 3* involve similar twist action, the measured flexion angles of the MCP joint in the index finger are around 50° in both cases. In opening *Bottle 2*, only the palm touches the lid and the fingers remain stretched, resulting in small flexion angle.

2.1.5 Conclusion and Future Work

We present a design of a glove-based system, capable of simultaneously collecting human hand pose and exerted contact force data during hand-object interaction with fine manipulative actions. The overall system design is firstly illustrated, following by the hardware implementations. In software implementation, we have defined the kinematic chain of a hand, in order to reconstruct the hand pose. Using custom Velostat force sensor taxels, we are able to measure the hand-object interaction forces across large regions of the hand. In the visualization framework, the simulated hand model successfully reflects subtle differences in grasping action sequences when interacting with three different types of bottles with various safety locks. By quantitatively analyzing the collected spatio-temporal signals of force and pose, we show the potentials using the proposed glove, as well as some preliminary analysis for studying hand-object dynamics. A direct application using the proposed system is to enable robot to learn and perform finer manipulative actions through human demonstrations [EGX17]. Recent study also shows the haptic feedback is crucial for recognizing interactions [STC16], indicating potential applications in social interactions [SGR17, SRZ16].

In future, a potential direction would be improving the current kinematic modeling of the thumb to better reflect the actual structure and the DoF of the thumb. Some industrial manufacturing methods, such as laser cutting, could be introduced in fabricating and assembling the Velostat force sensor to achieve a more consistent performance.

2.2 Unsupervised Learning of Hand-Object Interaction

Contact forces of the hand are visually unobservable, but play a crucial role in understanding hand-object interactions. In this section, we propose an unsupervised learning approach for manipulation event segmentation and manipulation event parsing. The proposed framework incorporates hand pose kinematics and contact forces using a low-cost easy-to-replicate tactile glove. We use a temporal grammar model to capture the hierarchical structure of events, integrating extracted force vectors from the raw sensory input of poses and forces. The temporal grammar is represented as a temporal And-Or graph (T-AOG), which can be induced in an unsupervised manner. We obtain the event labeling sequences by measuring the similarity between segments using the Dynamic Time Alignment Kernel (DTAK). Experimental results show that our method achieves high accuracy in manipulation event segmentation, recognition and parsing by utilizing both pose and force data.

2.2.1 Introduction

Consider a complex manipulation event of a person opening a medicine bottle with safety lock (Figure 2.13). During this process, a number of movement primitives were performed: *grasp*, *push-and-twist*, *push-and-twist*, *twist*, and finally *pull* the lid off the bottle. Even with the most state-of-the-art action understanding and recognition algorithms (see survey [Pop10, WRB11]), it is still challenging to segment such action sequence and parse the manipulation event. This is due to three major difficulties: i) severe occlusions happen during fine manipulation, especially self-occlusions, ii) in subtle manipulation tasks, visual data may not be able to reveal adequate knowledge to capture the quintessence. Certain actions are hard to detect using skeleton data alone but need additional force readings *e.g.*, whether an action of pushing was performed during twisting the lid, and iii) ground truth data is difficult to obtain using vision sensor alone, oftentimes impossible to obtain the needed information (*e.g.*, the force readings, and accurate finger poses during occlusions).

In this section, we present an unsupervised learning method for manipulation event segmentation, recognition and parsing. The method not only accounts for the aforementioned challenges, but also captures the temporal hierarchical structure of the manipulation sequence using a grammar model—a temporal And-Or graph (T-AOG). Specifically, we investigate the manipulation actions of opening different types of medicine bottles. Some examples are shown in Figure 2.16a. *Bottle 1* has no safety lock and can be opened by simply twisting the lid. *Bottle 2* requires pressing the lid while twisting. Pinching the safety lock is needed to open *Bottle 3*. Importantly, some actions (*e.g.*, pressing, pinching) are difficult to observe visually, thus require additional sensing for action recognition.

To obtain the force readings during manipulations, we propose to study hand-object interactions with additional force information through a low-cost, easy-to-replicate tactile glove [LXM17]. Although some efforts have been shown to recover the forces during interactions using vision-based methods [ZZM13, WMZ13, ZZZ15, PKQ15, ZJZ16], it remains an open problem without adopting a hardware-based solution. Using a tactile glove can reliably retrieve contact forces to overcome the limitation of using visual data alone.

By observing the data collected using the tactile glove, such as the force exerted on the palm, we can learn that a push-down action is performed as well as a set of motion primitives that can best describe the action sequences. Thus, our system is able to “see”, in numerical terms, the forces during hand-object interactions. We argue that this is an important step in recognizing manipulation actions with visually latent force information.

Still, it is nearly impossible to understand and transfer the raw data (poses and forces) retrieved from the tactile glove *directly* to a robot due to different embodiments. Therefore, we need to reconstruct the semantic meanings of manipulation events from the human demonstration, allowing the transfer of abstract knowledge to a robot.

To recover the semantic meaning and model the temporal structure of actions in a hand-object interaction, we represent the manipulation sequence using a T-AOG, a temporal grammar model that captures the hierarchical structure of the action sequences. Its terminal nodes are motion primitives, *e.g.*, twisting and pressing, which is learned by unsupervised clustering over extracted features of the pose and force sensory inputs. To evaluate the effectiveness of our model, we compare the segmentation and labeling results of different sensory data with several baseline methods.

2.2.1.1 Related Work

Action Recognition A number of approaches have been proposed for action recognition in various applications. This literature is too wide to survey here; we refer readers to two recent surveys for recognizing and parsing human actions [Pop10, WRB11]. Recently, due to additional sensory input, RGB-D sensors such as Kinect are capable of estimating 3D poses from a single image [SSK13]. Further studies have demonstrated impressive results of pose estimation and action recognition from RGB-D videos [WZZ13b, ZTH12, WLW13, WNX14, WZZ16, QHW17]. These works, however, focuses on body-size action recognition without force sensing. In contrast, the presented work addresses the hand-size finer-grained manipulation actions with reconstructed forces.

Vision-based Force Estimation Brubaker *et al.* estimated contact forces and internal joint torques using a mass-spring system [BF08, BSF09, BFH10]. More recently, Zhu *et al.* [ZZZ15] and Pham *et al.* [PKQ15] proposed to use numerical differentiation methods to estimate hand-object interactions during manipulation tasks. In computer graphics, sophisticated physics-based soft-body simulation can calculate contact force from video [WMZ13, ZZM13]. These work, however, requires prior knowledge of geometry and physical properties of the manipulated objects. By using a tactile glove, estimating forces in the present study does not rely on such assumptions.

Learning from Demonstration (LfD) A robot must recognize and understand the actions sufficiently in order to imitate the tasks from the demonstrations. LfD (also imitation learning, learning by watching, or apprenticeship learning) is too expansive to survey here; we refer readers to a survey [ACV09]. In the last few years, with the recent rise of Convolutions Neural Networks, there are increasing interests in providing and parsing demonstrations using pure visual data [BRM12] by learning action plans [YLF15] and physical interactions [PGH16] in complex and higher-level tasks. However, it is yet still difficult to convey force information from vision-based methods reliably.

Kinesthetic Teaching and Teleoperation To address the above issue, the robotics community has been developing kinesthetic teaching or teleoperation approaches to recognize low-level motion primitives during hand-object interactions. These approaches are capable of transferring certain rich physical information such as force knowledge to robots. Manschitz *et al.* [MKG14a] presented a method to teach robots to unscrew a light bulb by moving primitives, which are represented by sequences of graphs. A more recent work was presented in [MGK16]. Chebotar *et al.* [CKP14] used spectral clustering and PCA to reduce the dimensionality in learning tactile feedback during performing scraping task. More challenging hand-object interaction tasks involving the manipulation of deformable objects were discussed using a similar approach [LLG15]. Learning impedance behaviors and trajectory following skills was presented in [RCC16] by combining robot's dynamical system and stiffness estimation.

2.2.1.2 Contributions

This work makes three contributions:

1. We incorporate *invisible* force in addition to the conventional pose-based methods for event segmentation and parsing during fine-grained manipulation tasks. We show in the experiment that a better performance of motion recognition is achieved by jointly considering hand pose and force data.

2. We propose an unsupervised learning framework to learn a temporal grammar model (T-AOG) for hand-object interactions. The framework incorporates automatic clustering, segmentation, labeling, and high-level grammar induction. The grammar structure is shown to significantly improve the action recognition results compared to using clustering method alone.

3. We introduce a general method for modeling noisy and heterogeneous sensory data of hand-object manipulation.

2.2.1.3 Overview

The remainder of this section is organized as follows. In subsection 2.2.2, we introduce the representation T-AOG. In subsection 2.2.3, we present the learning algorithm consisting of hierarchical clustering and grammar induction. The inference algorithm of motion recognition is introduced in subsection 2.2.5. subsection 2.2.6, we demonstrate the data with additional force sensing indeed outperforms the data with either pose or force data only. Furthermore, our analysis shows the parsing results using T-AOG help improve the performance significantly compared with using clustering only.

2.2.2 Representation

We introduce a structural grammar model *Temporal And-Or Graph (T-AOG)* [ZM07] to represent the temporal structure of a task. An AOG is a directed graph which describes a stochastic context free grammar (SCFG), providing a hierarchical and compositional representation. Formally, the AOG is defined as a five-tuple $G = (S, V, R, P, \Sigma)$, where S is a start symbol; V is a set of nodes which includes the non-terminal nodes V^{NT} and terminal nodes V^T: $V = V^{NT} \cup V^T$; $R = \{r : \alpha \to \beta\}$ is a set of production rules that represent the top-down sampling process from a parent node α to its child nodes β; $P : p(r) = p(\beta|\alpha)$ is the probability for each production rule; Σ is the language defined by the grammar, *i.e.*, the set of all valid sentences given the grammar.

In an AOG, the **non-terminal** nodes can be divided into two types: $V^{NT} = V^{AND} \cup V^{OR}$. An **And-node** is used to represent the compositional relations. A node v is an And-node if the entity represented by v can be decomposed into multiple parts, which are represented by its child nodes. An **Or-node** is used to represent alternative configurations. A node v is an Or-node if the entity represented by v has multiple mutually exclusive configurations represented by its child nodes. The **terminal** nodes represent the entities that are not further decomposed or have different configurations. A **parse graph** pg is an instance of the AOG, where the And-nodes are decomposed and one of the child nodes is selected for the Or-nodes.

In particular, a T-AOG represents a set of all possible sequences to execute a certain task. The start node S represents an event category (*e.g.*, opening a bottle). The terminal nodes V^T represents the set of motion primitives that a human or a robot can perform (*e.g.*, approaching, twisting). An And-node is decomposed into sub-events or motion primitives as its child nodes. An Or-node encodes alternative solutions to perform a sub-task. A *pg* for an event is a sub-graph of T-AOG that captures the temporal structure of the scenario.

As shown in Figure 2.15, features are extracted from the raw input sensory data and further segmented for semantic parsing. Pose and force features Γ are extracted based on a raw sensory input sequence I in time interval $[1, T]$. Each frame is labeled with motion primitive a_t. Aggregating together, we obtain a label sequence $A = \{a_t\}$. The segmentation of the sensory input sequence is defined as $\mathcal{T} = \{\gamma_k\}, k = 1, \cdots, K$, where $\gamma_k = [t_k^1, t_k^2]$ represents a time interval in which the motion primitive remains the same. Later in this section, we use a_{γ_k} to denote the motion label for the segment I_{γ_k}.

2.2.3 Learning of Hand-Object Interactions

The unsupervised learning pipeline is illustrated in Figure 2.14. Given training sequences of raw sensory input of poses and forces, our goal of learning is to unsupervisedly learn i) the motion primitives in the sequences of hand-object interactions, ii) the event segmentation in every sequence, and iii) the high-level grammar structure (T-AOG) that captures every observed sequences of the hand-object interactions.

2.2.3.1 Unsupervised Learning of Motion Primitives

To recognize motion primitives of hand-object interactions, we adopt the agglomerative hierarchical clustering, capable of successively merging the similar features from the low-level features, without knowing the exact number of clusters in advance. The Ward's agglomerative method is used to determine whether a merge is needed in each iteration:

$$
\begin{aligned}
\triangle(A,B) = & \sum_{i \in A \cup B} ||\vec{x}_i - \vec{m}_{A \cup B}||^2 - \sum_{i \in A} ||\vec{x}_i - \vec{m}_A||^2 \\
& - \sum_{i \in B} ||\vec{x}_i - \vec{m}_B||^2 \\
= & \frac{n_A n_B}{n_A + n_B} ||\vec{m}_A - \vec{m}_B||^2,
\end{aligned}
\tag{2.6}
$$

where A, B denote two clusters in the current iteration, m_A, m_B are the cluster centers, and $\triangle(A,B)$ is the cost of merging clusters A and B.

By default, the hierarchical clustering always groups data points using spatial distance alone, without considering the temporal consistency. This becomes an issue when dealing with manipulation data, which naturally comes with temporal constraints. To alleviate this issue, we apply the Aligned Cluster Analysis (ACA) to reduce the noisiness based on Dynamic Time Alignment Kernel (DTAK) [ZTH12], resulting in a refined segmentation. The ACA is an extension of kernel k-means clustering that could be solved as a versatile energy minimization problem using coordinate descent algorithm:

$$
s^* = \arg\min_s \mathbf{J}(\mathbf{G}, s) = \sum_{c=1}^{k} \sum_{i=1}^{m} g_{ci} D_c(\mathbf{X}_{[s_i, s_{i+1})}), \tag{2.7}
$$

where $\mathbf{G}_{k \times n}^T \mathbf{1}_k = \mathbf{1}_n$ is the indicator matrix, $g_{ci} = 1$ if sample $\mathbf{X}_i$ belongs to cluster c, and D_c measures the kernel distance between sample point and cluster center. In practice, Equation 2.7 could be solved in a dynamic programming manner, which leverages the relationship between G and s by solving the Bellman's equation [ZTH12]:

$$
\mathbf{J}(v) = \min_{v - n_{\max} < i \leqslant v} (\mathbf{J}(i-1) + \min_g \sum_{c=1}^{k} g_c D_\psi^2(\mathbf{X}_{[i,v]}, \dot{\mathbf{z}}_c)), \tag{2.8}
$$

where $D_\psi^2(\mathbf{X}_{[i,v]}, \dot{\mathbf{z}_c})$ is the squared kernel distance between segment $\mathbf{X}_{i,v}$ and class center c, and $n_{\max}$ defines the maximum segment length of clustering.

2.2.4 Event Segmentation

The semantic label of each segment is required to learn a high-level temporal grammar based on the segmented sequences. Although event segmentation of a single segmented motion sequence is straightforward by following its clustering label, it is still difficult to extract the semantic meaning of one segment when having multiple segmented motion sequences performing the same task.

Considering two segmented sequences $\mathbf{X}_{[S_1,S_2 \dots S_n]}$ and $\mathbf{Y}_{[S_1,S_2 \dots S_m]}$, we assign semantic labels by merging those segments into clusters where each cluster contains segments that are 'close' in distance. Specifically, we adopt the DTAK [ZTH08] criterion $\mathcal{D}(\mathbf{X}_{S_i}, \mathbf{Y}_{S_j})$ to estimate the similarity of segments across different trials of motion primitives segmentation:

$$\mathcal{D}(\mathbf{X}_{S_i}, \mathbf{Y}_{S_j}) = \tau_{\left[\mathbf{X}_{S_i}, \mathbf{Y}_{S_j}\right]}, \tag{2.9}$$

where $\mathbf{X}_{S_i}, \mathbf{Y}_{S_j}$ are candidate segments that may be grouped together, $\tau_{[\mathbf{X}_{S_i}, \mathbf{Y}_{S_j}]}$ is the similarity metric between two segments calculated recursively using DTAK kernel matrix. Note that it could also be applied to the situation that $\mathbf{X}$ and $\mathbf{Y}$ are the same motion sequence that only differ in segment index i and j.

Based on the distance metric of DTAK, we further apply k-means algorithm to cluster those segments such that each cluster represents one semantic label. The semantic labels of each segmented motion sequence can therefore be obtained by cluster IDs of the corresponding segments.

2.2.4.1 Grammar Induction

After acquiring the semantic labels of multiple segmented motion sequences, we build a T-AOG grammar model using an unsupervised structural learning method [TPZ13]. We aim to learn a grammar from a set of sequence of instances that maximize the posterior probability.

An initial grammar is built in which the root node is an Or-node, and each branch is an And-node that represents a sequence instance. This initial grammar leads to the maximal likelihood of the training data but has a very small prior probability because of its large size. Starting from the initial grammar, new intermediate non-terminal nodes are generated in a bottom-up fashion to increase its posterior probability. At each iteration, a grammar fragment rooted at a non-terminal node is added into the grammar. In practice, we find it is sufficient to use greedy search with random restarts to identify good grammar fragments.

2.2.5 Inference

Given a sequence of pose and force data Γ as an input, our goal is to find the best motion label sequence A^*, *i.e.*, find the optimal label sequence of the segments that best explains the observation given the learned grammar $\mathcal{G}$ by maximizing the posterior probability:

$$A^* = \arg\max_{A} p(A|\Gamma, \mathcal{G}) = \arg\max_{A} p(\Gamma|A)p(A|\mathcal{G}), \tag{2.10}$$

where $p(\Gamma|A)$ is the likelihood given the motion label sequence, and $p(A|\mathcal{G})$ is the parsing probability of the parse graph given the grammar. The first term is given by:

$$p(\Gamma|A) = \prod_{k=1}^{K} p(\Gamma_{\gamma_k}|a_{\gamma_k}) = \prod_{k=1}^{K} \prod_{t=t_k^1}^{t_k^2} p(\Gamma_t|a_{\gamma_k}), \tag{2.11}$$

where k is the segment index, γ_k is the kth segment as introduced in subsection 2.2.2. This term is given by a Gaussian distribution fitted to the learned clusters in the training examples.

The second term $p(A|\mathcal{G})$ in Equation 2.10 is the Viterbi parsing likelihood, *i.e.*, the probability of the best parse of the string terminals.

Since it is intractable to directly compute the optimal label sequence, we infer the approximately optimal $\widehat{A^*}$ in two steps: i) use the unsupervised clustering method to obtain the segmentation and initialized labels, and ii) refine the labels according to Equation 2.10 by Gibbs sampling with simulated annealing to find the labeling that maximizes the posterior probability.

2.2.5.1 Gibbs Sampling with Simulated Annealing

After initializing the labels by clustering, we find the best parse by Gibbs sampling with simulated annealing. Given an input sequence, we assign one segment label according to the posterior probability (Equation 2.10) at each iteration. Specifically,

$$a'_{\gamma_k} \sim p(\Gamma_{\gamma_k}|a_{\gamma_k})p(A'|\mathcal{G}), \tag{2.12}$$

where a'_{γ_k} is the new label of segment Γ_{γ_k}, and A' is the new label sequence obtained by changing the kth label to a'_{γ_k} in the current labeling sequence A. To find the parse with the maximum probability, we adopt simulated annealing to the sampling process by dividing the log probability by a temperature T. We decrease the temperature through the sampling process until the labeling sequence converges.

2.2.6 Experiments

2.2.6.1 Human Data Acquisition

Tactile Glove To capture both pose and force in hand-object interactions, we utilize an open-source tactile glove [LXM17]. The tactile glove employs a network of 15 IMUs to measure the rotations between individual phalanxes. Hand pose is reconstructed using forward kinematics. With 6 customized force sensors using Velostat, a piezoresistive material, the force exerted by hand is recorded in two regions (proximal and distal) on each phalange and a 4 × 4 regions on the palm. The data is collected and visualized using the Robot Operating System (ROS).

Experimental Setup We utilize a Vicon motion capture system to obtain the relative poses between the wrist of hand and object parts. Figure 2.16a describes the schematic of the experimental environment setup in human data acquisition. Six Vicon cameras are placed on top left and top right in front of the area of interests.

Force Vectors Force vectors are computed as the extracted features from the force and pose data (see Figure 2.16b). Each force scalar measured on hand is normalized and treated as the magnitude of the force vector. The orientation of the force vector is set to be perpendicular to the fingers. All the force vectors are expressed with respect to one fixed frame by applying the chain product of homogeneous transforms. Hence, we are able to combine the heterogeneous pose and force information into one compact form of feature vector.

2.2.6.2 Evaluation

The performance is evaluated by the frame-wise recognition accuracy, *i.e.*, comparing the predicted event label with the ground truth frame by frame. The ground truth segmentation is manually labeled. Based on this protocol, we evaluate the correspondence in three metrics: i) *Pose* feature as the Euler angles of each phalanx, ii) *Force* feature as the magnitude of the force, and iii) the combination of *Pose* and *Force* in the form of force vectors. For fair comparison, the results reported below use the cluster number $k = 5$ and maximum segment length $n_{max} = 200$.

2.2.6.3 Event Segmentation and Recognition with Clustering

Figure 2.17 visualizes the event recognition results by segmenting each motion primitive of the trials in opening *Bottle 1, 2,* and *3*. Quantitative results are shown in Table 2.5. The segmentation using only pose data has the worst performance compared with the ground truth. The use of force data shows a significant improvement compared to the pose only data.

This result indicates the benefits of the force information during hand-object manipulation. Combining both pose and force data together outperforms that only uses either pose or force data.

2.2.6.4 Segmentation, Recognition and Parsing with T-AOG

To further reduce noise, mislabeling, and incoherence, T-AOG is integrated to refine the segmentation, recognition and parsing of the motion sequences by maximizing the posterior probability.

Figure 2.18 shows the motion frames during the interactions in opening three types of bottles. The number in each frame denotes its motion label which is produced by the proposed clustering pipeline. Additionally, we highlight the changes after applying the proposed annealing inference framework, indicated by the red arrow, which reveals the directions of label merging.

Experimental results after integrating a T-AOG parsing are both qualitatively and quantitatively presented. As depicted in Figure 2.17, comparing to model-free clustering methods, the T-AOG based parsing approach recovers some noisy and mislabeled segments, resulting in more coherent results. Last column of Table 2.5 shows the quantitative results. The

	Clustering only			With T-AOG
	Pose only	Force only	Pose and Force	Pose and Force
Bottle 1	55.3%	67.5%	70.3%	**78.6%**
Bottle 2	62.0%	70.9%	76.2%	**82.5%**
Bottle 3	54.1%	71.1%	72.9%	**78.5%**

Table 2.5: Quantitative Evaluation. With clustering only, we use the hand pose, in the forms of Euler angles of each phalanx; hand force, as scalars; and the combination of pose and force as force vectors as feature inputs. Including force factor yields higher correspondence with ground truth sequence. Parsing the events with T-AOG on top of the clustering, the performance improves significantly.

performance of both segmentation and recognition using T-AOG have a marked improvement compared to the methods only by clustering, demonstrating the usefulness of learning a grammar model for events parsing and inference.

2.2.7 Conclusion and Future Work

In this work, we present an unsupervised approach for manipulation event segmentation, recognition and parsing. Hand-object interaction sequences are segmented in an unsupervised learning fashion, based on which a temporal grammar is further induced. Through a tactile glove, our work explicitly incorporates forces imposed by hands in addition to its pose.

The experiments demonstrate that force is indeed an important factor as it significantly improves motion primitives segmentation. In addition, learning a grammar model T-AOG from the clustering results for parsing the motions can reduce noisiness and eliminate mislabeling and ultimately lead to a more coherent event segmentation and parsing.

In the future, the proposed approach could be used to improve the traditional event segmentation, recognition and parsing in computer vision by inferring the force from the videos [ZJZ16]. It is also possible to use the segmentation as the demonstrations to teach robots with LfD to open medicine bottles [EGX17] or more complex tasks, *e.g.*, tool uses [ZZZ15].

2.3 Robot Imitation Learning of Hand-Object Interaction

Learning complex robot manipulation policies for real-world objects is challenging, often requiring significant tuning within controlled environments. In this work, we learn a manipulation model to execute tasks with multiple stages and variable structure, which typically are not suitable for most robot manipulation approaches. The model is learned from human demonstration using a tactile glove that measures both hand pose and contact forces.

The tactile glove enables observation of visually latent changes in the scene, specifically the forces imposed to unlock the child-safety mechanisms of medicine bottles. From these observations, we learn an action planner through both a top-down stochastic grammar model (And-Or graph) to represent the compositional nature of the task sequence and a bottom-up discriminative model from the observed poses and forces. These two terms are combined during planning to select the next optimal action. We present a method for transferring this human-specific knowledge onto a robot platform and demonstrate that the robot can perform successful manipulations of unseen objects with similar task structure.

2.3.1 Introduction

Consider the task of opening medicine bottles that have child-safety locking mechanisms (Figure 2.19a). These bottles require the user to push or squeeze in various places to unlock the cap. By design, attempts to open these bottles using a standard procedure will result in failure. Even if the agent visually observes a successful demonstration, imitation of this procedure will likely omit critical steps in the procedure. The visual procedure for opening both medicine and traditional bottles are typically identical. The agent lacks understanding of the tactile interaction required to unlock the safety mechanism of the bottle. Only direct observation of forces or instruction can elucidate the correct procedure (Figure 2.19e). Even with knowledge of the correct procedure, opening medicine bottles poses several manipulation challenges that involve feeling and reacting to the internal mechanisms of the bottle cap. Although the presented study takes opening medicine bottles as an example, many other tasks share similar properties and require non-trivial reasoning such as opening locked doors [SGG08].

In this work, we learn a manipulation model from human demonstration that captures observed motion and kinematics, as well as visually latent changes such as forces and internal state Figure 2.19e). We learn this manipulation model for objects that have similar functional properties, but exhibit different geometries and internal configurations that affect how the object must be manipulated.

Two key problems are discussed:

1. how to naturally recover the visually latent force data from the human demonstrations, and
2. how to represent such knowledge and successfully transfer it to a robot?

For the first problem, although some initial results have been reported to reconstruct poses and/or forces exerted by the demonstrator using vision-based methods [ZZM13, WMZ13, ZZZ15, PKQ15, ZJZ16], these methods still have difficulty providing pose and force data precise enough for robot learning. Instead, we utilize an open-source tactile glove [LXM17] designed to measure both hand pose and contact forces across the surface of the hand. These demonstrations are performed naturally, and within a motion capture setup to obtain ground-truth tracking of the objects and human wrist.

For the second problem, our system takes into consideration: i) an And-Or-Graph (AOG) [ZM07] learned from human demonstrations as top-down knowledge for manipulations of an unseen medicine bottle, in which the AOG model uses *fluents* [NC36] to model the changes between pre- and post-conditions of demonstrations in a low-dimensional subspace; and ii) A bottom-up process learned from raw signal data when robot executes to encode transition between pre- and post-conditions. Together, these two processes learn a manipulation model to open medicine bottles.

2.3.1.1 Related Work

Tactile Gloves are common tools to capture demonstration data [DSD08]. In this work, we use a tactile glove [LXM17] to record both human pose and visually hidden forces applied at each proximal and distal phalange, as well as a 4-by-4 grid of sensors to detect forces exerted by the palm. In the literature, most data gloves use IMUs [TKM13, KSR14, SLS15] or curvature sensors [KSE08, KMS11] to track finger pose. To read force, FlexiForce [GSL15] sensors or Velostat [JLK11, LKY15, PMT16] are commonly adopted.

Learning from demonstration (LfD) is a crucial component to building general purpose robots, and a very broad field with rich history. This literature is too expansive to survey here; we refer readers to a survey [ACV09]. Instead, we focus on approaches related to our work: kinesthetic teaching, teleoperation, and imitation learning in the next paragraphs. Note that humans are able to learn quickly from one or only a few examples for a new task [LST15], thus teaching robots to achieve similar performance would enable robots to enter many routine human activities. Our approach requires a relatively small number of examples, approximately 10 examples per bottle.

Kinesthetic teaching and teleoperation both enable direct mappings between demonstrations and executions [ACV09] and have successfully demonstrated capability of learning both motor skills [LB04, KNC11] and manipulation policies [CPB06, KFM14]. However, this direct embodiment mapping, a typically complex function that maps states/actions in demonstrations to states/actions on the robot [ACV09], is ill-suited for manipulation tasks that incorporate forces. Although some robots have built-in force sensing, the demonstrator often cannot receive feedback from forces applied. To address this problem, [KCC11] used kinesthetic teaching to demonstrate positional requirements of a task and employed a secondary haptic demonstration to provide required forces. In contrast, our approach simultaneously integrates both poses and forces within a single demonstration using a tactile data glove, providing a more natural and efficient way to sense force from a demonstration.

Imitation learning has two main streams: i) behavior cloning through supervised demonstrations that directly mimic the demonstrator's behaviors [HD94, MGH09, RGB11, FJ13, LKG14, PJK16, SGR17], and ii) inverse reinforcement learning [AN04, RA07, ZMB08]. While inverse reinforcement learning is limited to Markvoian problems, our approach falls into behavior cloning and is capable of handling both Markovian and non-Markovian problems by utilizing a grammar model.

Two previous work stands out as most relevant to the presented work. [HLD16] use imitation learning coupled with a data glove for opening a set of standard bottles without understanding the internal configuration. This simplification is infeasible when dealing with locking mechanisms of medicine bottles, which require direct and complex manipulation of the cap beyond pure rotation. [SSS17b] uncovered haptic components of a task from teleoperated demonstrations. In contrast, our work learns the manipulation tasks directly from human demonstration using a tactile glove, resulting in more natural and larger variety of demonstrations. In addition, [SSS17b] used a recurrent neural network based method that typically only encodes a few steps of dependencies. However, our work uses an explicit grammar to generate actions, capable of incorporating long-term temporal dependencies.

2.3.1.2 Contribution

This work makes four contributions:

1. Using a tactile glove during demonstrations that enable the robot to utilize both the poses and forces exerted by the demonstrator. In contrast with previous work, our method focuses on integrating visual measurements with physical measurements not observable from vision (*e.g.* forces), capturing latent relationships that are imperceptible from vision alone.

2. Learning a stochastic grammar model that represents the compositional task hierarchy comprising of atomic actions for manipulation tasks, compactly capturing the admissible sequence of actions for all the bottles demonstrated.

3. Learning a bottom-up process that encodes raw haptic signals to account for the transition from a previous state to a new state. Together with the stochastic grammar model as a top-down process, these two processes jointly form the manipulation model.

4. Transferring the learned model from human demonstrations onto a Baxter robot by solv-

ing a correspondence problem [DH02]. This embodiment mapping function directly relates hand pose and contact force from the human to the force-torque sensing and gripper state of the robot; enabling the robot to reason about its haptic measurements using the relations learned from human demonstration.

2.3.1.3 Overview

subsection 2.3.2 outlines the AOG representation and related components. subsection 2.3.3 discusses our data collection environment, instruments, and procedures. subsection 2.3.4, we present how to learn an AOG representation from demonstrations, and how to combine it with raw signals using a bottle-up process to infer the next action. subsection 2.3.5 outlines our robotic system and execution framework. In subsection 2.3.6, we show the results of the system, showcasing our system that integrates both pose and force outperforms the baseline systems. Finally, we conclude and discuss the results in subsection 2.3.7.

2.3.2 Representation

We represent a task demonstrated by agents using an AOG consisting of: i) spatial knowledge to encode the poses of objects and manipulators, and ii) temporal knowledge to encode action sequencing.

2.3.2.1 And-Or Grammar (AOG)

An AOG is a graph-based grammar [ZM07] encoding compositional variability in the demonstrated task sequences. Formally, an AOG $\mathcal{G}$ is represented by a 4-tuple:

$$\mathcal{G} = \langle U, V, \Delta, \Omega_F \rangle. \tag{2.13}$$

An And-node $u \in U$ represents a decomposition of the graph into sub-graphs, and an Or-node $v \in V$ acts as a switch among multiple alternate sub-configurations. The terminal nodes Δ is a set of sub-components representing the lowest level of resolution in the graph. Ω_F represents a set of attributes derived from the terminal nodes. In the context of opening bottles, $\Delta = \{a_1, \ldots, a_m\}$ corresponds to a set of *atomic actions* (subsubsection 2.3.2.2) executed during the task, and Ω_F is a set of *fluent functions* (Section 2.3.2.3) that operate on terminal nodes.

A parse graph, denoted pg, is a specific parse of the AOG by selecting a sub-configuration at each Or-node in the graph. An example of a pg is shown in Figure 2.20, simultaneously incorporating both spatial and temporal knowledge, where the spatial knowledge captures the physical configuration of the robot environment and fluents, and temporal knowledge encodes the sequence of atomic action to complete the task.

2.3.2.2 Atomic Actions

The concept of atomic actions [PSY13] or action primitives [SB08] were proposed in the computer vision community. They are equivalent to the concept of movement primitives in robotics literature [SPN05, PDP13] and represent the finest resolution of an action sequence. In this work, both the human and robot actions are modeled using atomic actions. We aggregate each observed atomic action a_k^h from the demonstration to form the human dictionary of atomic action $\Delta_h = \{a_k^h\}$ and endow the robot with a dictionary of atomic actions, denoted $\Delta_r = \{a_k^r\}$. Here, the subscript k indicates the k-th atomic action in the action sequence. The correspondences between human and robot action labels were manually mapped. Each atomic action represents a 4D human-object interaction (4DHOI) unit, as in [WZZ13a].

2.3.2.3 Fluents

From the human demonstrations, an auto-encoder is trained to embed the space of observed hand geometries, force distributions and the corresponding action label into a low-dimensional subspace. Changes in this low-dimensional subspace correspond to fluent changes. Each fluent function maps the high-dimensional scene configuration, s_k, to a real value, $f(s_k) \mapsto \mathbb{R}$. A fluent change represents a transition between two scene configurations, $\nabla f(s_i, s_j) = f(s_j) - f(s_i)$. For generality, we denote the action at step k as a_k, regardless of whether the action was performed by a human or robot. We denote the scene configuration of the pre-condition as s_k and the post-condition as s_{k+1}. Each action can be characterized by the changes it imposes across all fluents, denoted $\nabla f^{a_k} = \{\nabla f_i(s_k, s_{k+1}), \quad i = 1...n\}$.

Using this notion of fluent changes, the AOG encodes perceptual causality [ST00], represented by state changes between terminal nodes. We express this causal change as a structured equation model (SEM) [Pea09]; *i.e.*, $f_{k+1} = g_{a_k}(f_k)$. This definition relies on the assumption that the human demonstrator/robot is the only causal agent in the environment and the *inertia action* assumption [MT97]. These two assumptions imply a perceptual causal chain between the agent's previous action and the next action; *i.e.*, the post-condition fluents of the previous action are the pre-condition fluents of the current action, depicted by the chain of fluents in Figure 2.20.

2.3.3 Data Collection

A human demonstrator performed opening various types of bottles shown in Figure 2.21. Some of the bottles contain child-safety locking mechanisms that require a procedure beyond simply twisting to unscrew the cap. Most child-safety locks require a particular force to be exerted on a particular part of the bottle. These forces are difficult to infer from visual observation alone. We collected human data on bottles 2, 3, and 5. The remaining bottles were reserved for testing.

Tactile Glove We use a tactile glove [LXM17] to capture these applied forces. The glove reconstructs the pose of each finger using IMUs and detects forces using Velostat sensors on the palm and phalanges. This glove provides 71 degrees of freedom including all pose and force measurements, resulting in an accurate model of the pose of the hand and the forces exerted by each phalange.

Experiment Setup A Vicon motion capture system is used to record the ground truth of poses. The experimental setup is shown in Figure 2.22. Fiducials are attached to each bottle and its lid to track the pose of object parts. One additional fiducial is attached to the back of the tactile glove to capture wrist pose in world space. A camera is used to record the video of data collection procedures to help label the ground truth later.

Data Collection Approximately 10 trials are collected for each grasping strategy for each bottle. Examples are shown in Figure 2.23. Bottle 2 only has one grasping strategy: *pinch-and-twist*. Bottle 3 has two different strategies: *push-and-twist* using the palm, or *push-and-twist* using fingers. Bottle 5 has three valid strategies because it lacks a safety mechanism: *twist*, *push-and-twist*, or *pinch-and-twist*.

Each demonstration is manually labelled, mitigating the correspondence problem between a human action and a robot action. The timestamps of the labeling provide the transition boundaries between actions, *i.e.*, the post-condition of the labelled action and the pre-condition of the next action.

2.3.4 Imitation Learning

2.3.4.1 Problem Definition

The planning objective is to find the best next action a^*_{k+1} given the observed partial parse graph $pg_k = (a_0, \ldots, a_k)$. The pg is planned within the pre-defined action space, and fluents are used as observations. We plan this problem by minimizing the energy of the partial parse graph at each time step:

$$p(pg_{k+1}|pg_k, f_k) = \frac{1}{Z} \exp\{-\mathcal{E}(pg_{k+1}|pg_k, f_k)\}, \tag{2.14}$$

where $Z = \sum_{pg_{k+1}} \exp\{-\mathcal{E}(pg_{k+1}|pg_k, f_k)\}$ is the partition function. We decompose the energy of the parse graph into a top-down term and a bottom-up term, and adopt the notion of top-down and bottom-up as γ and β channels [WZ11] of influence for inference in And-Or graphs, respectively. We define $\mathcal{E}(pg_{k+1}|pg_k, f_k)$ as

$$\mathcal{E}(pg_{k+1}|pg_k, f_k) = \mathcal{E}_\gamma(pg_{k+1}|pg_k) + \mathcal{E}_\beta(pg_{k+1}|pg_k, f_k), \tag{2.15}$$

$$\text{where} \quad \mathcal{E}_\gamma(pg_{k+1}|pg_k) = -\log\left[p(pg_{k+1}|pg_k))\right], \tag{2.16}$$

$$\mathcal{E}_\beta(pg_{k+1}|pg_k, f_k) = -\log\left[p(a_{k+1}|a_k, f_k)\right], \tag{2.17}$$

which incorporates two action planning mechanisms:

- **Top-down Term:** $p(pg_{k+1}|pg_k)$ plans the next action given the sequence of previous actions. It represents the *long-term* relation between all the previous actions and the next action. In this work, an action grammar represented by AOG is first induced using all the valid action sequences. An Earley parser [Ear70] is then adopted to parse the likelihood. See details in subsubsection 2.3.4.2.

- **Bottom-up Term:** $p(a_{k+1}|a_k, f_k)$ plans the next action using both the current action label and observed fluent. This term encodes a *short-term* relation using the current fluent in addition to the pose and force pose sensing. In this work, we convert this planning task to a classification problem, using a neural network to select the action with highest probability. See details in subsubsection 2.3.4.3.

2.3.4.2 Action Planning using AOG

AOG Induction From labelled action sequences of human demonstration, an action grammar G represented by AOG is induced using method presented by Tu *et al*. [TPZ13], resulting in a stochastic context-free grammar with probabilistic Or-nodes. Examples are shown in Figure 2.24. The objective function is the posterior probability of the grammar given the training data X:

$$p(G|X) \propto p(G)p(X|G) = \frac{1}{Z}e^{-\alpha||G||} \prod_{pg_i \in X} p(pg_i|G), \tag{2.18}$$

where $pg_i = (a_1, a_2, \ldots, a_m) \in X$ represents a valid parse graph of atomic actions with length m from the demonstrator.

Top-down Parsing Likelihood Given the learned AOG $\mathcal{G}$, for a grammatically complete parse graph $s = (a_0, \ldots, a_K)$, the parsing likelihood is simply the Viterbi likelihood, denoted by $p(s)$. For an incomplete parse $pg_k = (a_0, \ldots, a_k)$ with length of $k < K$, the parsing likelihood is given by the sum over all grammatically possible actions sequences that begin with pg_k:

$$p(pg_k) = \sum_{s \in \mathcal{G}, s_k = pg_k} p(s), \tag{2.19}$$

where pg_k denotes the first k actions in the parse graph pg. By computing $p(pg_{k+1})$ and $p(pg_k)$ using the Earley parsing likelihood, we compute the top-down term, $p(pg_{k+1}|pg_k)$, through Bayes' rule. The top-down term encodes long-range temporal constraints induced by the AOG.

2.3.4.3 Action Planning using Fluents

We use tactile glove measurements and haptic feedback signals to learn: i) a low-dimensional embedding of the human demonstration, ii) a bottom-up term to plan the next action based on the low-dimensional human embedding, and iii) an embodiment mapping between the robot and the low-dimensional human embedding.

Low-dimensional Embedding We use an auto-encoder to encode the scene configuration into a low-dimensional representation as fluents (Figure 2.25(a)). Changes inside this subspace are treated as fluent changes and are used to infer the next action with observed haptic feedback from the robot. Within this subspace, we train a bottom-up term, $p(a_{k+1}|a_k, f_k)$, to plan the next action using haptic observations of the post-condition of the previous action.

The contact force and pose measurements from the tactile glove are reoriented to the reference frame of the wrist, and concatenated into a feature vector with 159 dimensions. An encoder-decoder architecture, illustrated in Figure 2.25(a), is used to learn a 8-dimensional embedding and reconstructs the full feature from this embedding under a criterion that minimizes the squared residuals between the original feature and the reconstruction:

$$l(\theta; \mathbf{x}^h) = \frac{1}{N}\sum_{i=1}^{N}(x_i^h - \psi(x_i^h; \theta))^2, \tag{2.20}$$

where x_i^h represents one of the N human demonstrations and $\psi(x_i; \theta)$ represents the reconstruction.

Bottom-up Action Planning The bottom-up term $p(a_{k+1}|a_k, f_k)$ takes the form of a multi-class classifier to plan one of the 13 output actions (Figure 2.25(b)). This classification network takes its input from the embedding layer of the auto-encoder and a one-hot encoding of the current action. A softmax layer is used to interpret it as a probability distribution, and the network is trained by minimizing the normalized cross-entropy. All internal layers are linear matrix operators, and use sigmoids for their non-linearities. Combined with the low-dimensional embedding, the bottom-up term incorporates raw tactile signals during manipulations, thus complementing the top down constraints from the action grammar parsing.

Embodiment Mapping The embodiment mapping seeks a function $s_h = \hat{\phi}(s_r)$, where s_h represent the human state of the demonstration and s_r represents the robot's state during execution (Figure 2.25(c)). This function maps haptic sensing on the robot to the low-dimensional embedding of tactile measurements from the human demonstration. A neural network is trained to approximate this function using a small number of robot examples (approximately 15 examples). We supervise robot executions sampled from the learned AOG using the robot's dictionary Δ_r to ensure only successful robot states are mapped to successful demonstrator states. The loss function for this network is the squared residuals:

$$l(\theta; \mathbf{x}^h, \mathbf{x}^r) = \frac{1}{N} \sum_{i=1}^{N} (\phi(x_i^h) - \hat{\phi}(x_i^r; \theta))^2, \tag{2.21}$$

where $\mathbf{x}^h$ represents human states, $\mathbf{x}^r$ represents equivalent robot states, ϕ represents the low-dimensional embedding of human data, and $\hat{\phi}$ represents the embodiment mapping function. The robot utilizes this mapping to plan the next action using the bottom-up term: first map its state to an equivalent human state, then use the human state to plan which action to execute using the bottom-up action planner.

2.3.5 Implementation

2.3.5.1 Robot Platform Setup

We use a dual-armed 7-DoF Baxter robot from Rethink Robotics mounted on a DataSpeed Mobility Base as our robot platform. The robot is equipped with a ReFlex TackkTile gripper on the right wrist, and a Robotiq S85 parallel gripper on the left. In addition, we use Simtrack [PK15] for object pose estimation and tracking with a Kinect One sensor. The entire system runs on ROS [QCG09], and arm motion planning is computed using MoveIt! [SC13]. For object grasping, we implement a geometry based grasping planner to generate grasping poses from CAD models of the objects.

2.3.5.2 System Architecture

The system architecture consists of three major components shown in Figure 2.26:

- **Learning:** The learning phase includes a top-down process and a bottom-up process. The top-down representation is built from segmented human demonstrations, and an AOG is induced to represent valid action sequences (see subsubsection 2.3.4.2). To learn the bottom-up knowledge, three neural networks are trained from raw sensor data (see subsubsection 2.3.4.3).

- **Inference:** During the inference, the top-down term is computed by the Earley parser. The embodiment mapping and classification network are used to compute the bottom-up term, as outlined in subsubsection 2.3.4.1. We plan the next action using Equation 2.14 with the corresponding top-down and bottom-up terms.

- **Execution:** Robot executes the next action either by sampling the AOG, using haptic feedback, or both according to Equation 2.14.

2.3.6 Experiments and Results

2.3.6.1 Experiment Setup

Five bottles were used in the evaluation as shown in Figure 2.21. Bottles 2, 3, and 5 were used during data collection, while the remaining bottles were reserved for testing. Bottles 1, 2, 3, and 4 all have safety mechanisms while bottle 5 does not.

An action sequence is deemed successful if the robot opens the bottle; otherwise, the sequence is a failure. If the robot opens the bottle before finishing the sampled execution, we consider the action sequence that it performed is correct and discard remain actions. We conducted over 300 opening experiments over all of the bottles, resulting in three groups of quantitative results. Each bottle was tested approximately 60 times.

2.3.6.2 Evaluation Criteria

While there may be multiple ways to open each bottle, not all methods are considered equivalent. For instance, Bottle 5 has no safety mechanism, so while *push-and-twist* and *pinch-and-twist* may succeed in opening bottle 5, there is no reason to execute anything other than *twist*. This distinction naturally leads to two levels of evaluation criteria: i) by the end results only, *i.e.*, whether a sequence of actions can successfully open a bottle, and ii) not only successfully open a bottle but also efficiently.

As illustrated above, human demonstrator is treated as an oracle and the corresponding action sequences as perfect executions. We separate robot executions into four different categories:

1. Success, where the robot successfully executed an action sequence that is an exact match to one of the sequences from the human demonstrator;
2. Success, but using at least one extra or wrong action;
3. Failure due to using the wrong action sequence; and
4. Failure due to improper execution (*e.g.* low motor execution accuracy or grasping failure).

2.3.6.3 Qualitative and Quantitative Results

For qualitative analysis, Figure 2.27 shows the robot successfully opening two bottles with (Figure 2.27a) and without (Figure 2.27a) pushing the bottle lid. The force-torque sensor

Evaluation	**bot. 1**	**bot. 2**	**bot. 3**	**bot. 4**	**bot. 5**
Success	8.7%	5.6%	4.4%	8.7%	26.1%
Success (extra/wrong)	21.7%	5.6%	34.8%	47.8%	39.1%
Failure (action)	69.6%	77.7%	60.8%	34.8%	30.4%
Failure (execution)	0%	11.1%	0%	8.7%	4.4%

Table 2.6: Baseline 1, top-down only planning

readings reflect distinguishable differences between performing *push-and-twist* (Figure 2.27c) and *twist* (Figure 2.27d).

We set up three groups of experiments for quantitative results analysis. Table 2.6 shows the results of using top-down only planning, in which the robot executes a sampled action sequence only from the AOG. This method describes the order in which actions were executed but does not capture haptics during manipulations.

Table 2.7 shows the results of using bottom-up only planning. This method incorporates the haptic feedback from the robot sensing, but lacks long-term temporal constraints from the AOG, *i.e.*, it executes a Markovian planning process, in which the next action is determined by the previous action and the current observations as outlined in subsection 2.3.4.

Table 2.8 shows the results of integrating both the top-down planning provided by the AOG and the bottom-up haptic feedback. By utilizing both terms, the temporal sequence of actions is not generated only by sampling from the AOG; instead, each action is generated sequentially by minimizing Equation 2.14.

The proposed top-down and bottom-up planning (Table 2.8) yields large performance

Evaluation	**bot. 1**	**bot. 2**	**bot. 3**	**bot. 4**	**bot. 5**
Success	4.4%	0%	4.4%	0%	4.4%
Success (extra/wrong)	13%	11.8%	30.4%	42.9%	17.4%
Failure (action)	82.6%	76.4%	65.2%	57.1%	78.2%
Failure (execution)	0%	11.8%	0%	0%	0%

Table 2.7: Baseline 2, bottom-up only planning

Evaluation	**bot. 1**	**bot. 2**	**bot. 3**	**bot. 4**	**bot. 5**
Success	8.7%	17.6%	17.4%	20%	60.9%
Success (extra/wrong)	52.2%	17.6%	65.2%	73.3%	17.4%
Failure (action)	39.1%	64.8%	13%	6.7%	21.7%
Failure (execution)	0%	0%	4.4%	0%	0%

Table 2.8: Proposed, top-down and bottom-up planning

improvements over either the top-down (Table 2.6) or bottom-up (Table 2.7) only method. The rate of Success and Success with extra/wrong are dramatically improved while the failure rate due to wrong actions sequences drops significantly.

2.3.6.4 Discussion

a) Why it is important to integrate both top-down and bottom-up terms? In our proposed method, top-down planning generates an action from the non-Markovian AOG, while the bottom-up planning formulates a Markovian process according to robot's haptic feedback. These two processes are complementary to each other and crucial to correctly executing a manipulation task. Specifically, i) the top-down term represents the structure of the task, generating the next action based on previous semantic knowledge and preventing executing irrelevant actions. ii) The bottom-up term encodes real-time sensing information, capturing subtle interactions during manipulations. By combining these two terms, our method is capable of learning from small examples of human demonstrations and planning actions on the fly based on task structure and real-time haptic sensing.

b) Why the success rate of bottle 2 is low? The robot has no haptic feedback and geometry information prior to touching the bottle with its gripper. By sampling the first action after *approach* from the AOG, the probability to plan *pinch* is around 15%, due to the frequency in the human demonstrations. While not reported in Table 2.8, the perfect successful rate for bottle 2 is 100% if the first action after *approach* is *pinch*. Other work [PNZ15] has augmented AOG nodes with attributes to turn the AOG into a context-sensitive grammar. A context-sensitive grammar would increase perfect success rates by considering the type of bottle directly in the top-down term, rather than our current method implicitly inferring the bottle type from haptic feedback.

c) Can the robot derive novel manipulations that are not presented in human demonstrations? In our opinion, there are at least two types of novel manipulations that a robot can derive from human demonstrations: i) generating new action sequences, and

ii) generating new actions. In this work, the proposed method demonstrates the capability of generating novel action sequences through a compositional grammar. However, generating new actions is much more difficult, as the structure and capability of human hands and robot grippers could be dramatically different. For instance, a human demonstration may need to twist twice to open a bottle lid, while a robot gripper may only need to twist once, since some robot grippers are capable of rotating with more freedom than human wrist. Such differences lead to the different success rates of bottle 1, 3, and 4 even though they all require *push-and-twist*: bottle 1 must *push-and-twist* at least twice to open, while bottles 3 and 4 require only one *push-and-twist* action. If the robot could learn and infer the degree of rotation required to open the bottle, the robot could generate a new action to achieve tasks. However, the proposed method does not explore the parameterization of each atomic action in the presented work.

2.3.7 Conclusion

In this work, we present a novel method of naturally capturing visually hidden states of a task and transferring them to the robot through human demonstrations using a tactile glove. The tactile glove provides a data collection method to capture visually hidden causal changes in the scene. Using this latent encoding of the scene, we learn a model to plan the actions of the human demonstrator. The human demonstrations are used to induce an AOG, and the AOG is used to supervise successful executions of opening a bottle.

The robot states of successful executions are mapped to successful demonstrations from the human demonstrator using a low-dimensional embedding of the human tactile feedback. This embodiment mapping solves the correspondence problem using a relatively small number of supervised robot executions. The robot utilizes this mapping in conjunction with the top-down and bottom-up terms to infer the next action to execute.

The proposed method (Table 2.8) shows a marked improvement over two baselines (Table 2.6 and Table 2.7), demonstrating the top-down and bottom-up terms work together to increase the success rate in comparison to using either method alone.

2.3.7.1 Future Work

This work paves the way for additional work regarding visually latent states and corresponding embodiment mappings. We would like to investigate methods to make the system less supervised by clustering the human demonstrations. From the clusters, the robot may not possess an equivalent action in its dictionary and may need to search its action space for an action with equivalent pre- and post-conditions.

The framework presented here could be used to attempt functionally equivalent tasks [ZZZ15]. In this way, the robot could demonstrate understanding the dynamics of the task that needs to be replicated and which can be safely ignored. Experimenting to find functional equivalence is closely related to counterfactual reasoning in the causal domain; such explorations establish causal connections between actions and their effects.

2.4 Enhancing Explainability of Robot Imitation Learning

The ability to provide comprehensive explanations of chosen actions is a hallmark of intelligence. Lack of this ability impedes the general acceptance of AI and robot systems in critical tasks. This work examines what forms of explanations best impart trust and prediction accuracy to human subjects and proposes a framework capable of producing explanations from both functional and mechanistic perspectives. The robot system learns from human demonstrations to open medicine bottles using: (1) an embodied haptic prediction model to extract knowledge from sensory feedback; (2) a stochastic grammar model induced to capture the compositional structure of a multi-step task; and (3) an improved Earley parsing algorithm to jointly leverage both the haptic and grammar models. The robot system not only shows the ability to learn from human demonstrators but also succeeds in opening new, unseen bottles. Using different forms of explanations generated by the robot system, we conducted a psychological experiment to examine what forms of explanations best foster human trust in the robot. We found that comprehensive and real-time visualizations of the robot's internal decisions were more effective in promoting human trust than explanations based on summary text descriptions. In addition, forms of explanation that are best suited to impart trust do not necessarily correspond to the model components contributing to the best task performance. This divergence shows a need for the robotics community to integrate model components to enhance both task execution and human trust in machines.

2.4.1 Introduction

Centuries ago, Aristotle stated that "we do not have knowledge of a thing until we have grasped its why, that is to say, its cause" [Fal19]. A hallmark of humans as social animals is the ability to answer this "why" question by providing comprehensive explanations of the behavior of themselves and others. The drive to seek explanations is deeply rooted in human cognition. Preschool-age children tend to attribute functions to all kinds of objects—clocks, lions, clouds, and trees, as explanations of the activity that these objects were apparently designed to perform [Kel99, GMK99]. The strong human preference and intrinsic motivation for explanation are likely due to its central role in promoting mutual understanding, which fosters trust between agents and thereby enables sophisticated collaboration [Lom06, Tom10].

However, a strong human desire for explanations has not been sufficiently recognized by modern artificial intelligence (AI) systems, in which most methods primarily focus on task performance [Gun17]. Consequently, robot systems are still in their infancy in developing the ability to explain their own behavior when confronting noisy sensory inputs and executing complex multi-step decision processes. Planner-based robot systems can generally provide an interpretable account for their actions to humans (*e.g.*, by Markov decision processes [FHL16, HS17], HTN [EHN96], or STRIPS [FN71]); but these planners struggle to explain how their symbolic-level knowledge is derived from low-level sensory inputs. In contrast, robots equipped with Deep Neural Networks (DNNs) [HOT06] have demonstrated impressive performance in certain specific tasks due to their powerful ability to handle low-level noisy sensory inputs [DCH16, LLS15]. However, DNN-based methods have well-known limitations, notably including a lack of interpretability of the knowledge representation [Mar18, MP17, Dom15]. Some recent DNN work addresses this issue using saliency maps [KRD18, YKY18] or modularized components [HAD18, ZNZ18]. These data-driven approaches have demonstrated strong capabilities of handling noisy real-time sensory inputs, distilling the raw input to predict the effect and determine the next action. However, little work has been done to develop the synergy between the classic symbolic AI and the recent development of DNNs to empower machines with the ability to provide comprehensive explanations of their behavior.

To fill in this gap, the present project aims to disentangle explainability from task performance, measuring each separately to gauge the advantages and limitations of two major families of representations—symbolic representations and data-driven representations—in both task performance and imparting trust to humans. The goals are to explore: (i) what constitutes a good performer for a complex robot manipulation task? (ii) How can we construct an effective explainer to explain robot behavior and impart trust to humans?

To answer these questions, this work develops an integrated framework consisting of a symbolic action planner using a stochastic grammar as the planner-ba sed representation and a haptic prediction model based on neural networks to form the data-driven representation. We examine this integrated framework in a robot system using a contact-rich manipulation task of opening medicine bottles with various safety lock mechanisms. From the performer's perspective, this task is a challenging learning problem involving subtle manipulations, as it requires a robot to push or squeeze the bottle in various places to unlock the cap. At the same time, the task is also challenging for explanation, as visual information alone from a human demonstrator is insufficient to provide an effective explanation. Rather, the contact forces between the agent and the bottle provide the *hidden* "key" to unlock the bottle, and these forces cannot be observed directly from visual input.

To constitute a good performer, the robot system proposed here cooperatively combines multiple sources of information for high performance, enabling synergy between a high-level symbolic action planner and a low-level haptic prediction model based on sensory inputs. A stochastic grammar model is learned from human demonstrations and serves as a symbolic representation capturing the compositional nature and long-term constraints of a task [TPZ13]. A haptic prediction model is trained using sensory information provided by human demonstrations (*i.e.*, imposed forces and observed human poses) to acquire knowledge of the task. The symbolic planner and haptic model are combined in a principled manner using an improved Generalized Earley Parser (GEP) [QJZ18], which predicts the next robot action by integrating the high-level symbolic planner with the low-level haptic model. The learning from demonstration framework presented here shares a similar spirit of our previous work [EGX17] but with a new haptic model and a more principled manner, namely the GEP, to integrate the haptic and grammar models. Computational experiments demonstrate a strong performance improvement over the symbolic planner or haptic model alone.

To construct an effective explainer, the proposed approach draws from major types of explanations in human learning and reasoning that may constitute representations to foster trust by promoting mutual understanding between agents. Previous studies suggest humans generate explanations from *functional* perspectives that describe the *effects* or *goals* of actions and from *mechanistic* perspectives that focus on behavior as a process [Lom13]. The haptic prediction model is able to provide a functional explanation by visualizing the essential haptic signals (*i.e.*, *effects* of the previous action) to determine the next action. The symbolic action planner is capable of providing a mechanistic explanation by visualizing multiple planning steps (instead of just one) to describe the *process* of the task. The proposed robot system provides both functional and mechanistic explanations using the haptic model and symbolic planner, respectively.

To examine how well robot-generated explanations impart human trust, we conduct human experiments to assess whether explanations provided by the robot system can foster trust in human users, and if so, what forms of explanation are the most effective in enhancing human trust in machines. In this work, we refer to the cognitive component of "trust" [Sim07] based on rationality. Cognitive trust is especially important in forming trust within secondary groups (such as human-machine relations) [LW85] compared to the emotional component typically more important in primary group relations (such as family and close friends). Our psychological experiment focuses on cognitive trust, stressing on a belief or an evaluation with "good rational reasons," as this is the crucial ingredient of human-machine trust built on specific beliefs and goals with attention to evaluations and expectations [CF98]. Specifically, human participants were asked to report qualitative trust ratings after observing robot action sequences along with different forms of explanations for the robot's internal decision-making as it solved a manipulation task. Then, participants observed similar but new robot executions *without* access to explanations and were asked to predict how the robot system is likely to behave across time. These empirical findings shed light on the importance of learning human-centric models that make the robot system explainable, trustworthy, and predictable to human users. Our results show that forms of explanation that are best suited to impart trust do not necessarily correspond to those components contributing to the best task performance. This divergence shows a need for the robotics community to adopt model components that are more likely to foster human trust and integrate these components with other model components enabling high task performance.

2.4.2 Results

Figure 2.28 illustrates the overall procedures, wherein the proposed integration framework, the Generalized Earley Parser (GEP) [QJZ18], efficiently combines a symbolic action planner and a data-driven haptic model to achieve high task performance and effective explanation. To this end, we first describe the procedure and data collection of human demonstrations, followed by the learning approaches. Next, we provide quantitative results as the success rate of the robot system in performing the task and assess the contributions from different modules of the system in task performance. We end the section with an analysis of human experiments with different types of explanations generated from the learned models, showing how human qualitative trust and prediction accuracy are reflected according to various provided explanations.

2.4.2.1 Robot Learning

To learn from human demonstrations, our robot system utilizes an efficient encoding and representation of both haptic inputs and symbolic semantics of the manipulation task. The specific task, opening medicine bottles, requires inferring about both the hand pose and the forces imposed on the bottle; agents must understand and enact the correct sequence of pose and force manipulations to succeed based on both the learned knowledge from human demonstrations and the real-time haptic sensory input.

We utilize a tactile glove with force sensor [LXM17] to capture both the poses and the forces involved in human demonstrations in opening medicine bottles that require a visually latent interaction between the hand and the cap, *e.g.*, pushing as indicated in Figure 2.28A. A total of 64 human demonstrations, collected in [EGX17], of opening 3 different medicine bottles serve as the training data. These 3 bottles have different locking mechanisms: no safety lock mechanism, a *push-twist* locking mechanism, and a *pinch-twist* locking mechanism. To test the generalization ability of the robot system, we conduct a generalization experiment with new scenarios different from training data, either a new bottle (Figure 2.31B) or a bottle with a modified cap with significantly different haptic signals (Figure 2.36). The locking mechanisms of the bottles in the generalization experiment are similar but not identical (in terms of size, shape, and haptic signals) to the bottles used in human demonstrations. The haptic signals for the generalization bottles are significantly different from bottles used in testing, posing challenges in transferring the learned knowledge to novel unseen cases.

Embodied haptic model Using human demonstrations, the robot learns a manipulation strategy based on the observed poses and forces exerted by human demonstrators. One challenge in learning manipulation policies from human demonstration involves different embodiments between robots and human demonstrators. A human hand has five fingers, whereas a robot gripper may only have two or three fingers; each embodiment exerts different sensory patterns even when performing the very same manipulation. Hence, the embodied haptic model for the robot system cannot simply duplicate human poses and forces exerted by human hands; instead, a robot should imitate the actions with the goal to produce the same end-effect in manipulating the medicine bottle (*e.g.*, imposing a certain force on the cap). The critical approach in our model is to employ embodied prediction, *i.e.*, let the robot imagine its current haptic state as a human demonstrator and predict what action the demonstrator would have executed under similar circumstances in the next time step.

Figure 2.29 illustrates the force patterns exerted by a robot and a human demonstrator. As shown in panels A and C, due to the differences between a robot gripper and a human hand, the haptic sensing data from robots and humans show very different patterns from each other in terms of dimensionality and duration within each segmented action (illustrated by the colored segments).

To address the cross-embodiment problem, we train a haptic model in a similar approach as in [EGX17] to predict which action the robot should take next based on perceived human and robot forces and poses. The present haptic model learns a prediction model in a three-step process: (i) learning an autoencoder that constructs a low-dimensional embedding of human demonstrations containing poses and forces, as shown in Figure 2.29B. (ii) Training an embodiment mapping to map robot states to equivalent human embeddings, thereby allowing the robot to imagine itself as a human demonstrator to produce the same force, achieving functional equivalence to generate the same end effect as the human demonstrator. This embodiment mapping is trained in a supervised fashion, using labeled equivalent robot and human states. (iii) Training a next action predictor based on the human embeddings and the current action. This action predictor is also trained in a supervised fashion, using segmented human demonstrations. See section 2.4.4.1 for additional training details.

The robot predicts the next action based on the mapped human embedding using a multi-class classifier; see details in Model Learning Details in subsection 2.4.4. We denote this prediction process as our *haptic model*. Intuitively, the embodied haptic predictions endow the robot with the ability to ask itself: *if I imagine myself as the human demonstrator, which action would the human have taken next based on the poses and forces exerted by their hand?* Hence, the resulting haptic model provides a *functional* explanation regarding the forces exerted by the robot's actions.

Symbolic action planner Opening medicine bottles is a challenging multi-step manipulation, as one may need to push on the cap to unlock it (visually unobservable), twist it, and then pull it open. A symbolic representation is advantageous to capture the necessary long-term constraints of the task. From labeled action sequences of human demonstrations, we induce a temporal And-Or Graph (T-AOG), a probabilistic graphical model describing a stochastic, hierarchical, and compositional context-free grammar [ZM07], wherein an And-node encodes a decomposition of the graph into sub-graphs, an Or-node reflects a switch among multiple alternate sub-configurations, and the terminal nodes consist of a set of action primitives (such as *push*, *twist*, *pull*). A corpus of sentences (*i.e.*, action sequences in our case) is fed to the grammar induction algorithm presented in [TPZ13], and the grammar is induced by greedily generating And-Or fragments according to the data likelihood; the fragments represent compositional substructures that are combined to form a complete grammar. In our case, the grammar is learned from segmented and labeled human demonstrations. The resulting grammar offers a compact symbolic representation of the task and captures the hierarchical structure of the task, including different action sequences for different bottles, as well as different action sequences for the same bottle. Examples of the T-AOG learning progress are shown in Figure 2.30. The nodes connected by red edges in

Figure 2.30C indicate a parse graph sampled from the grammar, and its terminal nodes compose an action sequence for robot execution.

Based on the action sequences observed in human demonstrations, the induced grammar can be used to parse and predict robot action sequences likely to lead to successfully opening the medicine bottle, assuming each robot action corresponds to an equivalent human action. The induced grammar can be parsed to generate new, unseen, and valid action sequences for solving similar tasks (*e.g.*, opening different medicine bottles), and thus the grammar can be used with symbolic planning methods, such as the Earley Parser [QJZ18]. We denote the process of planning actions using a parser and the action grammar as the *symbolic planner*. Hence, the symbolic planner endows the robot with the ability to ask itself from a *mechanistic* perspective: *based on what I have done thus far and what I observed the human do, which actions are likely to open the bottle at the end of the sequence?*

Integration of symbolic planner and haptic model To integrate the long-term task structure induced by the symbolic planner and manipulation strategy learned from haptic signals, we seek to combine the symbolic action planner and embodied haptic model using the Generalized Earley Parser (GEP) [QJZ18]. The GEP is a grammar parser that works on a sequence of sensory data; it combines any context-free grammar model with probabilistic beliefs over possible labels (grammar terminals) of sensory data. The output of the GEP is the optimal segmentation and label sentence of the raw sensory data; a label sentence is optimal when its probability is maximized according to the grammar priors and the input belief over labels while being grammatically correct. The core idea of the GEP is to efficiently search in the language space defined by the grammar to find the optimal label sentence.

To adopt the GEP for a robot system, we modify the GEP presented in [QJZ18] for online planning. The grammar for the GEP remains the same grammar used in the symbolic planner; however, the GEP's probabilistic beliefs come from the softmax distribution from the haptic model. During the action planning process, a stochastic distribution of action labels predicted by the haptic model is fed into the GEP at every time step. The GEP aggregates the entire symbolic planning history with the current haptic prediction and outputs the best parse to plan the most likely next action. subsection 2.4.4 introduces more details about the algorithm. Intuitively, such an integration of the symbolic planner and haptic model enables the robot to ask itself: *based on the human demonstration, the poses and forces I perceive right now and the action sequence I have executed thus far, which action has the highest likelihood of opening the bottle?*

Robot Results Figure 2.31A and Figure 2.31B show the success rate of the robot in performing the task of opening the 3 medicine bottles used in human demonstrations and 2 new, unseen medicine bottles, respectively; see more generalization results in subsubsection 2.4.5.1. The 2 generalization bottles locking mechanisms that are similar (but not identical) to the ones used in human demonstrations, and the low-level haptic signals are significantly different, posing challenges in transferring the learned knowledge to novel unseen cases. Each bottle and model was executed 31 times on our robot platform. In the testing experiments, Bottle 1 is a regular bottle without a locking mechanism, Bottle 2 has a *push-twist* locking mechanism, and Bottle 3 requires pinching *specific points* on the lid to unlock. In the generalization experiments, Bottle 4 also does not have a locking mechanism, and Bottle 5 has a *push-twist* locking mechanism but with a different shape, size, and haptic signals compared with the ones in the human demonstrations. For both the testing and generalization experiments, the robot's task performance measured by the success rates decreases as the bottle's locking mechanism becomes more complex, as expected.

To quantitatively compare the difference between the model components, we conduct ablative experiments on robot task performance using only the symbolic planner and only the haptic model; see Figure 2.31. The haptic model and symbolic planner vary in their relative individual performance, but the combined planner using the GEP yields the best performance for all cases. Hence, integrating both the long-term task structure provided by the symbolic planner and the real-time sensory information provided by the haptic model yields the best robot performance. The symbolic planner provides long-term action planning and ensures the robot executes an action sequence capturing the high-level structure of the task. However, models that solely rely on these symbolic structures are brittle to adjust to perturbations of haptic signals, especially when the task relies more on the haptics as the complexity increases. On the other hand, models that rely purely on haptic signals are unable to impose multi-step task constraints, thus may fail to infer a correct sequence of actions based on the execution history. Our results confirm that by combining these modalities together, the robot achieves the highest task performance.

Given that multiple modalities are involved in the GEP's performance, it is crucial to assess the contributions from different model components. We ran the χ^2-test to determine if different models (GEP, symbolic, and haptic) are statistically different in their ability to open five bottles (3 bottles used in human demonstrations and 2 new bottles used in the generalization task). The robot performs the manipulation task 31 times per medicine bottle. With the significance level of 0.05, the results show that the performance of the GEP model is significantly better than both symbolic model ($\chi^2(1) = 10.0916, p = 0.0015$) and haptic model ($\chi^2(1) = 13.0106, p < 0.001$). Performance does not show difference between the symbolic model and the haptic model, $\chi^2(1) = 0.1263, p = 0.7232$. These results suggest that both haptic model and symbolic planner contribute to good task performance; when the two processes are integrated with the GEP, the success rate of the robot for opening medicine bottles is improved compared to the performance by the single-module models based on either the haptic model or the symbolic planner.

2.4.2.2 Explanation Generation

The haptic model and symbolic planner are capable of providing explanations to humans about robot behavior in real-time. Mechanistic explanations can be generated by the symbolic planner in the form of action sequences as they represent the process of opening a medicine bottle. Functional explanations can be provided by a visualization of the internal robot gripper state (effects) used in the haptic model. It is worth noting that these models are capable of *providing* such explanations but are not the only means of producing them. Alternative action planners and haptic models could produce similar explanations, as long as the robot systems are able to learn the corresponding representations for haptic prediction and task structure. Figure 2.32 shows the explanation panels over an action sequence. These visualizations are shown in real-time, providing direct temporal links between explanation and execution.

2.4.2.3 Human Experiment

Experimental Design The human experiment aims to examine whether providing explanations generated from the robot's internal decisions foster human trust to machines and what forms of explanation are the most effective in enhancing human trust. We conducted a psychological study with 150 participants; each was randomly assigned to one of five groups. Our experimental setup consisted of two phases: familiarization and prediction. During familiarization, all groups viewed the RGB video, and some groups were also provided with an explanation panel. During the second phase of the prediction task, all groups only observed RGB videos.

The five groups consist of the baseline no-explanation group, symbolic explanation group, haptic explanation group, GEP explanation group, and text explanation group. For the baseline no-explanation group, participants only viewed RGB videos recorded from a robot attempting to open a medicine bottle, as shown in Figure 2.33A. For the other four groups, participants viewed the same RGB video of robot executions and simultaneously were presented with different explanatory panels on the right side of the screen. Specifically, the symbolic group viewed the symbolic action planner illustrating the robot's real-time inner decision-making, as shown in Figure 2.33B. The haptic group viewed the real-time haptic visualization panel, as shown in Figure 2.33C. The GEP group viewed the combined explanatory panel, including the real-time robot's symbolic planning and an illustration of haptic signals from the robot's manipulator, namely both Figure 2.33B-C. The text explanation group was provided a text description that summarizes why the robot succeeded or failed to open the medicine bottle *at the end* of the video, as shown in Figure 2.33D. See a summary in Figure 2.33E for the five experimental groups.

During the familiarization phase, participants were provided two demonstrations of robot executions, with one successful execution of opening a medicine bottle and one failed execution without opening the same bottle. The presentation order of the two demonstrations was counterbalanced across participants. After observing robot executions with explanation panels, participants were first asked to provide a trust rating for the question: *to what extent do you* ***trust/believe*** *this robot possesses the ability to open a medicine bottle?* on a scale between 0 and 100. The question was adopted from the questionnaire of measuring human trust in automated systems [JBD00]. This question also clearly included the goal of the system, *to open a medicine bottle*, to enhance the reliability in trust measures [CF98]. Hence, the rating provided a direct qualitative measure of human trust to the robot's ability to open medicine bottles.

In addition, we designed the second measure to assess the quantitative aspects of trust. We adopted the definition by Castelfranchi and Falcone [CF98] that quantitative trust is based on the quantitative dimensions of its cognitive constituents. Specifically, as the greater the human's belief in the machine's competence and performance, the greater the human trust in machines. In the prediction phase, we asked participants to predict the robot's next action in a new execution with the same task of opening the same medicine bottle. Participants viewed different segments of actions performed by the robot and were asked to answer the prediction question over time. For this measure, participants in all five groups only observed RGB videos of robot execution during the prediction phase; no group had access to any explanatory panel after the familiarization phase. The prediction accuracy was computed as the quantitative measure of trust, with the presumption that, as the robot behavior is more predictable to humans, greater prediction accuracy indicates higher degrees of trust.

Human Study Results Figure 2.34A shows human trust ratings from the five different groups. The analysis of variance (ANOVA) reveals a significant main effect of groups ($F(4, 145) = 2.848; p = 0.026$) with the significance level of 0.05. This result suggests that providing explanations about robot behavior in different forms impacts the degree of human trust to the robot system. Furthermore, we find that the GEP group with both symbolic and haptic explanation panels yields the highest trust rating, with a significantly better rating than the baseline group in which explanations are not provided (independent-samples t-test, $t(58) = 2.421; p = 0.019$). Interestingly, the GEP group shows greater trust rating than the text group in which a summary description is provided to explain the robot behavior ($t(58) = 2.352; p = 0.022$), indicating detailed explanations of robot's internal decisions over time is much more effective in fostering human trust than a summary text description to explain robot behavior. In addition, trust ratings in the symbolic group are also higher than ratings in the baseline group ($t(58) = 2.269; p = 0.027$) and higher than ratings in the text explanation group ($t(58) = 2.222; p = 0.030$), suggesting symbolic explanations play an important role in fostering human trust of the robot system. However, the trust ratings in the haptic explanation group are not significantly different from the baseline group, implying that explanations only based on haptic signals are not effective ways to gain human trust despite the explanations are also provided in real-time. No other significant group differences are observed between any other pairing of the groups.

The second trust measure based on prediction accuracy yields similar results. All groups provide action predictions above the chance-level performance of 0.125 (as there are 8 actions to choose from), showing that humans are able to predict the robot's behavior after only a couple of observations of a robot performing a task. The ANOVA analysis shows a significant main effect of groups ($F(4, 145) = 3.123; p = 0.017$), revealing the impact of provided explanations on the accuracy of predicting the robot's actions. As shown in Figure 2.34B, participants in the GEP group yield significantly higher prediction accuracy than those in the baseline group ($t(58) = 3.285; p = 0.002$). Prediction accuracy of the symbolic group also yields better performance than the baseline group ($t(58) = 2.99; p = 0.004$). Interestingly, we find that the text group shows higher prediction accuracy than the baseline group ($t(58) = 2.144; p = 0.036$). This result is likely due to the summary text description providing a loose description of the robot's action plan; such a description decouples the explanation from the temporal execution of the robot. The prediction accuracy data did not reveal any other significant group differences among other pairs of groups.

In general, humans appear to need real-time, symbolic explanations of the robot's internal decisions for performed action sequences in order to establish trust in machines when performing multi-step complex tasks. Summary text explanations and explanations only based on haptic signals are not effective ways to gain human trust, and the GEP and symbolic group foster similar degrees of human trust to the robot system according to both measures of trust.

2.4.3 Discussion

In terms of performance, our results demonstrate that a robot system can learn to solve challenging tasks from a small number of human demonstrations of opening three medicine bottles. This success in learning from small data is primarily supported by learning multiple models for joint inference of task structure and sensory predictions. We found that neither purely symbolic planning nor a haptic model is as successful as an integrated model including both processes.

Our model results also suggest that the relative contributions from individual modules, namely the symbolic planner and haptic predictions, can be influenced by the complexity of the manipulation task. For example, in testing scenarios, for Bottle 1 with no safety locking mechanism, the symbolic planner slightly outperforms the haptic model. Conversely, to open

Bottle 3 that has complex locking mechanisms, the haptic model outperforms the symbolic planner as haptic signals provide critical information for the pinch action needed to unlock the safety cap. For generalization scenarios with new medicine bottles that are unseen in human demonstrations, the symbolic planner maintains a similar performance compared to equivalent bottles in the testing scenarios, whereas the haptic model performance decreases significantly. We also note that the symbolic planner performance decreases faster as complexity increases, indicating pure symbolic planners are more brittle to circumstances that require additional haptic sensing. Furthermore, as bottle complexity increases, model performance benefits more from integrating symbolic planner and haptic signals. This trend suggests that more complex tasks require the optimal combination of multiple models to produce effective action sequences.

In terms of explainability, we found that reasonable explanations generated by the robot system are important for fostering human trust in machines. Our experiments show that human users place more trust in a robot system that has the ability to provide explanations using symbolic planning. An intriguing finding from these experiments is that providing explanations in the form of a summarized text description of robot behavior is not an effective way to foster human trust. The symbolic explanation panel and text summary panel both provide critical descriptions of the robot's behavior at the abstract level, explaining why a robot succeeded or failed the task. However, the explanations provided by the two panels differ in their degree of detail and temporal presentation. The text explanation provides a loose description of the important actions that the robot executes after the robot finished the sequence. In contrast, the symbolic explanation included in the GEP's panel provides human participants with real-time internal decisions that the robot is planning to execute at each step. This mode of explanation enables the visualization of task structure for every action executed during the sequence and likely evokes a sense that the robot actively makes rational decisions.

However, it is not the case that a detailed explanation is always the best approach to foster human trust. A functional explanation of real-time haptic signals is not very effective in gaining human trust in this particular task. Information at the haptic level may be excessively tedious and may not yield a sense of rational agency that allows the robot to gain human trust. To establish human trust in machines and enable humans to predict robot behaviors, it appears that an effective explanation should provide a symbolic interpretation and maintain a tight temporal coupling between the explanation and the robot's immediate behavior.

Taking together of both performance and explanation, we found that the relative contributions of different models for generating explanations may differ from their contributions to maximizing robot performance. For task performance, the haptic model plays an important role for the robot to successfully open a medicine bottle with high complexity. However, the major contribution to gain human trust is made by real-time mechanistic explanations provided by the symbolic planner. Hence, model components that impart the most trust do not necessarily correspond to those components contributing to the best task performance. This divergence is intuitive as there is no requirement that components responsible for generating better explanations are the same components contributing to task performance; they are optimizing different goals. This divergence also implies that the robotics community should adopt model components that gain human trust, while also integrating these components with high-performance ones to maximize both human trust and successful execution. Robots endowed with explainable models offer an important step towards integrating robots into daily life and work.

2.4.4 Materials and Methods

2.4.4.1 Model Learning Details

Embodied haptic model details The embodied haptic model leverages low-level haptic signals obtained from the robot's manipulator to make action predictions based on the human poses and forces collected with the tactile glove. This embodied haptic sensing allows the robot to reason about (i) its own haptic feedback by imagining itself as a human demonstrator, and (ii) what a human would have done under similar poses and forces. The critical challenge here is to learn a mapping between equivalent robot and human states, which is difficult due to the different embodiments. From the perspective of generalization, manually designed embodiment mappings are not desirable. To learn from human demonstrations on arbitrary robot embodiments, we propose an embodied haptic model general enough to learn between an arbitrary robot embodiment and a human demonstrator.

The embodied haptic model consists of three major components: (i) an autoencoder to encode the human demonstration in a low-dimensional subspace; we refer to the reduced embedding as the *human embedding.* (ii) An *embodiment mapping* that maps robot states onto a corresponding human embedding, providing the robot with the ability to imagine itself as a human demonstrator. (iii) An *action predictor* that takes a human embedding and the current action executing as the input and predicts the next action to execute, trained using the action labels from human demonstrations. Figure 2.29B shows the embodied haptic network architecture. Using this network architecture, the robot infers what action a human was likely to execute based on this inferred human state. This embodied action prediction model picks the next action according to:

$$a_{t+1} \sim p(\cdot | f_t, a_t), \tag{2.22}$$

where a_{t+1} is the next action, f_t is the robot's current haptic sensing, and a_t is the current action.

The autoencoder network takes an 80-dimensional vector from the human demonstration (26 for the force sensors and 54 for the poses of each link in the human hand) and uses the post-condition vector, *i.e.*, the average of last N frames (we choose $N = 2$ to minimize the variance), of each action in the demonstration as input; see the Autoencoder portion of Figure 2.29B. This input is then reduced to an 8-dimensional human embedding. Given a human demonstration, the autoencoder enables the dimensionality reduction to an 8-dimensional representation.

The embodiment mapping maps from the robot's 4-dimensional post-condition vector, *i.e.*, the average of the last N frames (different from human post-condition due a faster sample rate on the robot gripper compared to the tactile glove; we choose $N = 10$), to an imagined human embedding; see the Embodiment Mapping portion of Figure 2.29B. This mapping allows the robot to imagine its current haptic state as an equivalent low-dimensional human embedding. The robot's 4-dimensional post-condition vector consists of the gripper position (1 dimension) and the forces applied by the gripper (3 dimensions). The embodiment mapping network uses a 256-dimensional latent representation, and this latent representation is then mapped to the 8-dimensional human embedding.

To train the embodiment mapping network, the robot first executes a series of supervised actions where if the action produces the correct final state of the action, the robot post-condition vector is saved as input for network training. Next, human demonstrations of equivalent actions are fed through the autoencoder to produce a set of human embeddings. These human embeddings are considered as the ground-truth target outputs for the embodiment mapping network, regardless of the current reconstruction accuracy of the autoencoder network. Then the robot execution data is fed into the embodiment mapping network, producing an imagined human embodiment. The embodiment mapping network optimizes to reduce the loss between its output from the robot post-condition input and the target output.

For the action predictor, the 8-dimensional human embedding and the 10-dimensional current action are mapped to a 128-dimensional latent representation, and the latent representation is then mapped to a final 10-dimensional action probability vector (*i.e.*, the next action); see Action Prediction portion of Figure 2.29B. This network is trained using human demonstration data, where a demonstration is fed through the autoencoder to produce a human embedding, and that human embedding and the one-hot vector of the current action execution are fed as the input to the prediction network; the ground-truth is the next action executed in the human demonstration.

The network in Figure 2.29B is trained in an end-to-end fashion with three different loss functions in a two-step process: (i) a forward pass through the autoencoder to update the human embedding z_h. After computing the error $L_{\text{reconstruct}}$ between the reconstruction s'_h and the ground-truth human data s_h, we back-propagate the gradient and optimize the autoencoder:

$$L_{\text{reconstruct}}(s'_h, s_h) = \frac{1}{2}(s'_h - s_h)^2. \tag{2.23}$$

(ii) A forward pass through the embodiment mapping and the action prediction network. The embodiment mapping is trained by minimizing the difference L_{mapping} between the embodied robot embedding z_r and target human embedding z_h; the target human embedding z_h is acquired through a forward pass through the autoencoder using a human demonstration post-condition of the same action label, s_h. We compute the cross-entropy loss $L_{\text{prediction}}$ of the predicted action label a' and the ground-truth action label a to optimize this forward pass:

$$\begin{aligned} L_{\text{planning}}(a', a) &= L_{\text{mapping}} + \beta \cdot L_{\text{prediction}} \\ L_{\text{mapping}} &= \frac{1}{2}(z_r - z_h)^2 \\ L_{\text{prediction}} &= H(p(a'), q(a)), \end{aligned} \tag{2.24}$$

where H is the cross-entropy, p is the model prediction distribution, q is the ground-truth distribution, and β is the balancing parameter between the two losses; see subsubsection 2.4.5.3 for detailed parameters and network architecture.

A similar embodied haptic model was presented in [EGX17] but with two separate loss functions, which is more difficult to train compared to the single loss function presented in this work. A clear limitation of the haptic model is the lack of long-term action planning. To address this problem, we discuss the symbolic task planner below and then discuss how we integrate the haptic model with the symbolic planner to jointly find the optimal action.

Symbolic planner details To encode the long-term temporal structure of the task, we endow a symbolic action planner that encodes semantic knowledge of the task execution sequence. The symbolic planner utilizes stochastic context-free grammars to represent tasks, where the terminal nodes (words) are actions and sentences are action sequences. Given an action grammar, the planner finds the optimal action to execute next based on the action history, analogous to predicting the next word given a partial sentence.

The action grammar is induced using labeled human demonstrations, and we assume the robot has an equivalent action for each human action. Each demonstration forms a sentence, x_i, and the collection of sentences from a corpus, $x_i \in X$. The segmented demonstrations are used to induce a stochastic context-free grammar using the method presented in [TPZ13]. This method pursues T-AOG fragments to maximize the likelihood of the grammar producing the given corpus. The objective function is the posterior probability of the grammar given the training data X:

$$p(G|X) \propto p(G)p(X|G) = \frac{1}{Z}e^{-\alpha||G||}\prod_{x_i \in X} p(x_i|G), \tag{2.25}$$

where G is the grammar, $x_i = (a_1, a_2, \dots, a_m) \in X$ represents a valid sequence of actions with length m from the demonstrator, α is a constant, $||G||$ is the size of the grammar, and Z is the normalizing factor. Figure 2.30 shows examples of induced grammars of actions.

During the symbolic planning process, this grammar is used to compute which action is the most likely to open the bottle based on the action sequence executed thus far and the space of possible future actions. A pure symbolic planner picks the optimal action based on the grammar prior:

$$a^*_{t+1} = \arg\max_{a_{t+1}} p(a_{t+1} \mid a_{0:t}, G), \tag{2.26}$$

where a_{t+1} is the next action, and $a_{0:t}$ is the action sequence executed thus far. This grammar prior can be obtained by a division of two grammar prefix probabilities: $p(a_{t+1} \mid a_{0:t}, G) = \frac{p(a_{0:t+1} \mid G)}{p(a_{0:t} \mid G)}$, where the grammar prefix probability $p(a_{0:t} \mid G)$ measures the probability that $a_{0:t}$ occurs as a prefix of an action sequence generated by the action grammar G. Based on a classic parsing algorithm—the Earley parser [Ear70]—and dynamic programming, the grammar prefix probability can be obtained efficiently by the Earley-Stolcke parsing algorithm [Sto95]. An example of pure symbolic planning is shown in Figure 2.39.

However, due to the fixed structure and probabilities encoded in the grammar, always choosing the action sequence with the highest grammar prior is problematic since it provides no flexibility. An alternative pure symbolic planner picks the next action to execute by sampling from the grammar prior:

$$a_{t+1} \sim p(\cdot \mid a_{0:t}, G). \tag{2.27}$$

In this way, the symbolic planner samples different grammatically-correct action sequences and increases the adaptability of the symbolic planner. In the experiments, we choose to sample action sequences from the grammar prior.

In contrast to the haptic model, this symbolic planner lacks the adaptability to real-time sensor data. However, this planner encodes long-term temporal constraints that are missing from the haptic model, since only grammatically-correct sentences have non-zero probabilities. The GEP adopted in this work naturally combines the benefits of both the haptic model and the symbolic planner; see the next section.

Generalized Earley Parser (GEP) details The robot imitates the human demonstrator by combining the symbolic planner and the haptic model. The integrated model finds the next optimal action considering both the action grammar G and the haptic input f_t:

$$a^*_{t+1} = \arg\max_{a_{t+1}} p(a_{t+1} \mid a_{0:t}, f_t, G). \tag{2.28}$$

Conceptually, this can be thought of as a posterior probability that considers both the grammar prior and the haptic signal likelihood. The next optimal action is computed by an improved Generalized Earley Parser (GEP) [QJZ18]; GEP is an extension of the classic Earley parser [Ear70]. In the present work, we further extend the original GEP to make it applicable to multisensory inputs and provide explanation in real-time for robot systems, instead of for offline video processing; see details in section 2.4.4.1 in suppementary material.

The computational process of GEP is to find the optimal label sentence according to both a grammar and a classifier output of probabilities of labels for each time step. In our case, the labels are actions, and the classifier output is given by the haptic model. Optimality here means maximizing the joint probability of the action sequence according to the grammar prior and haptic model output while being grammatically correct.

The core idea of the algorithm is to directly and efficiently search for the optimal label sentence in the language defined by the grammar. The grammar constrains the search space to ensure that the sentence is always grammatically correct. Specifically, a heuristic search is performed on the prefix tree expanded according to the grammar, where the path from the root to a node represents a partial sentence (prefix of an action sequence).

GEP is a grammar parser, capable of combining the symbolic planner with low-level sensory input (haptic signals in this work). The search process in the GEP starts from the root node of the prefix tree, which is an empty terminal symbol indicating no terminals are parsed. The search terminates when it reaches a leaf node. In the prefix tree, all leaf nodes are parsing terminals e that represent the end of parse, and all non-leaf nodes represent terminal symbols (*i.e.*, actions). The probability of expanding a non-leaf node is the prefix probability, *i.e.*, how likely is the current path being the prefix of the action sequence. The probability of reaching a leaf node (parsing terminal e) is the parsing probability, *i.e.*, how likely is the current path to the last non-leaf node being the executed actions and the next action. In other words, the parsing probability measures the probability that the last non-leaf node in the path will be the next action to execute. It is important to note that this prefix probability is computed based on both the grammar prior and the haptic prediction; in contrast, in the pure symbolic planner, the prefix probability is computed based on only the grammar prior. An example of the computed prefix and parsing probabilities and output of GEP is given by Figure 2.35, and the search process is illustrated in Figure 2.40. For an algorithmic description of this process, see algorithm 1.

The original GEP is designed for offline video processing. In this work, we made modifications to enable online planning for a robotic task. The major difference between parsing and planning is the uncertainty about past actions: there is uncertainty about observed actions during parsing. However, during planning, there is no uncertainty about executed actions—the robot directly chooses which actions to execute, thereby removing any ambiguity regarding which action was executed at a previous timestep. Hence, we need to prune the impossible parsing results after executing each action; each time after executing an action, we change the probability vector of that action to a one-hot vector. This modification effectively prunes the action sequences that contain the impossible actions executed thus far by the robot.

2.4.4.2 Tactile Glove

For manipulation tasks that require reasoning about latent forces, demonstrations that contain solely visual information (*e.g.*, RGB videos) are insufficient for learning. Using a glove-based system to capture hand-related data has long been proposed; however, it remains an active research topic due to the high articulation and degrees-of-freedom of a human hand. Conventionally, a network of IMUs measures finger poses, but capturing haptic signals is challenging due to hand deformation and a scarcity of force sensing hardware. In this work, we use the tactile glove developed in [LXM17]. The glove uses IMUs to obtain the relative poses of finger phalanges with respect to the wrist and develop a customized force sensor using a soft piezoresistive material (Velostat) whose resistance changes under pressure; see more hardware details in section 2.1.

2.4.4.3 Robot Platform

We evaluate the learned model on a dual-armed 7-DoF Baxter robot mounted on a DataSpeed mobility base. The robot is equipped with a ReFlex TakkTile gripper on the right wrist and a Robotiq S85 parallel gripper on the left. The grippers have minimal haptic sensing capability; they can only determine whether or not the gripper is in contact with an object. Therefore, further force data on the robot is obtained from the 6-degree force and torque sensors located in Baxter's wrists. In addition, a Kinect One sensor is integrated for object pose estimation and tracking. The entire system runs on ROS, and the arm motion is planned by *MoveIt!*.

2.4.4.4 Human Experiment Details and Demographics

Human participants were recruited from the University of California, Los Angeles (UCLA) Department of Psychology subject pool and were compensated with course credit for their participation. A total of 163 students were recruited, each randomly assigned to one of the five experimental groups. Thirteen participants were removed from the analysis for failing to understand the haptic display panel by not passing a recognition task. Hence, the analysis included 150 participants (mean age of 20.7). The symbolic and haptic explanation panels were generated as described in subsubsection 2.4.2.2. The text explanation was generated by the authors based on the robot's action plan to provide an alternate text summary of robot behavior. Although such text descriptions were not directed yielded by the model, they could be generated by modern natural language processing methods.

The human experiment included two phases: familiarization and prediction. In the familiarization phase, participants viewed two videos showing a robot interacting with a medicine bottle, with one successful attempt of opening the bottle and a failure attempt without opening the bottle. In addition to the RGB videos showing the robot's executions, different groups viewed the different forms of explanation panels. At the end of familiarization, participants were asked to assess how well they trusted/believed the robot possessed the ability to open the medicine bottle; see subsubsection 2.4.5.5 and Figure 2.41 for the illustration of the trust rating question.

Next, the prediction phase presented all groups with only RGB videos of a successful robot execution; no group had access to any explanatory panels. Specifically, participants viewed videos segmented by the robot's actions; for segment i, videos start from the beginning of the robot execution up to the ith action. For each segment, subjects were asked to predict what action the robot would execute next; see subsubsection 2.4.5.5 and Figure 2.42 for an illustration of the action prediction question.

Regardless of group assignment, all RGB videos were the same across all groups; *i.e.*, we show the same RGB video for all groups with varying explanation panels. This experimental design isolates potential effects of execution variations in different robot execution models presented in subsubsection 2.4.2.1; we only seek to evaluate how well explanation panels foster qualitative trust and enhance prediction accuracy and keep robot execution performance constant across groups to remove potential confounding.

For both qualitative trust and prediction accuracy, the null hypothesis is that the explanation panels foster equivalent levels of trust and yield the same prediction accuracy across different groups, and therefore no difference in trust or prediction accuracy would be observed. The test is a two-tailed independent samples t-test to compare performance from two groups of participants, as we used between-subjects design in the study, with a commonly used significance level $\alpha = 0.05$, assuming t-distribution, and the rejection region is $p < 0.05$.

2.4.5 Supplementary Materials

2.4.5.1 Additional Model Results

Figure 2.36 presents additional generalization experiment results using the improved GEP. Each augmented 3D-printed cap generates a significantly different time-series haptic signal, indicating the haptic interactions are substantially different from one another. These results demonstrate the GEP's ability to transfer to bottles with haptic signals that are different from the ones in the demonstration.

We qualitatively analyze one example in Figure 2.37 to further justify that the generalization scenarios are significantly different from the testing scenarios by comparing the haptic signals between the two. Specifically, given one haptic signal in testing and one in generalization performing the very same action, we treat these two sets of haptic signals as time series data and estimate the trend using kernel methods. After obtaining the trend, a rigorous testing procedure [ZW11] for evaluating whether these two haptic signals have the same distribution is performed by comparing the L2 distance between the curves. The results indicate that the haptic signals of all bottles are significantly different from one another; Δ is not closed to 0. A comprehensive, quantitative analysis of more haptic signal data is presented as a confusion matrix in Figure 2.38).

We also note that an alternative model commonly used for this type of analysis is the ARIMA method. However, our observations (haptic signals) are mean non-stationary, which is not suitable for ARIMA; ARIMA works well for an integrated process.

2.4.5.2 Additional Materials and Methods

Model Limitations Generating a smooth action sequence mapped from a human demonstration to a robot is a non-trivial task. In this work, for the symbolic planner, we simplified this process by assuming each mapped action is executable by the Baxter robot; each atomic action or motion primitive (terminal node in the grammar) is designed, not automatically learned. However, this should not impact the overall contributions presented in the work, as the mapping in a supervised fashion does not affect the experiments for evaluating human trust. In addition, this embodiment problem is solvable in some instances if we introduce the concept of "mirroring" [LZZ19].

Our approach assumes that each robot action corresponds to an equivalent human action. However, if adopted after learning the grammar, trajectory optimization methods (*e.g.*, CHOMP [RZB09], STOMP [KCT11], and TrajOpt [SHL13]) could improve the action/behavior of the robot to generate smoother action sequences, or even produce different actions that are not the same as the human demonstrations. In addition, a grammar can, in fact, generate sequences that are not seen in demonstrations because of the compositional nature of the grammar rules; in other words, it is possible for the robot to solve the tasks using different action sequences from human sampled from the grammar model. Nevertheless, a grammar has no inherent mechanism for the robot to discover entirely new actions for the task.

2.4.5.3 Training Details of Embodied Haptic Model

In this section, we present the implementation detail for reproducibility.

Network Architecture The autoencoder is constructed with a multi-layer perceptron (MLP); see Table 2.10. The human embedding can be obtained with a forward pass through the network. The supervision for the autoencoder is the original human post-condition. The loss is measured by the reconstruction error. The robot-human embodiment mapping is implemented with an MLP; see Table 2.11. The embodiment mapping is trained using equivalent human and robot post-conditions (equivalent here means the post-condition of executing the same action successfully). The human post-condition is fed through the autoencoder to produce a human embedding, and this embedding serves as the supervision target for the embodiment mapping network. The last major component of the embodied haptic prediction model is the action predictor, also implemented with an MLP; see Table 2.12. The supervision for the action predictor is the ground-truth human action labels.

Training Details We adopt a two-step updating schema for the embodied haptic model. In the first step, we feed forward the human post-condition data into the autoencoder. The encoder will reduce the high-dimensional human data to a low-dimensional human embedding; the encoder and the decoder are learned with hyper-parameter shown in Table 2.13. The supervision for the autoencoder is the reconstructed original human post-condition. In the second step, with the human embedding and the action labels, the action predictor and the embodiment mapping are training jointly with the hyper-parameters shown in Table 2.13. The embodiment mapping is trained using equivalent human and robot post-conditions (equivalent here means the post-condition of executing the same action successfully). The human post-condition is fed through the autoencoder to produce a human embedding, and this embedding serves as the supervision target for the embodiment mapping network. The supervision for the action predictor is the ground-truth human action labels.

2.4.5.4 Force Visualization

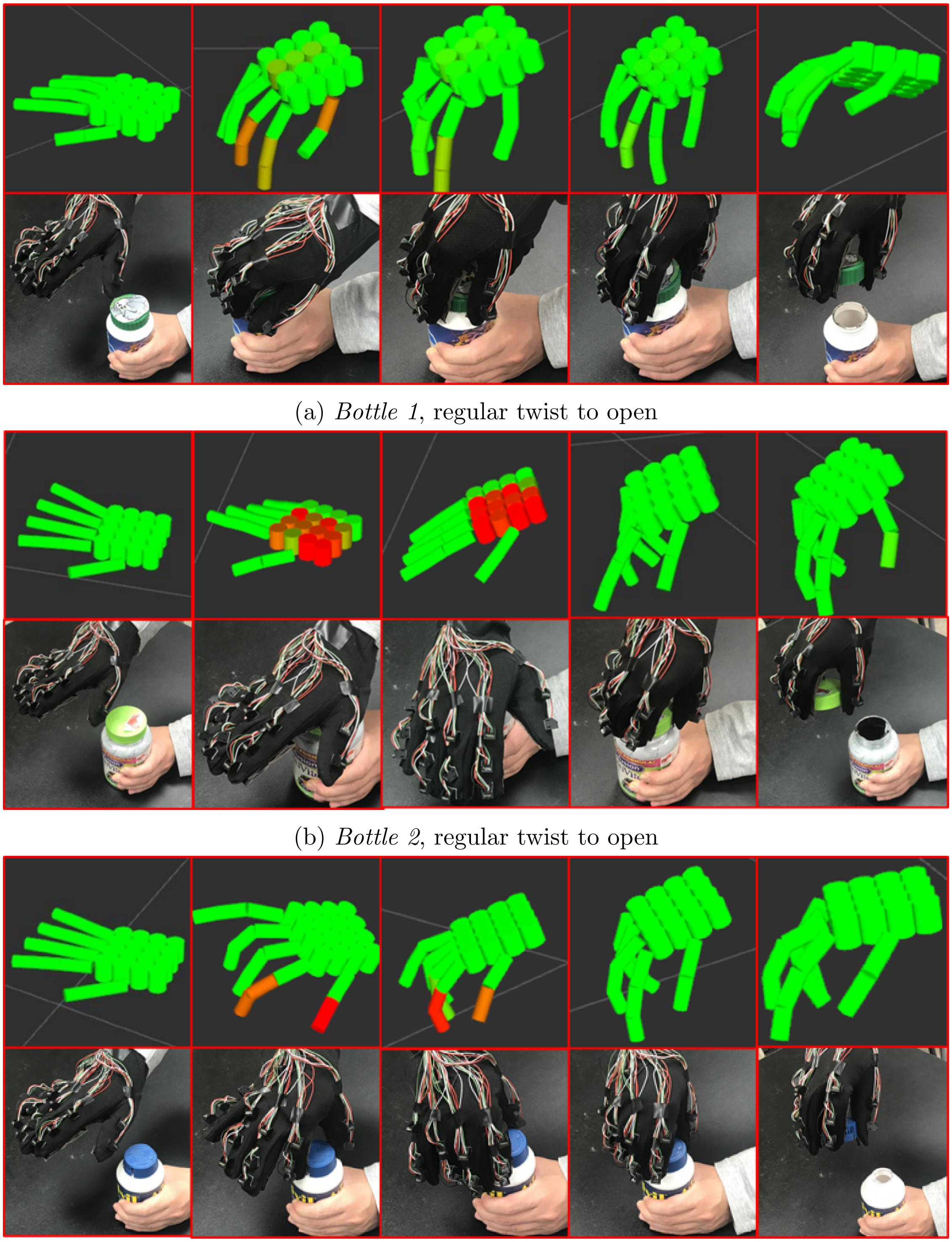
To visualize the forces imposed by the robot gripper, we first identify the max force magnitude in all the force signal data collected from human demonstrations. Then, all force data is normalized to the value between 0 and 1, where 0 corresponds to pure green in the visualization, and 1 pure red. The value in between is interpolated linearly and displayed on the robot's palm.

2.4.5.5 Additional Human Experiment Details

The prediction phase evaluates how well each explanation panel imparts prediction ability after observing a robot's behaviors in solving the problem of opening a medicine bottle. Note that during the familiarization phase, the robot explains its behavior through explanatory panels, but during the prediction phase, subjects observe the robot executing the task with only the RGB videos. Thus our prediction question asks "*after familiarizing with explanatory panels, how well are human subjects able to predict robot behavior when observing only RGB robot executions?*" The prediction accuracy is computed as the percentage of correct action predictions in the sequence. This experimental design examines how well each explanatory panel imparts prediction ability under new robot executions where no explanation panel is available. For each question, participants selected from 8 actions: *push on the cap*, *pinch the cap*, *pull the cap*, *twist the cap*, *grasp the cap*, *ungrasp the cap*, *move the left robot arm to grasping position*, and *nothing.*

	Qualitative Trust		**Prediction Accuracy**	
	Mean	Std. dev.	Mean	Std. dev.
Baseline	71.7	16.8	0.481	0.176
GEP	82.6	17.5	0.644	0.202
Symbolic	81.9	17.4	0.641	0.228
Haptic	75.7	16.2	0.541	0.231
Text	70.1	22.7	0.593	0.218

Table 2.9: Numerical results and standard deviations for human subject study; the same data was used in Figure 2.34.



(a) *Bottle 1*, regular twist to open

(b) *Bottle 2*, regular twist to open

(c) *Bottle 3*, regular twist to open

Figure 2.11: Action sequences and visualizations of opening three types of bottles

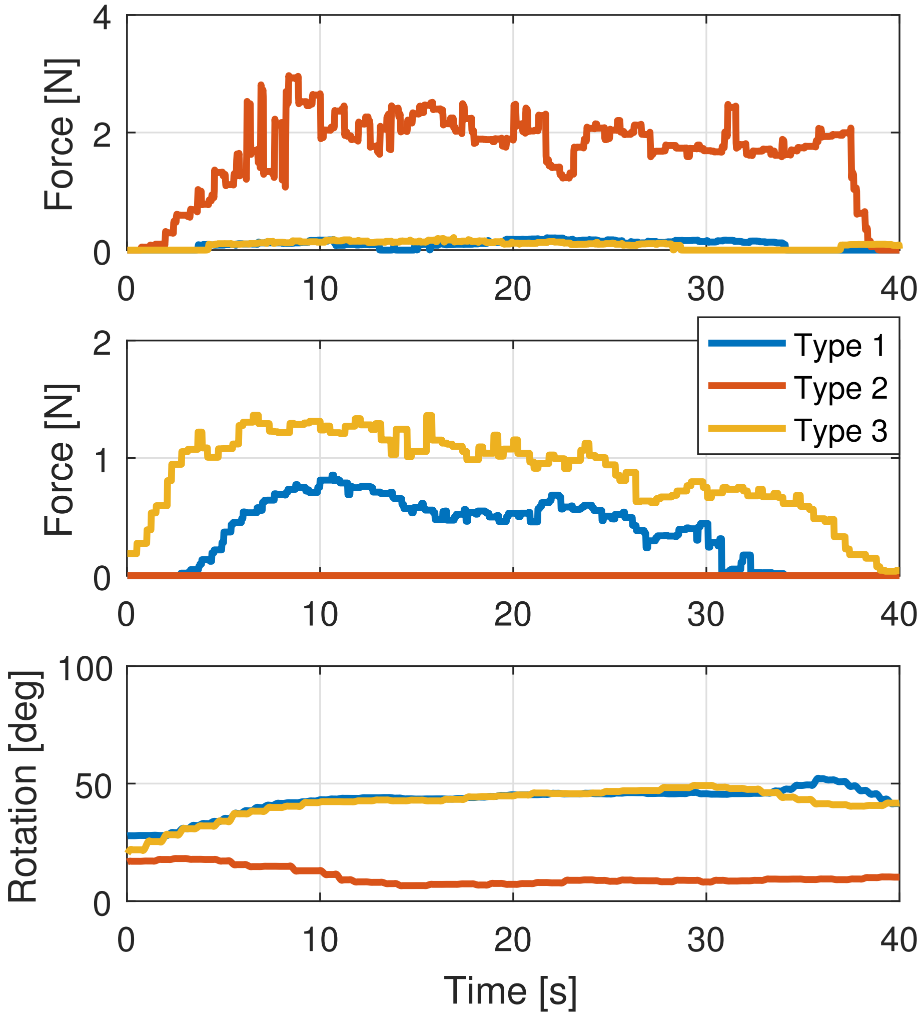


Figure 2.12: Force signals captured in palm (top) and the fingertip of thumb (middle), and flexion angle of index finger's MCP joint (bottom).

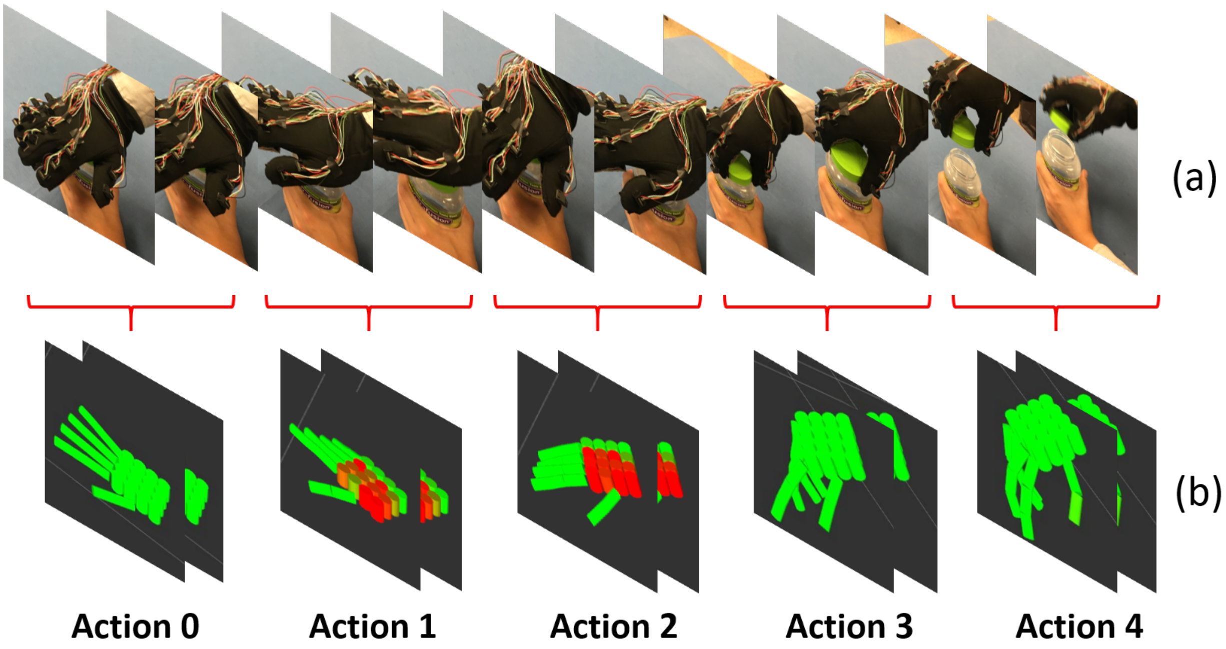


Figure 2.13: (a) A sequence of movement primitive demonstrated by an agent for a manipulation task–opening a medicine bottle captured by a tactile glove. (b) Reconstructed force and pose data using the tactile glove. Our purposed method segments and parses the noisy inputs of force and pose in an unsupervised fashion.

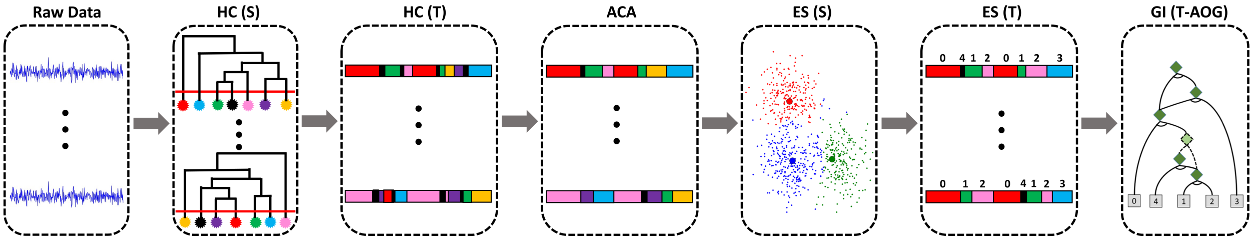


Figure 2.14: Unsupervised learning pipeline of hand-object motion recognition. After collecting the raw data using a tactile glove, a spatial (HC (S)) and temporal (HC (T)) hierarchical clustering is performed on both force and pose data. An aligned cluster analysis (ACA) is adopted to further reduce the noise. Event segmentation (ES (S) and ES (T)) is achieved by merging motion primitives based on the distance measured by DTAK. Finally, a grammar is induced (GI) based on the segmented events, forming a T-AOG.

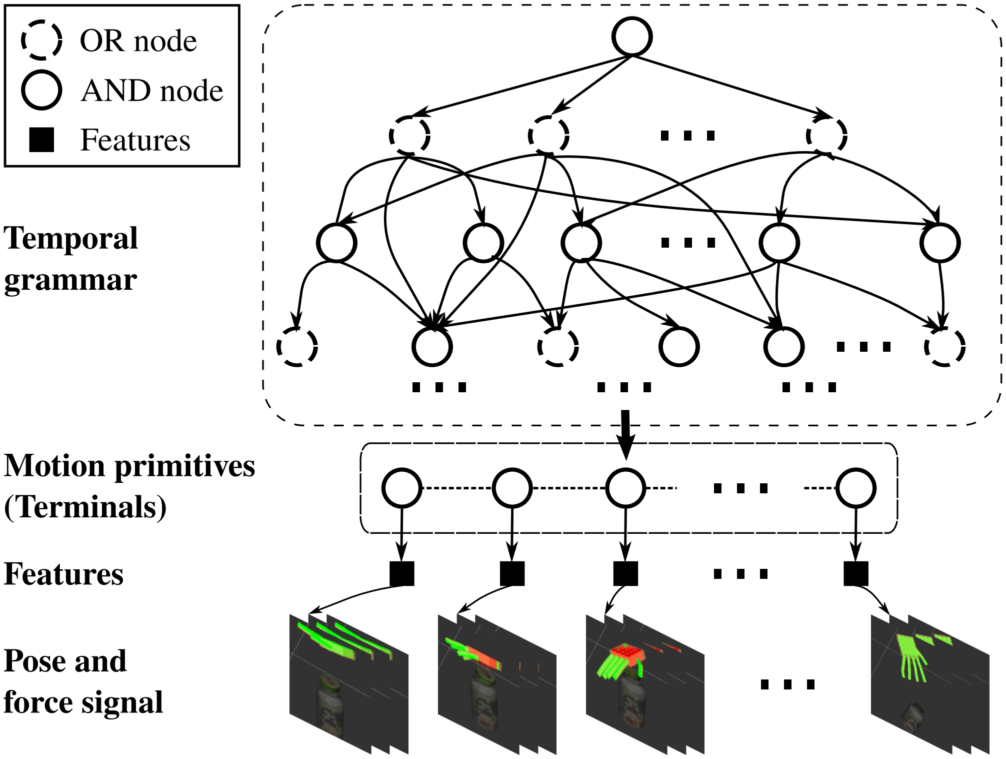


Figure 2.15: Illustration of the T-AOG. The T-AOG is a temporal grammar in which the terminal nodes are motion primitives of hand-object interactions.

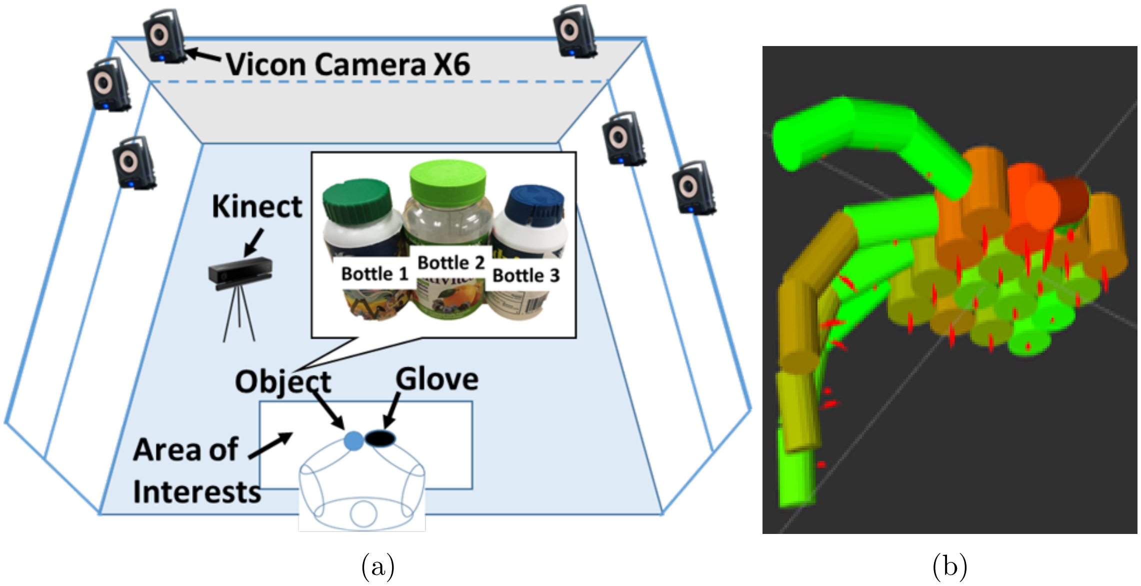


Figure 2.16: (a) The experimental setup for data collection. We use Vicon system to obtain the poses of human's wrist and object's parts. The camera is used to record the data collection procedure. (b) Visualization of force vectors, which contains both pose and force features.

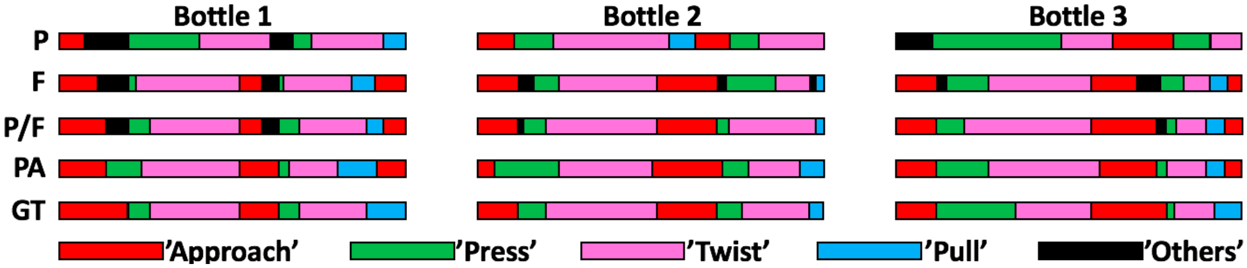


Figure 2.17: Qualitative evaluation. Event segmentation and recognition of opening Bottle 1, 2, and 3, from left to right, respectively. P denotes *pose only feature*, F *force only feature*, P/F *force vector feature*, PA *with parsing*, and GT *ground truth*. Each segment represents one type of motion primitive which color is determined by the ground truth sequence.

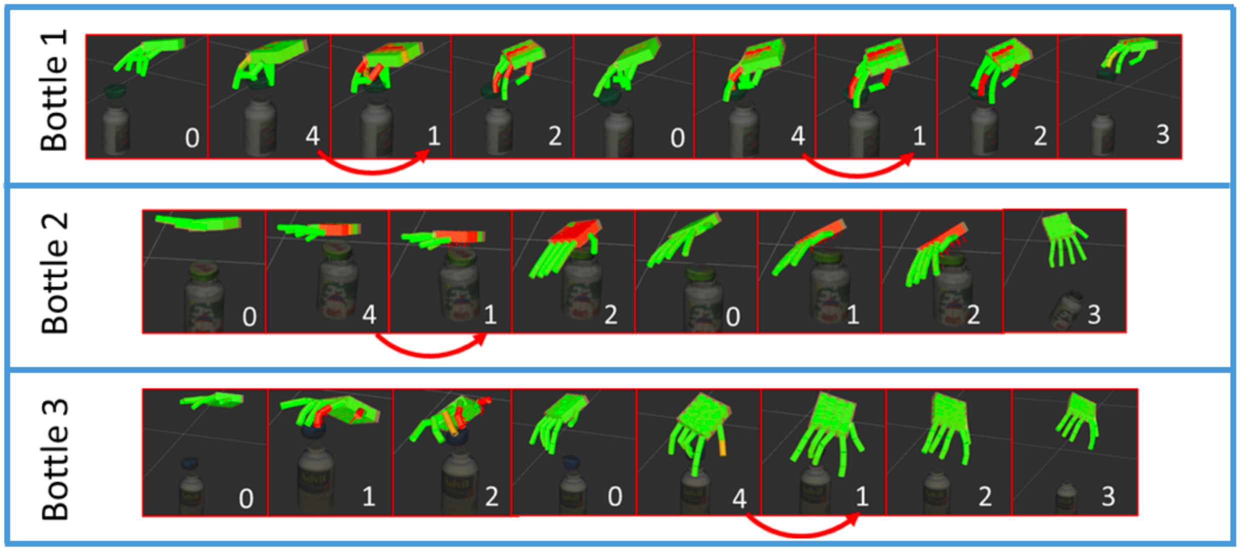
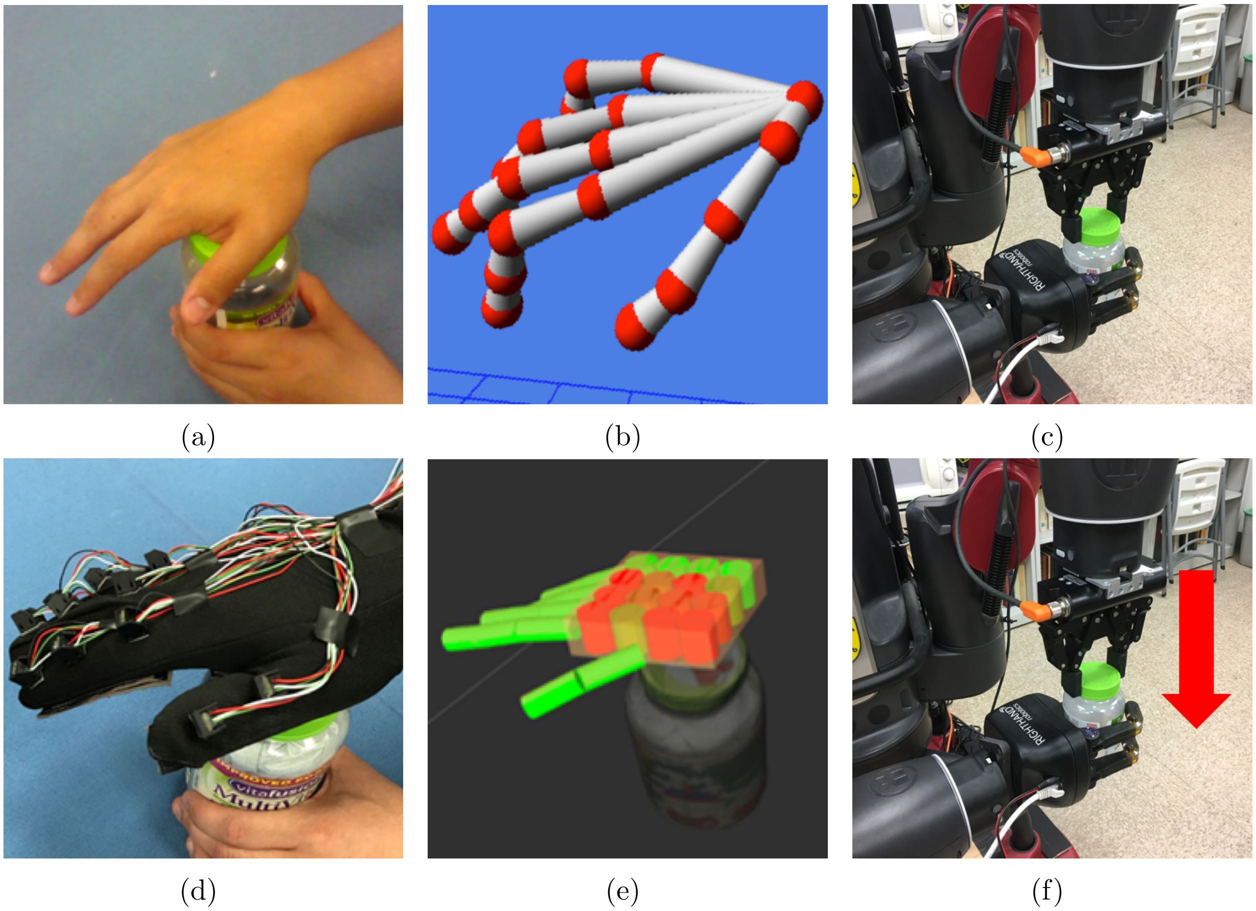


Figure 2.18: Key frames of opening various bottles with T-AOG. The numbers indicate the cluster labels and the red arrows indicate the merges triggered by the parsing of T-AOG.



(a) (b) (c)

(d) (e) (f)

Figure 2.19: Given a RGB-D-based image sequence (a), although we can infer the skeleton of hand using vision-based methods (b), such knowledge cannot be easily transferred to a robot to open a medicine bottle (c), due to the lack of force sensing during human demonstrations. In this work, we utilize a tactile glove (d) and reconstruct both forces and poses from human demonstrations (e), enabling robot to directly observe forces used in demonstrations so that the robot can successfully open a medicine bottle (f).

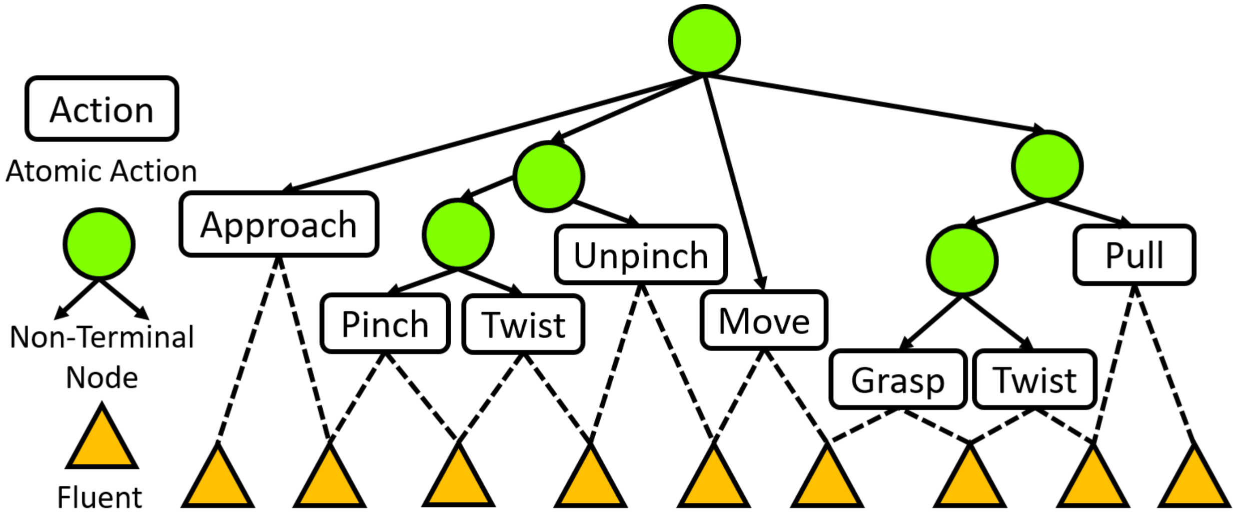


Figure 2.20: An example of grammar parsing with T-AOG. Actions are executed in temporal order from left to right.

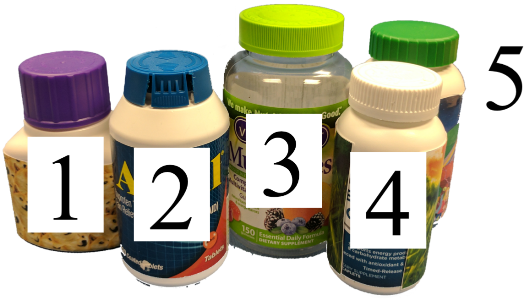


Figure 2.21: Bottles used in experiments with different safety mechanism: (1) *push-and-twist*, (2) *pinch-and-twist*, (3) *push-and-twist*, and (4) *push-and-twist*. (5) Bottle with no safety mechanism.

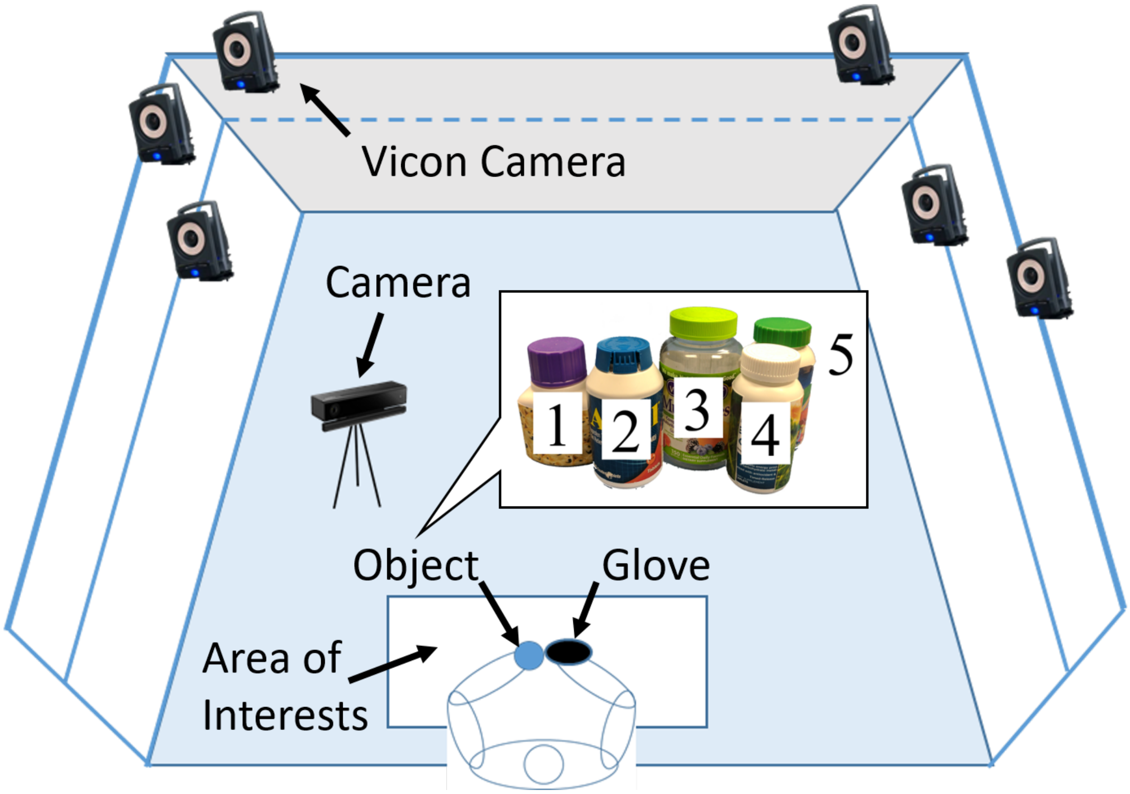


Figure 2.22: We use a Vicon system to obtain the poses of human's wrist and object's parts. The camera is used to record the data collection procedure. The data is collected on bottles (2), (3) and (5), which require *pinch-and-twist*, *push-and-twist* and *twist* to open, respectively.

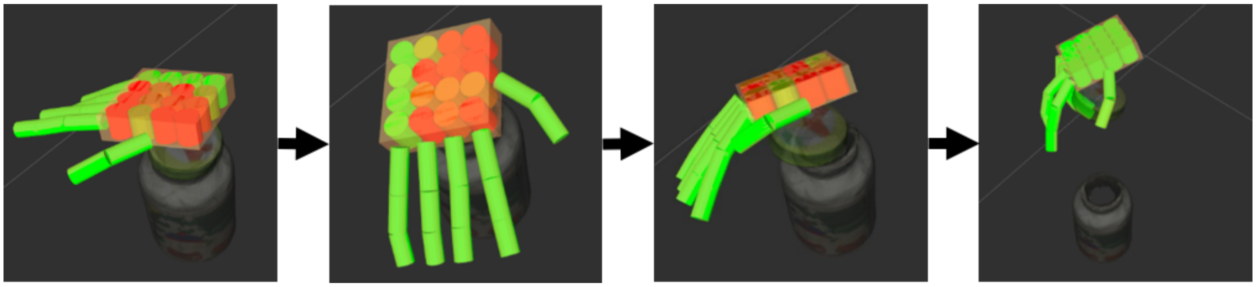


Figure 2.23: The tactile glove computes the pose of human's phalanxes according to the pose of human's wrist and measure the force applied on human's hand.

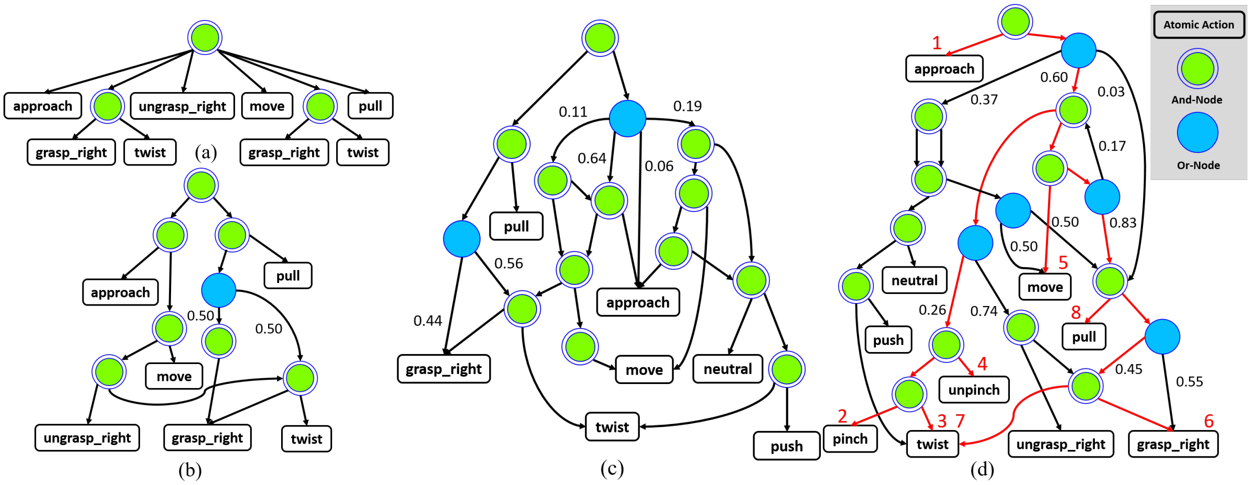


Figure 2.24: AOG induced from human demonstrations using 1 example (a), 5 examples (b), 36 examples (c), and 65 examples (d). (d) also shows Figure 2.20 parsed in an AOG, highlighted in red. Numbers indicate temporal ordering of atomic actions.

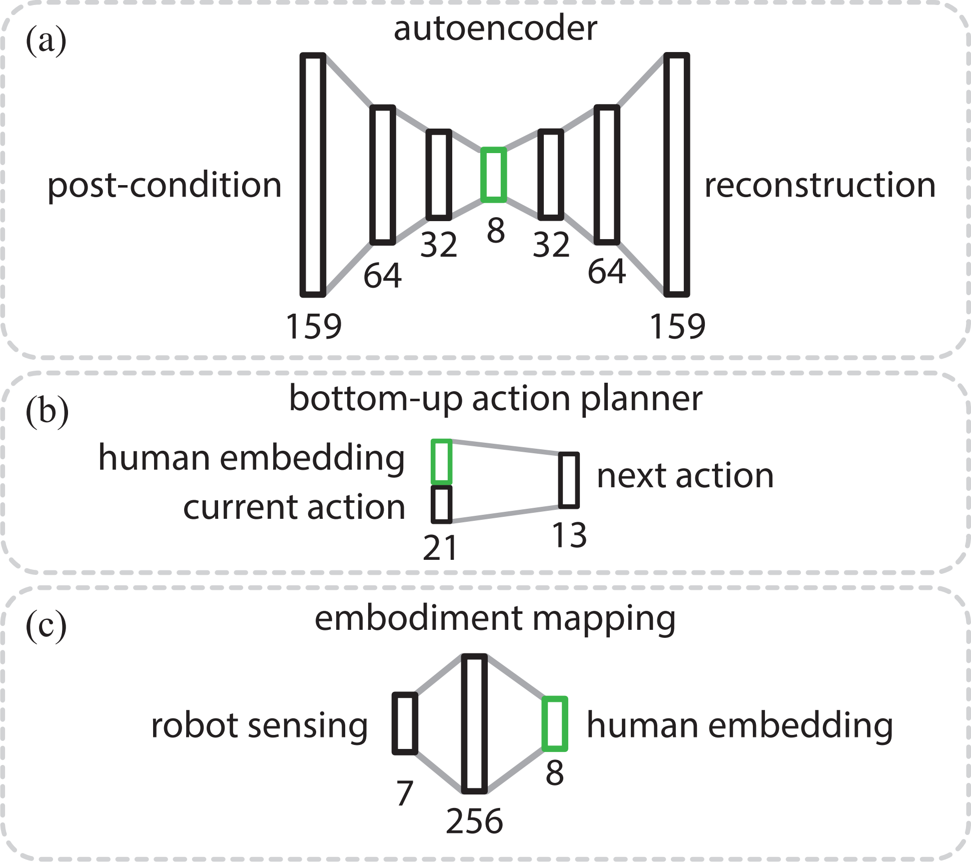


Figure 2.25: (a) Autoencoder to project human demonstration into low-dimensional subspace. (b) Classifier used to plan the next action using a low-dimensional embedding of human tactile feedback. (c) Embodiment mapping used to map robot states to equivalent human demonstration states. Each rectangle represents a vector, and each corresponding number is the length of the vector. The green rectangle represents the low-dimensional human embedding vector.

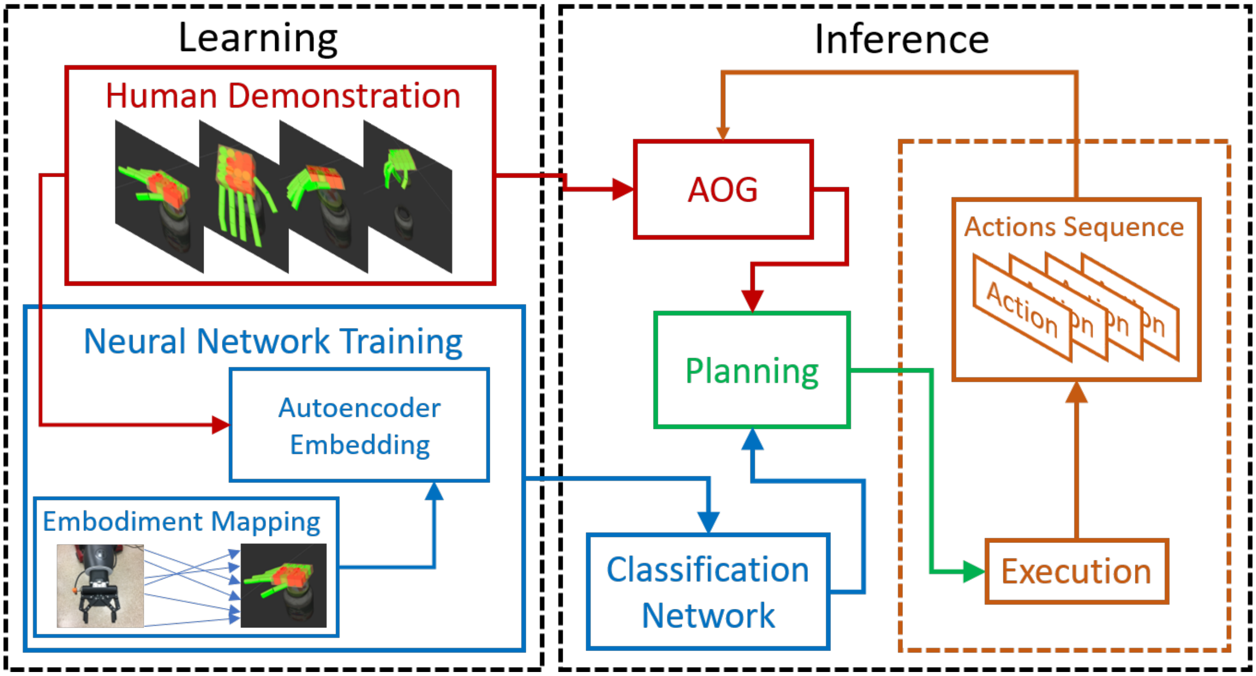
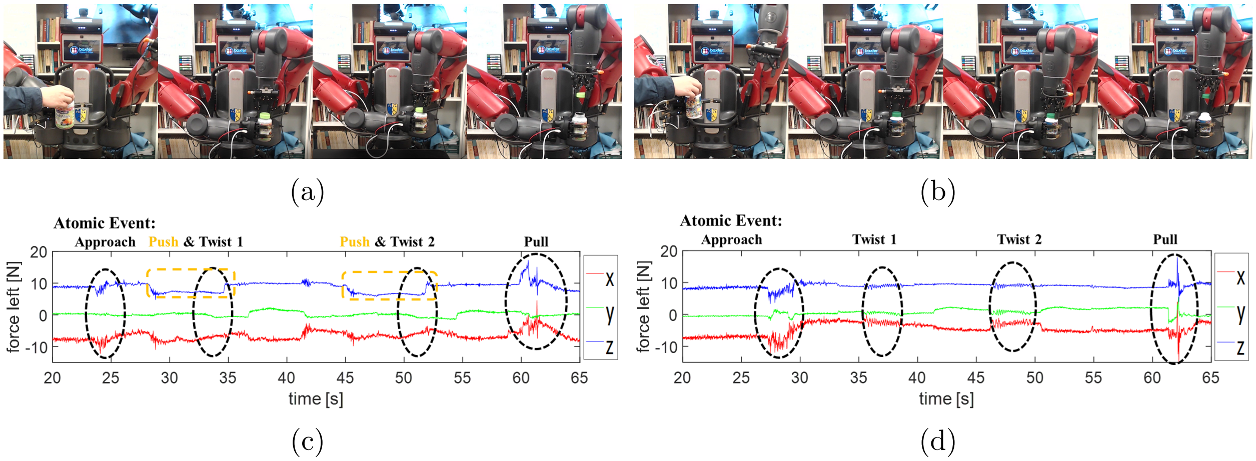


Figure 2.26: System architecture. Blue: action planning using fluents as a bottom-up process. Red: action planning using AOG as a top-down process. Green: action planning. Brown: robot execution.



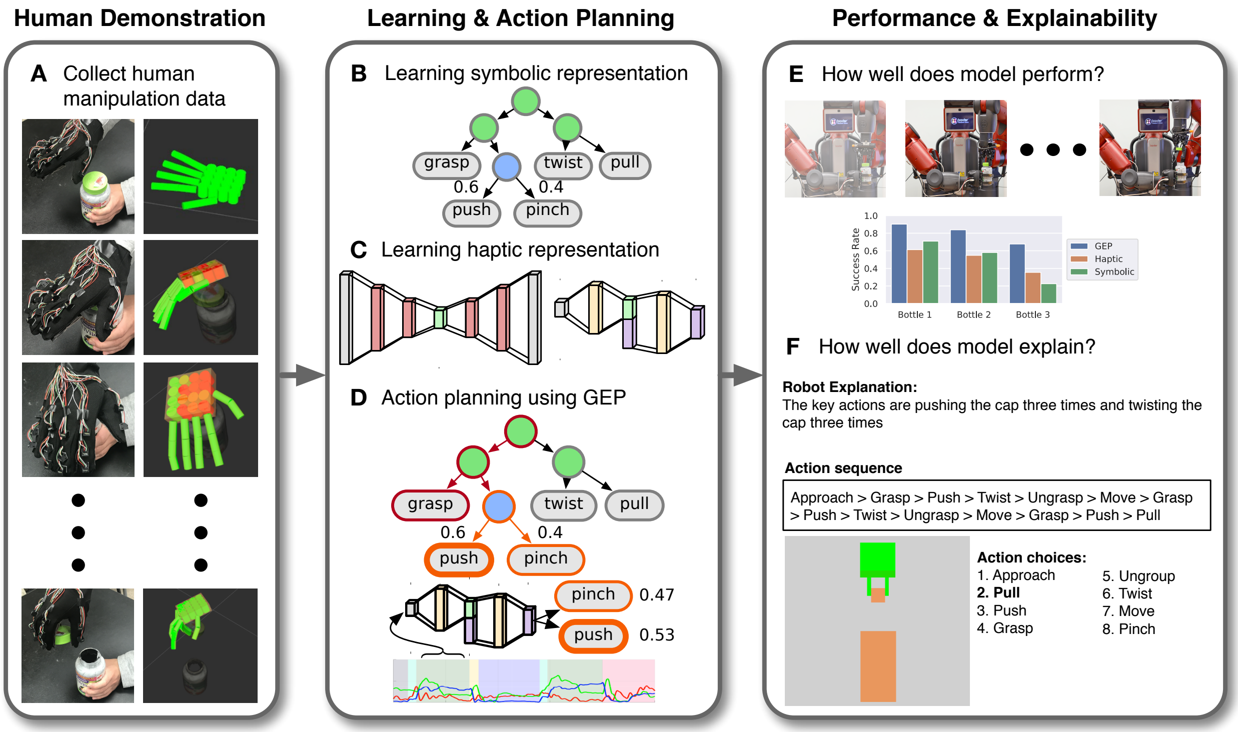
(a)

(b)

(c)

(d)

Figure 2.27: (a) Robot opening bottle 3, showing actions *approach*, *push*, *twist*, and *pull* from left to right. (b) Robot opening bottle 5, showing actions *approach*, *grasp*, *twist*, and *pull*. Force-torque sensor readings while opening bottle 3 (c) and bottle 5 (d), showing clear, distinguishable differences from raw sensor data.


Human Demonstration
Learning & Action Planning
Performance & Explainability
A Collect human manipulation data
B Learning symbolic representation
grasp
twist
pull
0.6
0.4
push
pinch
C Learning haptic representation
D Action planning using GEP
grasp
twist
pull
0.6
0.4
push
pinch
pinch 0.47
push 0.53
E How well does model perform?
1.0
0.8
0.6
0.4
0.2
0.0
Success Rate
Bottle 1
Bottle 2
Bottle 3
GEP
Haptic
Symbolic
F How well does model explain?

Robot Explanation:
The key actions are pushing the cap three times and twisting the cap three times

Action sequence

Approach > Grasp > Push > Twist > Ungrasp > Move > Grasp > Push > Twist > Ungrasp > Move > Grasp > Push > Pull

Action choices:

1. Approach
2. **Pull**
3. Push
4. Grasp
5. Ungroup
6. Twist
7. Move
8. Pinch

Figure 2.28: **Overview of demonstration, learning, evaluation, and explainability.** By observing human demonstrations, the robot learns, performs, and explains using both a symbolic representation and a haptic representation. **(A)** Fine-grained human manipulation data is collected using a tactile glove. Based on the human demonstrations, the model learns **(B)** symbolic representations by inducing a grammar model that encodes long-term task structure to generate mechanistic explanations, and **(C)** embodied haptic representations using an autoencoder to bridge the human and robot sensory input in a common space, providing a functional explanation of robot action. These two components are integrated using the **(D)** GEP for action planning. These processes complement each other in both **(E)** improving robot performance and **(F)** generating effective explanations that foster human trust.

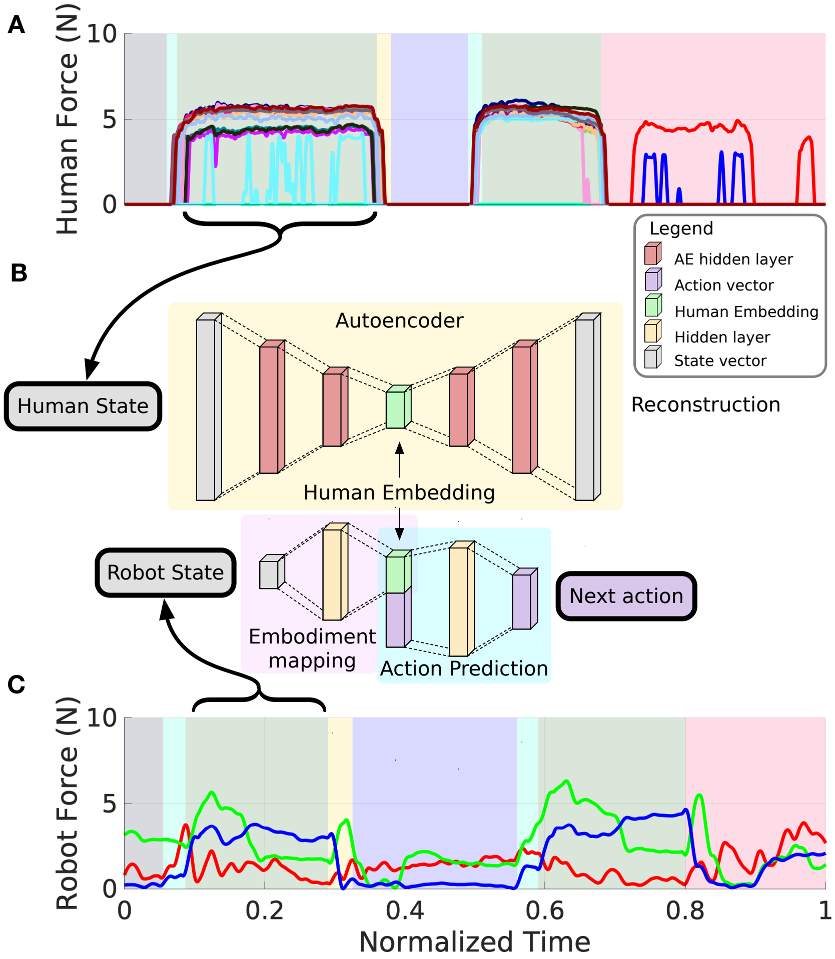


Figure 2.29: **Illustration of learning embodied haptic representation and action prediction model.** An example of the force information in (**A**) the human state, collected by the tactile glove (with 26 dimensions of force data), and force information in (**C**) the robot state, recorded from the force sensors in the robot's end-effector (with 3-dimensions of force data). The background colors indicate different action segments. For equivalent actions, the human and the robot may take a different amount of time to execute, resulting in different action segment lengths. (**B**) Embodied haptic representation and action prediction model. The autoencoder (yellow background) takes a human state, reduces its dimensionality to produce a human embedding, and uses the reconstruction to verify that the human embedding maintains the essential information of the human state. The embodiment mapping network (purple background) takes in a robot state and maps to an equivalent human embedding. The action prediction network (light blue background) takes the human embedding and the current action and predicts what action to take next. Thus, the robot imagines itself as a human based on its own haptic signals and predicts what action to take next.

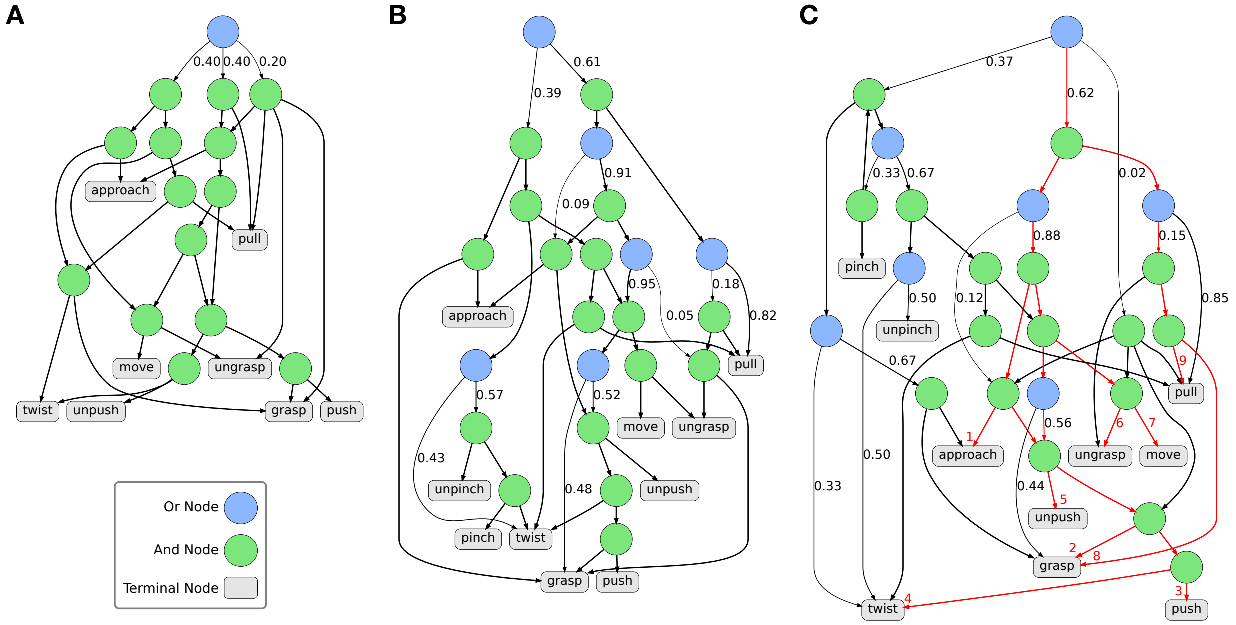


Figure 2.30: **An example of action grammar induced from human demonstrations.** Green nodes represent And-nodes, and blue nodes represent Or-nodes. Probabilities along edges emanating from Or-nodes indicate the parsing probabilities of taking each branch. Grammar model induced from (**A**) 5 demonstrations, (**B**) 36 demonstrations, (**C**) 64 demonstrations. The grammar model in (**C**) also shows a parse graph highlighted in red, where red numbers indicate temporal ordering of actions.

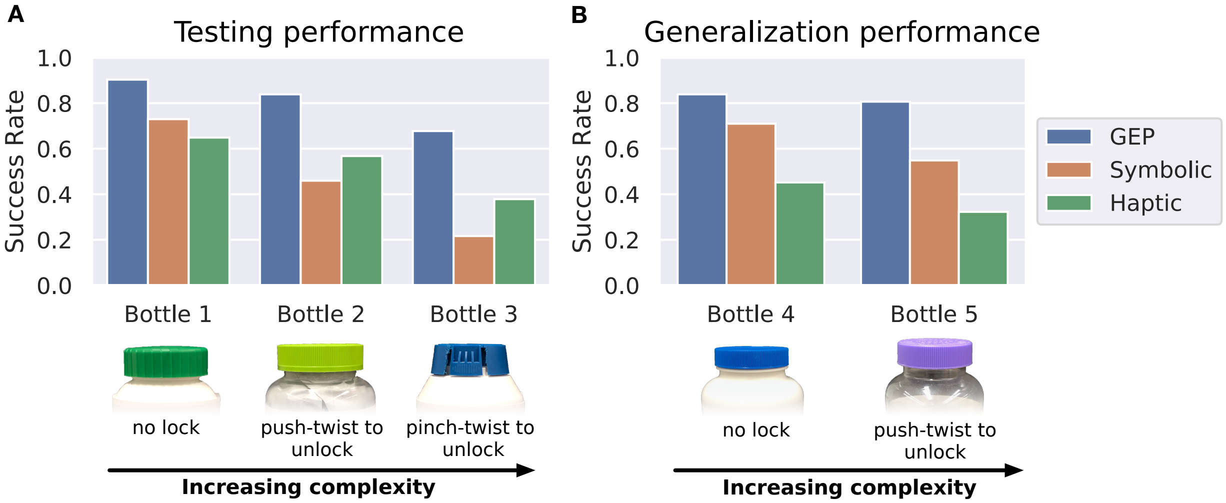


Figure 2.31: **Robot task performance on different bottles with various locking mechanisms using the symbolic planner, haptic model, and the GEP that integrates both.** (**A**) Testing performance on bottles observed in human demonstrations. Bottle 1 does not have a locking mechanism, Bottle 2 employs a *push-twist* locking mechanism, and Bottle 3 employs a *pinch-twist* locking mechanism. (**b**) Generalization performance on new, unseen bottles. Bottle 4 does not have a locking mechanism, and Bottle 5 employs a *push-twist* locking mechanism. The bottles used in generalization have similar locking mechanisms but evoke significantly different haptic feedback; see subsubsection 2.4.5.1. Regardless of testing on demonstration or unseen bottles, the best performance is achieved by the GEP that combines the symbolic planner and haptic model.

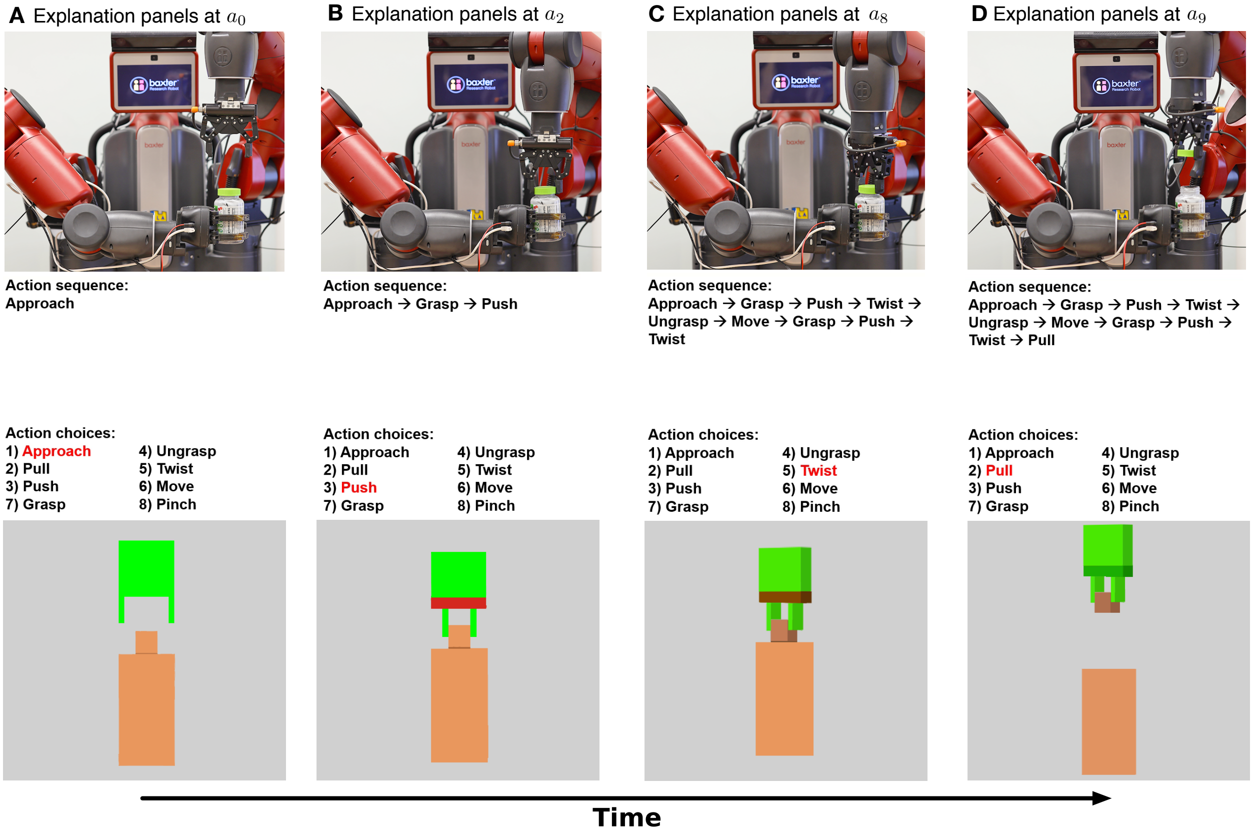
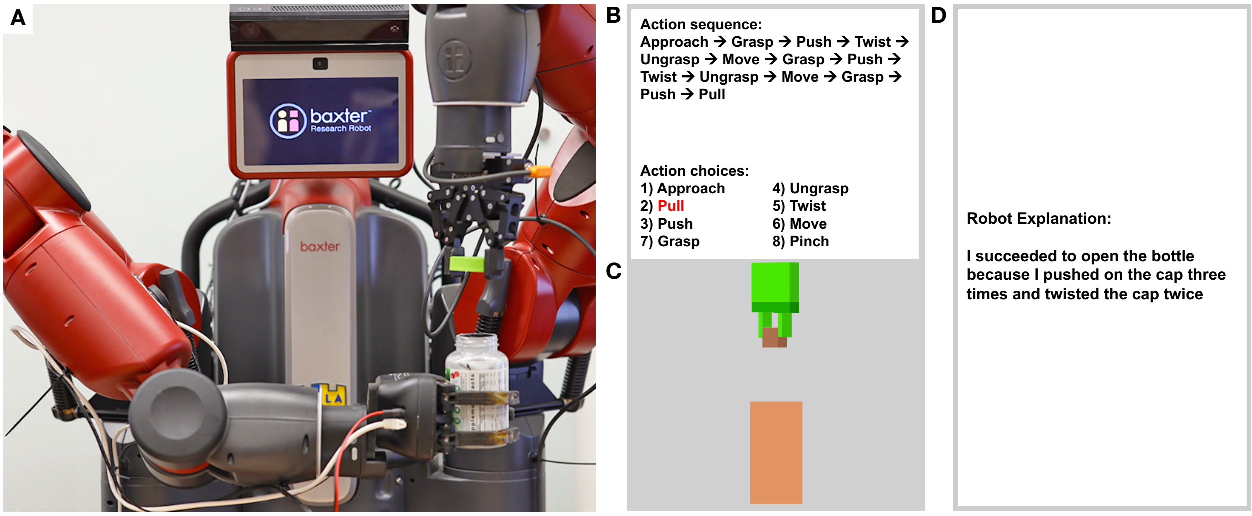


Figure 2.32: **Explanations generated by the symbolic planner and the haptic model.** (**A**) Symbolic (mechanistic) and haptic (functional) explanations at a_0 of the robot action sequence. (**B**), (**C**), and (**D**) show the explanations at times a_2, a_8, and a_9, where a_i refers to the ith action. Note that the red on the robot gripper's palm indicates a large magnitude of force applied by the gripper, and green indicates no force; other values are interpolated. These explanations are provided in real-time as the robot executes.



Action sequence:
Approach → Grasp → Push → Twist →
Ungrasp → Move → Grasp → Push →
Twist → Ungrasp → Move → Grasp →
Push → Pull

Action choices:
1) Approach
2) Pull
3) Push
7) Grasp
4) Ungrasp
5) Twist
6) Move
8) Pinch

Robot Explanation:

I succeeded to open the bottle
because I pushed on the cap three
times and twisted the cap twice

E Summary of human subject groups and explanations presented

Group	RGB (A)	Symbolic (B)	Haptic (C)	Text (D)
Baseline	✓			
Symbolic	✓	✓		
Haptic	✓		✓	
GEP	✓	✓	✓	
Text	✓			✓

Figure 2.33: **Illustration of visual stimuli used in human experiment**. All five groups observed the RGB video recorded from robot executions, but differed by the access to various explanation panels. (**A**) RGB video recorded from robot executions. (**B**) Symbolic explanation panel. (**C**) Haptic explanation panel. (**D**) Text explanation panel. (**E**) A summary of which explanation panels were presented to each group.

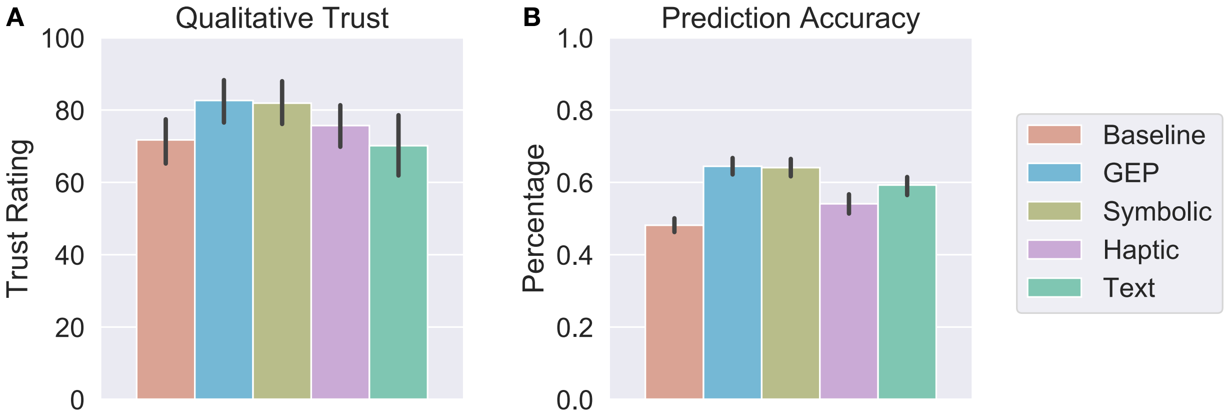


Figure 2.34: **Human results for trust ratings and prediction accuracy.** (**A**) Qualitative measures of trust: average trust ratings for the five groups. and (**B**) Average prediction accuracy for the five groups. The error bars indicate the 95% confidence interval. Across both measures, the GEP performs the best. For qualitative trust, the text group performs most similarly to the baseline group. For a tabular summary of the data, see Table 2.9

A Input probability matrix

Time step	*grasp*	*push*	*pinch*	*twist*	*pull*
0	0.6	0.1	0.1	0.1	0.1
1	0.6	0.1	0.1	0.1	0.1
2	0.1	0.1	0.6	0.1	0.1
3	0.1	0.1	0.6	0.1	0.1
4	0.1	0.1	0.1	0.6	0.1
5	0.1	0.1	0.1	0.6	0.1

B Cached probabilities

Time step	ϵ	*grasp*	*grasp push*	*grasp pinch*	*grasp push twist*	*grasp pinch twist*	*grasp push twist pull*	*grasp pinch twist pull*
0	0.000	0.600	0.000	0.000	0.000	0.000	0.000	0.000
1	0.000	0.360	0.036	0.024	0.000	0.000	0.000	0.000
2	0.000	0.036	0.025	0.101	0.004	0.002	0.000	0.000
3	0.000	0.004	0.005	0.069	0.003	0.010	3.6e-04	2.4e-04
4	0.000	3.6e-04	6.8e-04	0.007	0.005	0.048	3.2e-04	0.001
5	0.000	3.6e-05	9.0e-05	7.2e-04	0.003	0.033	4.9e-04	0.005
prefix	1.000	0.600	0.060	0.119	0.009	0.058	0.001	0.006

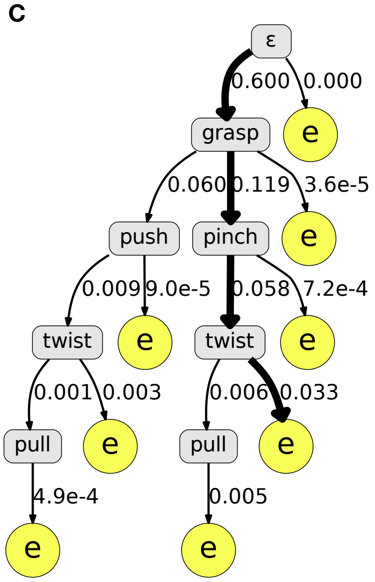


Figure 2.35: An example of the Generalized Earley Parser (GEP). (**A**) A classifier is applied to a 6-frame signal and outputs a probability matrix as the input. (**B**) A table of the cached probabilities of the algorithm. For all expanded action sequences, it records the parsing probabilities at each time step and prefix probabilities. (**C**) Grammar prefix tree with the classifier likelihood. The GEP expands a grammar prefix tree and searches in this tree. It finds the best action sequence when it hits the parsing terminal *e*. It finally outputs the best label "grasp, pinch, pull" with a probability 0.033. The probabilities of children nodes do not sum to 1 because grammatically incorrect nodes are eliminated from the search and the probabilities are not re-normalized [QJZ18].

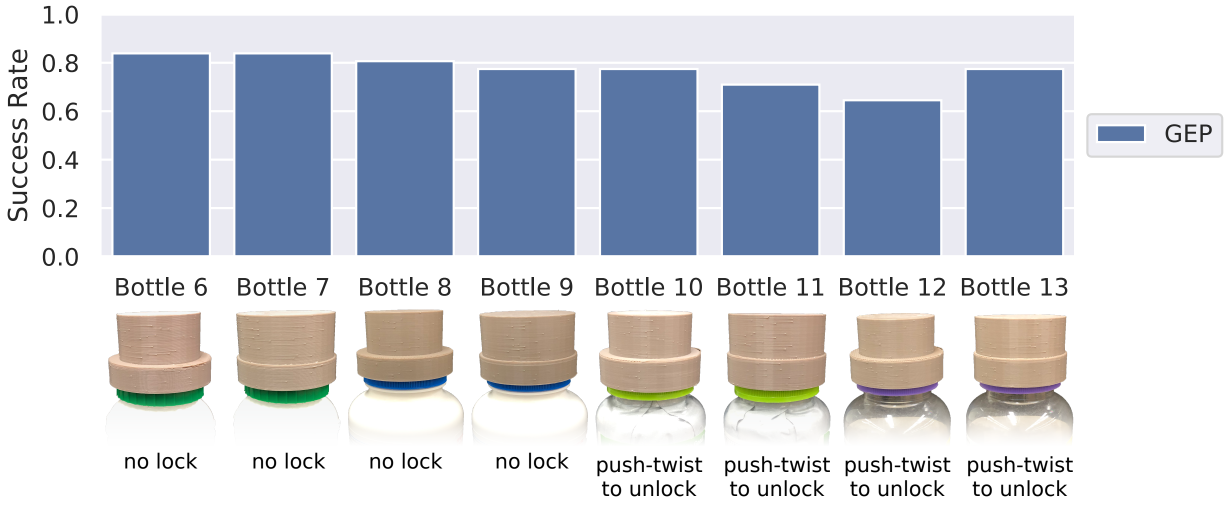


Figure 2.36: Additional generalization experiments on bottles augmented with different 3D-printed caps. The GEP shows good performance across all bottles, indicating the GEP is able to generalize to bottles with similar locking mechanisms as in the human demonstrations, but significantly different haptic signals.

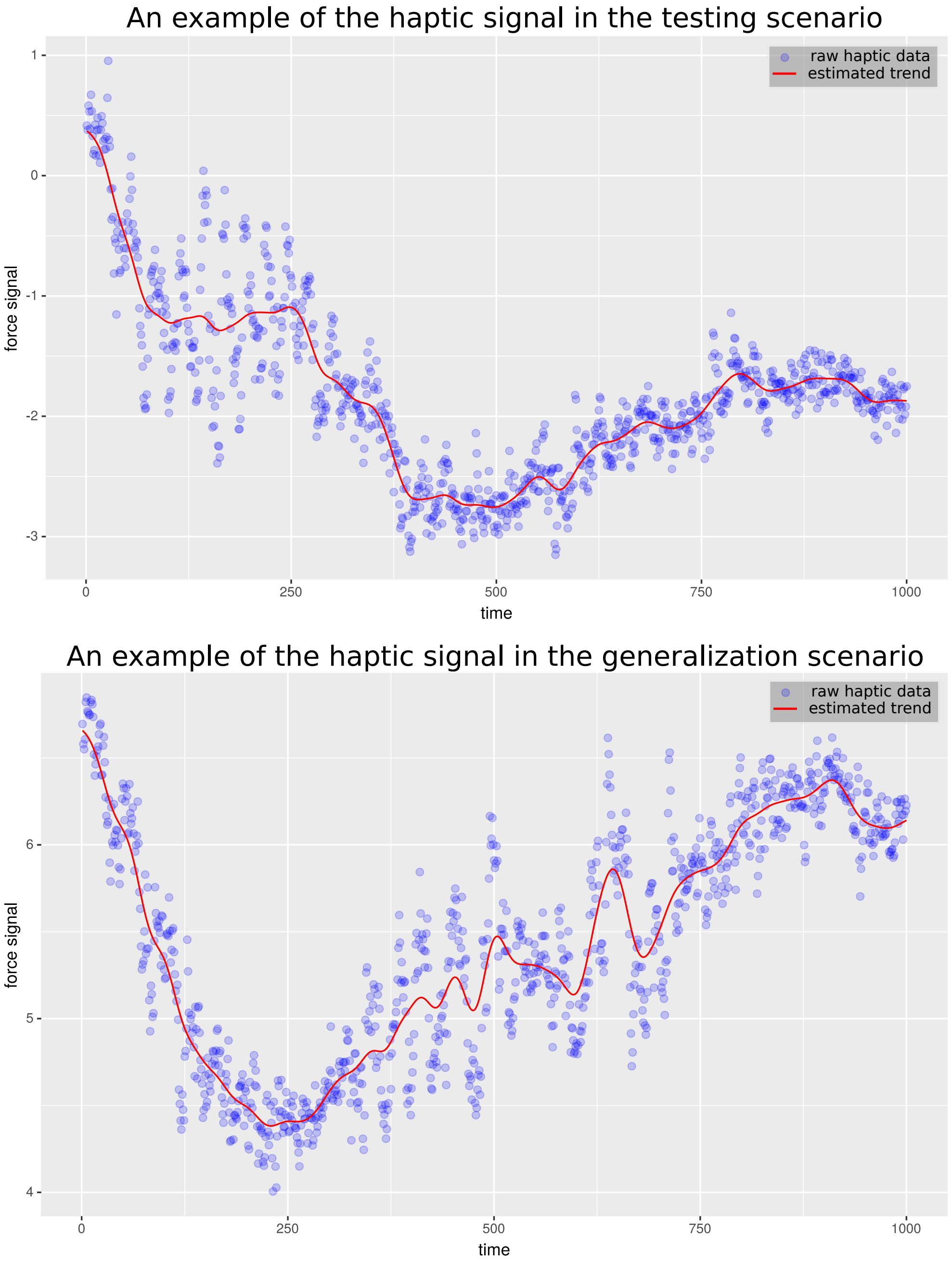
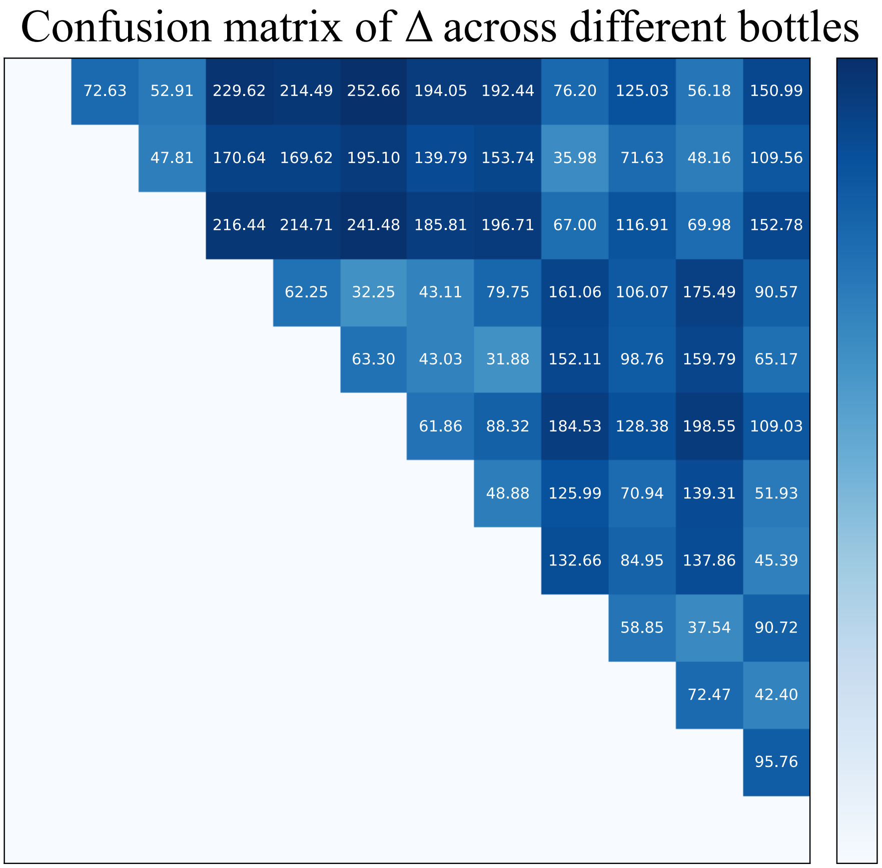


Figure 2.37: Examples of aligned haptic signals in time used in testing (top) and generalization (down) data. The haptic data was collected by executing the same actions on various bottles in testing and generalization scenarios. Light blue dots denotes raw noisy haptic signals. The solid red line denotes the estimated trend for statistical analysis.



	72.63	52.91	229.62	214.49	252.66	194.05	192.44	76.20	125.03	56.18	150.99
		47.81	170.64	169.62	195.10	139.79	153.74	35.98	71.63	48.16	109.56
			216.44	214.71	241.48	185.81	196.71	67.00	116.91	69.98	152.78
				62.25	32.25	43.11	79.75	161.06	106.07	175.49	90.57
					63.30	43.03	31.88	152.11	98.76	159.79	65.17
						61.86	88.32	184.53	128.38	198.55	109.03
							48.88	125.99	70.94	139.31	51.93
								132.66	84.95	137.86	45.39
									58.85	37.54	90.72
										72.47	42.40
											95.76

Figure 2.38: The confusion matrix of Δ across different bottles based on the haptic signals. Higher Δ values indicate lower similarity.

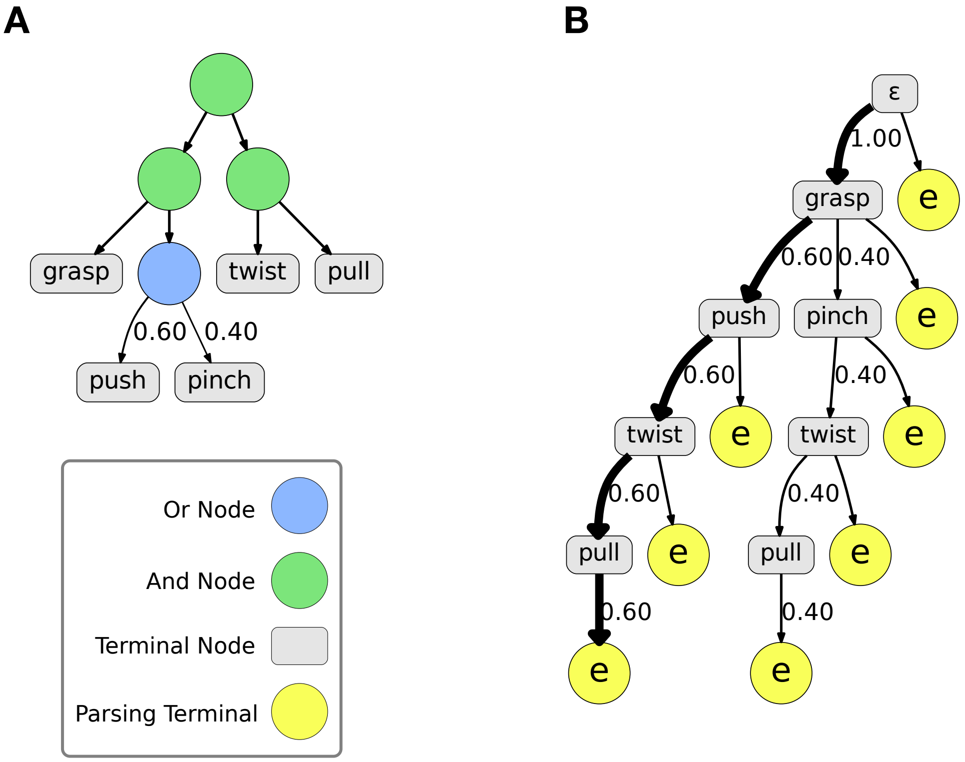


Figure 2.39: **Action grammars and grammar prefix trees used for parsing.** (**A**) An example action grammar. (**B**) A grammar prefix tree with grammar priors. The numbers along edges are the prefix or parsing probabilities of the action sequence represented by the path from the root node to the node pointed by the edge. When the corresponding child node of an edge is an action terminal, the number along the edge represents a prefix probability; when the corresponding child is a parsing terminal e, the number represents the parsing probability of the entire sentence. In this example, the action sequence "grasp, push, twist, pull" has the highest probability of 0.6. The root ϵ represents the empty symbol where no terminals were parsed.

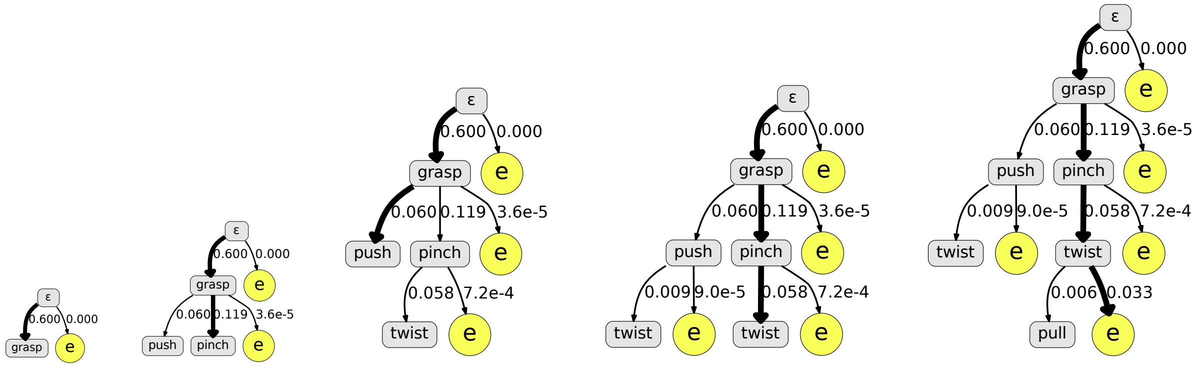
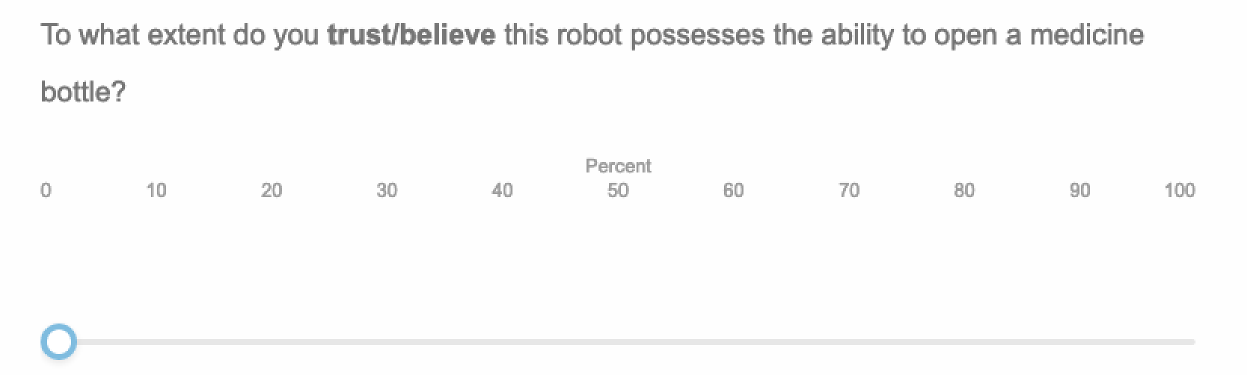


Figure 2.40: An illustration of the parsing process of the Generalized Earley Parser (GEP). It performs a heuristic search in the prefix tree according to the prefix/parsing probability. It iteratively expands the tree and computes the probabilities as it expands the tree. The search ends when it hits a parsing terminal e. The paths in bold indicate the best candidates at each search step.

Algorithm 1: Algorithm of the improved Generalized Earley Parser (GEP) for robot planning.

```
Input          : Grammar $G$, Haptic Model $H$, Maximum Step $T$
InputStream    : Haptic Signal $f_t$
OutputStream: Robot Executable Action $a_{t+1}$
$t = 0$
Initialize empty matrix $y_0$
while $t <= T$ do
    if $t == 0$ then
        $p(a_{t+1})$ = uniformVector()
    else
        $f_t$ = getHapticSignal()
        $p(a_{t+1})$ = hapticPlanner($f_t, a_t; H$)  // Equation 2.22
    end
    $y'_{t+1} = [y_t; p(a_{t+1})]$  // Append probability vector to one-hot matrix $y_t$
    $a_{t+1}$ = prefixSearch($G, y'_{t+1}$)  // Equation 2.28
    $y_{t+1} = [y_t$; oneHot($a_{t+1}$)]  // Extend one-hot matrix from $y_t$ to $y_{t+1}$
    executeRobotAction($a_{t+1}$)
    if goalAchieved() then break
    $t = t + 1$
end
```



To what extent do you **trust/believe** this robot possesses the ability to open a medicine bottle?

Figure 2.41: Qualitative trust question asked to human subjects after observing two demonstrations of robot execution. This question was immediately asked after the familiarization phase of the experiment; in other words, we asked this question immediately after the subjects had observed robot executions *with* access to the explanation panel (if the subject's group had access to an explanation panel; *i.e.* all groups except baseline).

Table 2.10: Network architecture and parameters of the autoencoder. Network architecture is defined from the top of the table to the bottom, with the first and last layer being input and output, respectively.

Operator	**Params**
Linear	80
ReLU	
Linear	64
ReLU	
Linear	16
ReLU	
Linear	8
ReLU	
Linear	16
ReLU	
Linear	64
ReLU	
Linear	80

Table 2.11: Network architecture and parameters for robot to human embedding. Network architecture is defined from the top of the table to the bottom, with the first and last layer being input and output, respectively.

Operator	**Params**
Linear, Linear	3, 1
ReLU, ReLU	
Linear, Linear	128, 128
ReLU	
Linear	8

Table 2.12: Network architecture and parameters for action prediction. Network architecture is defined from the top of the table to the bottom, with the first and last layer being input and output, respectively.

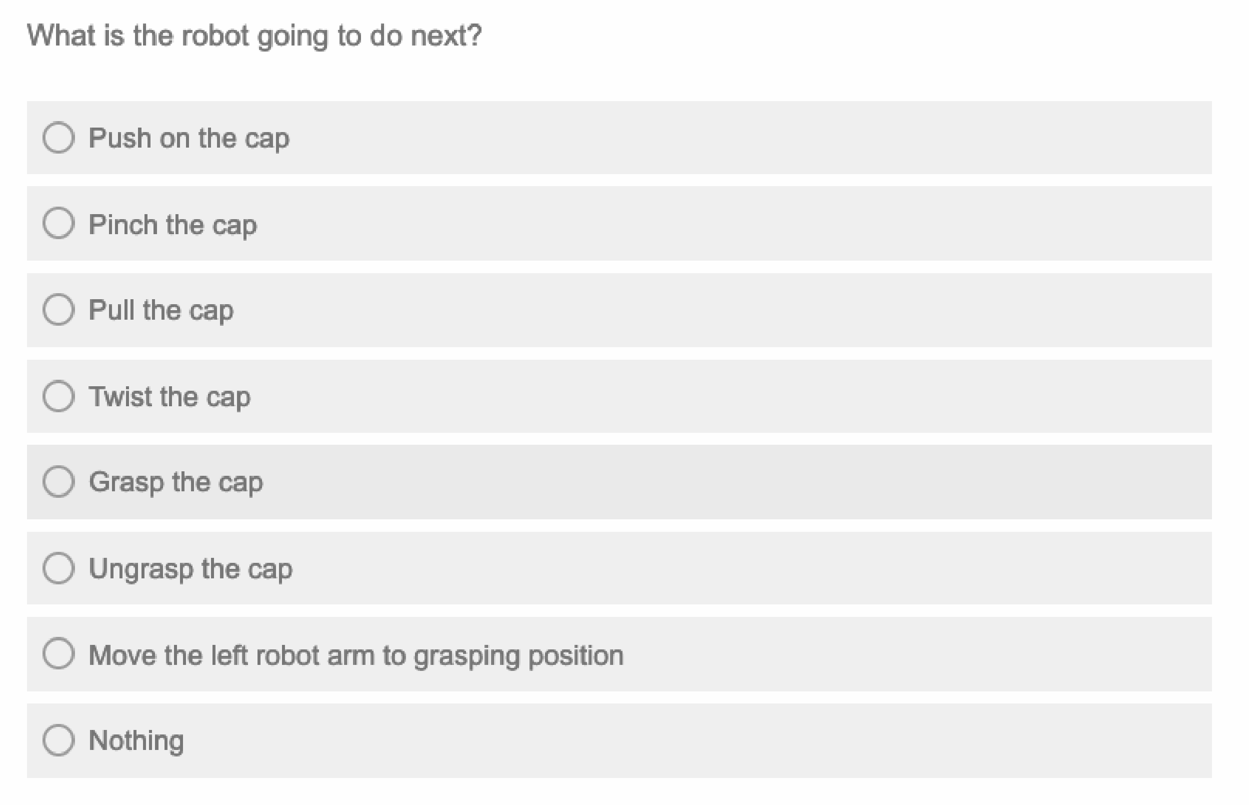
Operator	**Params**
Linear, Linear	8, 13
ReLU, ReLU	
Linear, Linear	64, 64
ReLU	
Linear	10

Table 2.13: Hyper-parameters used during training.

Parameter	**Value**
Autoencoder learning rate	5e-5
Action predictor learning rate	5e-5
Balance param. (β)	1
Batch size	16
No. of epochs	150

Table 2.14: Specifications of the computing platform used in the experiments.

Parts	**Description**
Robot	Baxter
Manipulator	Right: ReFlex TackkTile gripper. Left: Robotiq S85 parallel gripper
Computer	ZOTAC ZBOX-EN1070K: i5-7500T with GTX 1070
Vision sensor	Kinect v2



What is the robot going to do next?

○ Push on the cap

○ Pinch the cap

○ Pull the cap

○ Twist the cap

○ Grasp the cap

○ Ungrasp the cap

○ Move the left robot arm to grasping position

○ Nothing

Figure 2.42: Prediction accuracy question asked to human subjects after each segment of the robot's action sequence during the prediction phase of the experiment. No group had access to explanation panels during the prediction phase; subjects had to predict the action while only observing RBG videos of each action segment.

CHAPTER 3

Reasoning: Multi-Modal Visual Reasoning

3.1 Building A Dataset for Visual Analogy Reasoning

Dramatic progress has been witnessed in basic vision tasks involving low-level perception, such as object recognition, detection, and tracking. Unfortunately, there is still an enormous performance gap between artificial vision systems and human intelligence in terms of higher-level vision problems, especially ones involving reasoning. Earlier attempts in equipping machines with high-level reasoning have hovered around Visual Question Answering (VQA), one typical task associating vision and language understanding. In this work, we propose a new dataset, built in the context of Raven's Progressive Matrices (RPM) and aimed at lifting machine intelligence by associating vision with structural, relational, and analogical reasoning in a hierarchical representation. Unlike previous works in measuring abstract reasoning using RPM, we establish a semantic link between vision and reasoning by providing structure representation. This addition enables a new type of abstract reasoning by jointly operating on the structure representation. Machine reasoning ability using modern computer vision is evaluated in this newly proposed dataset. Additionally, we also provide human performance as a reference. Finally, we show consistent improvement across all models by incorporating a simple neural module that combines visual understanding and structure reasoning.

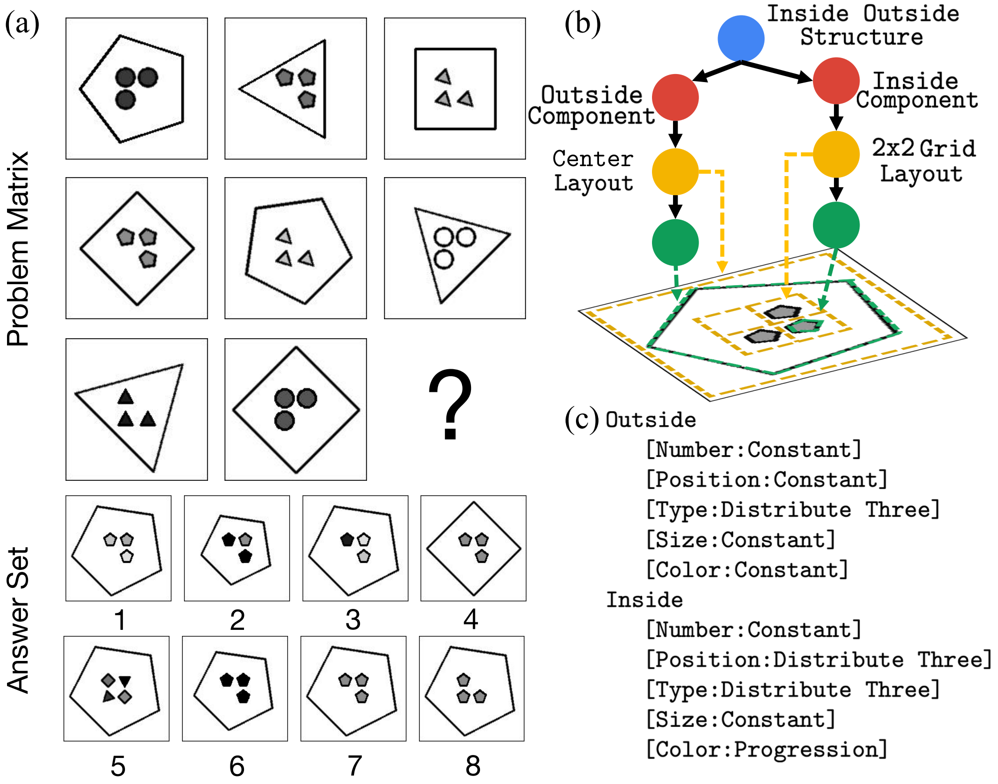
3.1.1 Introduction

> The study of vision must therefore include not only the study of how to extract from images . . . , but also an inquiry into the nature of the *internal representations* by which we *capture* this information and thus make it available as a **basis** for *decisions about our thoughts and actions.*
>
> — David Marr, 1982 [Mar82]

Computer vision has a wide spectrum of tasks. Some computer vision problems are clearly purely visual, "capturing" the visual information process; for instance, filters in early vision [CR68], primal sketch [GZW07] as the intermediate representation, and Gestalt laws [KK79] as the perceptual organization. In contrast, some other vision problems have trivialized requirements for perceiving the image, but engage more generalized problem-solving in terms of relational and/or analogical visual reasoning [HHT96]. In such cases, the vision component becomes the "basis for decisions about our thoughts and actions".

Currently, the majority of the computer vision tasks focus on "capturing" the visual information process; few lines of work focus on the later part—the relational and/or analogical visual reasoning. One existing line of work in equipping artificial systems with reasoning ability hovers around Visual Question Answering (VQA) [AAL15, JHM17a, RKZ15, YWG18, ZGB16]. However, the reasoning skills required in VQA lie only at the periphery of the cognitive ability test circle [CJS90]. To push the limit of computer vision or more broadly speaking, Artificial Intelligence (AI), towards the center of cognitive ability test circle, we need a test originally designed for measuring human's intelligence to challenge, debug, and improve the current artificial systems.

A surprisingly effective ability test of human visual reasoning has been developed and identified as the Raven's Progressive Matrices (RPM) [KMG13, Rav38, SCS13], which is widely accepted and believed to be highly correlated with real intelligence [CJS90]. Unlike VQA, RPM lies directly at the center of human intelligence [CJS90], is diagnostic of abstract



```
Outside
    [Number:Constant]
    [Position:Constant]
    [Type:Distribute Three]
    [Size:Constant]
    [Color:Constant]
Inside
    [Number:Constant]
    [Position:Distribute Three]
    [Type:Distribute Three]
    [Size:Constant]
    [Color:Progression]
```

Figure 3.1: (a) An example RPM. One is asked to select an image that best completes the problem matrix, following the structural and analogical relations. Each image has an underlying structure. (b) Specifically in this problem, it is an inside-outside **structure** in which the outside **component** is a **layout** with a single centered object and the inside **component** is a 2 × 2 grid **layout**. Details in Figure 3.2. (c) lists the rules for (a). The compositional nature of the rules makes this problem a difficult one. The correct answer is 7.

and structural reasoning ability [EKM84], and characterizes the defining feature of high-level intelligence, *i.e.*, *fluid intelligence* [JBJ08].

Figure 3.1 shows an example of RPM problem together with its structure representation. Provided two rows of figures consisting of visually simple elements, one must efficiently derive the correct image structure (Figure 3.1(b)) and the underlying rules (Figure 3.1(c)) to jointly reason about a candidate image that best completes the problem matrix. In terms of levels of reasoning required, RPM is arguably harder compared to VQA:

- Unlike VQA where natural language questions usually imply what to pay attention to in the image, RPM relies merely on visual clues provided in the matrix and the *correspondence problem* itself, *i.e.*, finding the correct level of attributes to encode, is already a major factor distinguishing populations of different intelligence [CJS90].
- While VQA only requires spatial and semantic understanding, RPM needs joint spatial-temporal reasoning in the problem matrix and the answer set. The limit of *short-term memory*, the ability of *analogy*, and the discovery of the *structure* have to be taken into consideration.
- Structures in RPM make the compositions of rules much more complicated. Unlike VQA whose questions only encode relatively simple first-order reasoning, RPM usually includes more sophisticated logic, even with recursions. By composing different rules at various levels, the reasoning progress can be extremely difficult.

To push the limit of current vision systems' reasoning ability, we generate a new dataset to promote further research in this area. We refer to this dataset as the Relational and Analogical Visual rEasoNing dataset (RAVEN) in homage to John Raven for the pioneering work in the creation of the original RPM [Rav38]. In summary:

- RAVEN consists of $1,120,000$ images and $70,000$ RPM problems, equally distributed in 7 distinct figure configurations.
- Each problem has 16 tree-structure annotations, totaling up to $1,120,000$ structural labels in the entire dataset.
- We design 5 rule-governing attributes and 2 noise attributes. Each rule-governing attribute goes over one of 4 rules, and objects in the same component share the same set of rules, making in total $440,000$ rule annotations and an average of 6.29 rules per problem.

The RAVEN dataset is designed inherently to be light in visual recognition and heavy in reasoning. Each image only contains a limited set of simple gray-scale objects with clear-cut boundaries and no occlusion. In the meantime, rules are applied row-wise, and there could be one rule for each attribute, attacking visual systems' major weaknesses in *short-term* *memory* and *compositional reasoning* [JHM17a].

An obvious paradox is: in this innately compositional and structured RPM problem, no annotations of structures are available in previous works (*e.g.*, [BHS18, WS15]). Hence, we set out to establish a semantic link between visual reasoning and structure reasoning in RPM. We ground each problem instance to a sentence derived from an Attributed Stochastic Image Grammar (A-SIG) [Fu74, LWP09, PZ15, WXZ07, ZWZ16, ZM07] and decompose the data generation process into two stages: the first stage samples a sentence from a pre-defined A-SIG and the second stage renders an image based on the sentence. This structured design makes the dataset very diverse and easily extendable, enabling generalization tests in different figure configurations. More importantly, the data generation pipeline naturally provides us with abundant dense annotations, especially the structure in the image space. This semantic link between vision and structure representation opens new possibilities by breaking down the problem into image understanding and tree- or graph-level reasoning [KW16, TSM15]. As shown in Section 4.2.5, we empirically demonstrate that models with a simple structure reasoning module to incorporate both vision-level understanding and structure-level reasoning would notably improve their performance in RPM.

The organization of this section is as follows. In Section 3.1.2, we discuss related work in visual reasoning and computational efforts in RPM. Section 3.1.3 is devoted to a detailed description of the RAVEN dataset generation process, with Section 3.1.4 benchmarking human performance and comparing RAVEN with a previous RPM dataset. In Section 3.1.5, we propose a simple extension to existing models that incorporates vision understanding and structure reasoning. All baseline models and the proposed extensions are evaluated in Section 3.1.6. The notable gap between human subjects (84%) and vision systems (59%) calls for further research into this problem. We hope RAVEN could contribute to the long-standing effort in human-level reasoning AI.

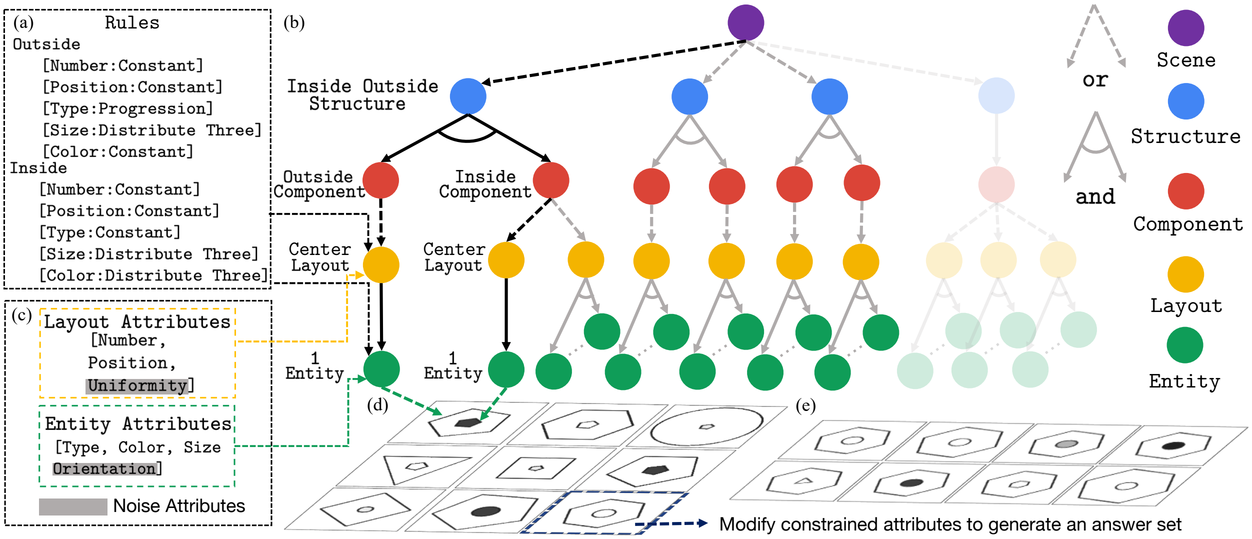


Figure 3.2: RAVEN creation process. A graphical illustration of the grammar production rules used in A-SIG is shown in (b). Note that `Layout` and `Entity` have associated attributes (c). Given a randomly sampled rule combination (a), we first prune the grammar tree (the transparent branch is pruned). We then sample an image structure together with the values of the attributes from (b), denoted by black, and apply the rule set (a) to generate a single row. Repeating the process three times yields the entire problem matrix in (d). (e) Finally, we sample constrained attributes and vary them in the correct answer to break the rules and obtain the candidate answer set.

3.1.2 Related Work

Visual Reasoning Early attempts were made in 1940s-1970s in the field of logic-based AI. Newell argued that one of the potential solutions to AI was "to construct a single program that would take a standard intelligence test" [New73]. There are two important trials: (i) Evans presented an AI algorithm that solved a type of geometric analogy tasks in the Wechsler Adult Intelligence Scale (WAIS) test [Eva62, Eva64], and (ii) Simon and Kotovsky devised a program that solved Thurstone letter series completion problems [TT41]. However, these early attempts were heuristic-based with hand-crafted rules, making it difficult to apply to other problems.

The reasoning ability of modern vision systems was first systematically analyzed in the CLEVR dataset [JHM17a]. By carefully controlling inductive bias and slicing the vision systems' reasoning ability into several axes, Johnson *et al.* successfully identified major drawbacks of existing models. A subsequent work [JHM17b] on this dataset achieved good performance by introducing a program generator in a structured space and combining it with a program execution engine. A similar work that also leveraged language-guided structured reasoning was proposed in [HAR17]. Modules with special attention mechanism were latter proposed in an end-to-end manner to solve this visual reasoning task [HM18, SRB17, ZZH17]. However, superior performance gain was observed in very recent works [CLL18, MTS18, YWG18] that fell back to structured representations by using primitives, dependency trees, or logic. These works also inspire us to incorporate structure information into solving the RPM problem.

More generally, Bisk *et al.* [BSC18] studied visual reasoning in a 3D block world. Perez *et al.* [PSD18] introduced a conditional layer for visual reasoning. Aditya *et al.* [AYB18] proposed a probabilistic soft logic in an attention module to increase model interpretability. And Barrett *et al.* [BHS18] measured abstract reasoning in neural networks.

Computational Efforts in RPM The research community of cognitive science has tried to attack the problem of RPM with computational models earlier than the computer science community. However, an oversimplified assumption was usually made in the experiments that the computer programs had access to a symbolic representation of the image and the operations of rules [CJS90, LF17, LFU10, LTF09]. As reported in Section 3.1.4.4, we show that giving this critical information essentially turns it into a searching problem. Combining it with a simple heuristics provides us an optimal solver, easily surpassing human performance. Another stream of AI research [LLG12, MG14, MKG14b, MSD18, SG18b] tries to solve RPM by various measurements of image similarity. To promote fair comparison between computer programs and human subjects in a data-driven manner, Wang and Su [WS15] first proposed a systematic way of automatically generating RPM using first-order logic. Barrett *et al.* [BHS18] extended their work and introduced the PGM dataset by instantiating each rule with a relation-object-attribute tuple. Hoshen and Werman [HW17] first trained a CNN to complete the rows in a simplistic evaluation environment, while Barrett *et al*. [BHS18] used an advanced Wild Relational Network (WReN) and studied its generalization.

3.1.3 Creating RAVEN

Our work is built on prior work aforementioned. We implement all relations in Advanced Raven's Progressive Matrices identified by Carpenter *et al*. [CJS90] and generate the answer set following *the monotonicity of RPM's constraints* proposed by Wang and Su [WS15].

Figure 3.2 shows the major components of the generation process. Specifically, we use the A-SIG as the representation of RPM; each RPM is a parse tree that instantiates from the A-SIG. After rules are sampled, we prune the grammar to make sure the relations could be applied on any sentence sampled from it. We then sample a sentence from the pruned grammar, where rules are applied to produce a valid row. Repeating such a process three times yields a problem matrix. To generate the answer set, we modify attributes on the correct answer such that the relationships are broken. Finally, the structured presentation is fed into a rendering engine to generate images. We elaborate the details below[1].

3.1.3.1 Defining the Attributed Grammar

We adopt an A-SIG as the hierarchical and structured image grammar to represent the RPM problem. Such representation is advanced compared with prior work (*e.g.*, [BHS18, WS15]) which, at best, only maintains a flat representation of rules.

See Figure 3.2 for a graphical illustration of the grammar production rules. Specifically, the A-SIG for RPM has 5 levels—`Scene`, `Structure`, `Component`, `Layout`, and `Entity`. Note that each grammar level could have multiple instantiations, *i.e.*, different categories or types.

[1] See the supplementary material for productionexamples.
rules, semantic meanings of rules and nodes, and more

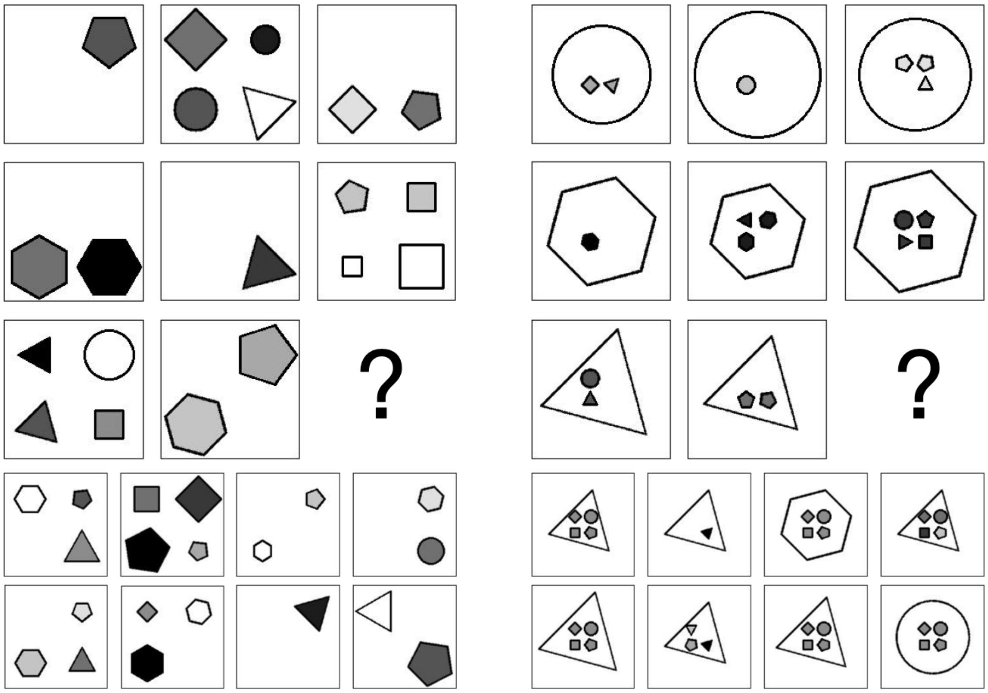


Figure 3.3: Examples of RPM that show the effects of adding *noise* attributes. (Left) `Position`, `Type`, `Size`, and `Color` could vary freely as long as `Number` follows the rule. (Right) `Position` and `Type` in the inside group could vary freely.

The `Scene` level could choose any available `Structure`, which consists of possibly multiple `Components`. Each `Component` branches into `Layouts` that links `Entities`. Attributes are appended to certain levels; for instance, (i) `Number` and `Position` are associated with `Layout`, and (ii) `Type`, `Size`, and `Color` are associated with `Entity`. Each attribute could take a value from a finite set. During sampling, both image structure and attribute values are sampled.

To increase the challenges and difficulties in the RAVEN dataset, we further append 2 types of *noise* attributes—`Uniformity` and `Orientation`—to `Layout` and `Entity`, respectively. `Uniformity`, set false, will not constrain `Entities` in a `Layout` to look the same, while `Orientation` allows an `Entity` to self-rotate. See Figure 3.3 for the effects of the noise attributes.

This grammatical design of the image space allows the dataset to be very diverse and easily extendable. In this dataset, we manage to derive 7 configurations by combining

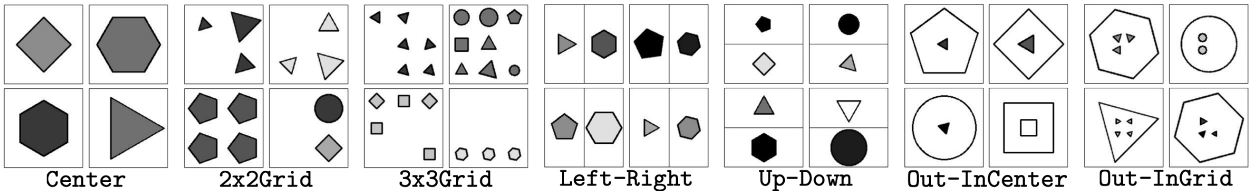


Figure 3.4: Examples of 7 different figure configurations in the proposed RAVEN dataset.

different `Structures`, `Components`, and `Layouts`. Figure 3.4 shows examples in each figure configuration.

3.1.3.2 Applying Rules

Carpenter *et al.* [CJS90] summarized that in the advanced RPM, rules were applied row-wise and could be grouped into 5 types. Unlike Berrett *et al.* [BHS18], we strictly follow Carpenter *et al.*'s description of RPM and implement all the rules, except that we merge `Distribute Two` into `Distribute Three`, as the former is essentially the latter with a null value in one of the attributes.

Specifically, we implement 4 types of rules in RAVEN: `Constant`, `Progression`, `Arithmetic`, and `Distribute Three`. Different from [BHS18], we add internal parameters to certain rules (*e.g.*, `Progression` could have increments or decrements of 1 or 2), resulting in a total of 8 distinct rule instantiations. Rules do not operate on the 2 noise attributes. As shown in Figure 3.1 and Figure 3.2, they are denoted as `[attribute:rule]` pairs.

To make the image space even more structured, we require each attribute to go over one rule and all `Entities` in the same `Component` to share the same set of rules, while different `Components` could vary.

Given the tree representation and the rules, we first prune the grammar tree such that all sub-trees satisfy the constraints imposed by the relations. We then sample from the tree and apply the rules to compose a row. Iterating the process three times yields a problem matrix.

3.1.3.3 Generating the Answer Set

To generate the answer set, we first derive the correct representation of the solution and then leverage the monotonicity of RPM constraints proposed by Wang and Su [WS15]. To break the correct relationships, we find an attribute that is constrained by a rule as described in Section 3.1.3.2 and vary it. By modifying only one attribute, we could greatly reduce the computation. Such modification also increases the difficulty of the problem, as it requires attention to subtle difference to tell an incorrect candidate from the correct one.

3.1.4 Comparison and Analysis

In this section, we compare RAVEN with the existing PGM, presenting its key features and some statistics in Section 3.1.4.1. In addition, we fill in two missing pieces in a desirable RPM dataset, *i.e.*, structure and hierarchy (Section 3.1.4.2), as well as the human performance (Section 3.1.4.3). We also show that RPM becomes trivial and could be solved instantly using a heuristics-based searching method (Section 3.1.4.4), given a symbolic representation of images and operations of rules.

3.1.4.1 Comparison with PGM

Table 3.1 summarizes several essential metrics of RAVEN and PGM. Although PGM is larger than RAVEN in terms of size, it is very limited in the average number of rules (**AvgRule**), rule instantiations (**RuleIns**), number of structures (**Struct**), and figure configurations (**FigConfig**). This contrast in PGM's gigantic size and limited diversity might disguise model fitting as a misleading reasoning ability, which is unlikely to generalize to other scenarios.

To avoid such an undesirable effect, we refrain from generating a dataset too large, even though our structured representation allows generation of a combinatorial number of problems. Rather, we set out to incorporate more rule instantiations (8), structures (4), and figure configurations (7) to make the dataset diverse (see Figure 3.4 for examples). Note that an equal number of images for each figure configuration is generated in the RAVEN dataset.

3.1.4.2 Introduction of Structure

A distinctive feature of RAVEN is the introduction of the structural representation of the image space. Wang and Su [WS15] and Barrett *et al.* [BHS18] used plain logic and flat rule representations, respectively, resulting in no base of the structure to perform reasoning on. In contrast, we have in total $1,120,000$ structure annotations (**StructAnno**) in the form of parsed sentences in the dataset, pairing each problem instance with 16 sentences for both the matrix and the answer set. These representations derived from the A-SIG allow a new form of reasoning, *i.e.*, one that combines visual understanding and structure reasoning. As shown in [LF17, LFU10, LTF09] and our experiments in Section 4.2.5, incorporating structure into RPM problem solving could result in further performance improvement across different models.

3.1.4.3 Human Performance Analysis

Another missing point in the previous work [BHS18] is the evaluation of human performance. To fill in the missing piece, we recruit human subjects consisting of college students from a subject pool maintained by the Department of Psychology to test their performance on a

	PGM [BHS18]	**RAVEN (Ours)**
AvgRule	1.37	6.29
RuleIns	5	8
Struct	1	4
FigConfig	3	7
StructAnno	0	1,120,000
HumanPerf		✓

Table 3.1: Comparison with the PGM dataset.

subset of representative samples in the dataset. In the experiments, human subjects were familiarized by solving problems with only one non-`Constant` rule in a fixed configuration. After the familiarization, subjects were asked to answer RPM problems with complex rule combinations, and their answers were recorded. Note that we deliberately included all figure configurations to measure generalization in the human performance and only "easily perceptible" examples were used in case certain subjects might have impaired perception. The results are reported in Table 3.2. The notable performance gap calls for further research into this problem. See Section 4.2.5 for detailed analysis and comparisons with vision models.

3.1.4.4 Heuristics-based Solver using Searching

We also find that the RPM could be essentially turned into a searching problem, given the symbolic representation of images and the access to rule operations as in [LF17, LFU10, LTF09]. Under such a setting, we could treat this problem as constraint satisfaction and develop a heuristics-based solver. The solver checks the number of satisfied constraints in each candidate answer and selects one with the highest score, resulting in perfect performance. Results are reported in Table 3.2. The optimality of the heuristic-based solver also verifies the well-formedness of RAVEN in the sense that there exists only one candidate that satisfies all constraints.

3.1.5 Dynamic Residual Tree for RPM

The image space of RPM is inherently structured and could be described using a symbolic language, as shown in [CJS90, LF17, LFU10, LTF09, Rav38]. To capture this characteristic and further improve the model performance on RPM, we propose a simple tree-structure neural module called Dynamic Residual Tree (DRT) that operates on the joint space of image understanding and structure reasoning. An example of DRT is shown in Figure 3.5.

In the DRT, given a sentence S sampled from the A-SIG, usually represented as a serialized n-ary tree, we could first recover the tree structure. Note that the tree is **dynamically** generated following the sentence S, and each node in the tree comes with a label. With a structured tree representation ready, we could now consider assigning a neural computation operator to each tree node, similar to Tree-LSTM [TSM15]. To further simplify computation, we replace the LSTM cell [HS97] with a ReLU-activated [NH10] fully-connected layer f. In this way, nodes with a single child (leaf nodes or OR-production nodes) update the input features by

$$I = \mathrm{ReLU}(f([I, w_n])), \tag{3.1}$$

where $[\cdot, \cdot]$ is the concatenation operation, I denotes the input features, and w_n the distributed representations of the node's label [MSC13, PSM14]. Nodes with multiple children (AND-production nodes) update input features by

$$I = \mathrm{ReLU}\left(f\left(\left[\sum_c I_c, w_n\right]\right)\right), \tag{3.2}$$

where I_c denotes the features from its child c.

In summary, features from the lower layers are fed into the leaf nodes of DRT, gradually updated by Equation 3.1 and Equation 3.2 from bottom-up following the tree structure, and output to higher-level layers.

Inspired by [HZR16], we make DRT a **residual** module by adding the input and output of DRT together, hence the name Dynamic Residual Tree (DRT)

$$I = \mathrm{DRT}(I, S) + I. \tag{3.3}$$

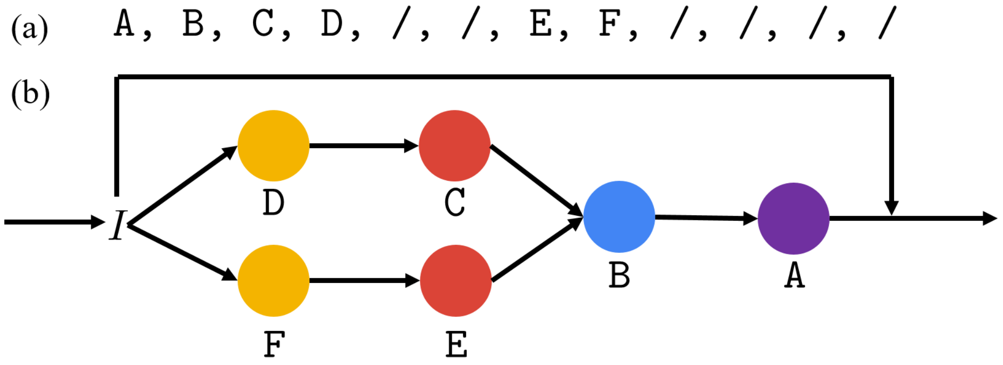


Figure 3.5: An example computation graph of DRT. (a) Given the serialized n-ary tree representation (pre-order traversal with `/` denoting end-of-branch), (b) a tree-structured computation graph is dynamically built. The input features are wired from bottom-up following the tree structure. The final output is the sum with the input, forming a residual module.

3.1.6 Experiments

3.1.6.1 Computer Vision Models

We adopt several representative models suitable for RPM and test their performances on RAVEN [BHS18, HZR16, KSH12, XCW15]. In summary, we test a simple sequential learning model (LSTM), a CNN backbone with an MLP head (CNN), a ResNet-based [HZR16] image classifier (ResNet), the recent relational WReN [BHS18], and all these models augmented with the proposed DRT.

LSTM The partially sequential nature of the RPM problem inspires us to borrow the power of sequential learning. Similar to ConvLSTM [XCW15], we feed each image feature extracted by a CNN into an LSTM network sequentially and pass the last hidden feature into a two-layer MLP to predict the final answer. In the DRT-augmented LSTM, *i.e.*, LSTM-DRT, we feed features of each image to a shared DRT before the final LSTM.

CNN We test a neural network model used in Hoshen and Werman [HW17]. In this model, a four-layer CNN for image feature extraction is connected to a two-layer MLP with a softmax layer to classify the answer. The CNN is interleaved with batch normalization [IS15] and ReLU non-linearity [NH10]. Random dropout [SHK14] is applied at the penultimate layer of MLP. In CNN-DRT, image features are passed to DRT before MLP.

ResNet Due to its surprising effectiveness in image feature extraction, we replace the feature extraction backbone in CNN with a ResNet [HZR16] in this model. We use a publicly available ResNet implementation, and the model is randomly initialized without pre-training. After testing several ResNet variants, we choose ResNet-18 for its good performance. The DRT extension and the training strategy are similar to those used in the CNN model.

WReN We follow the original paper [BHS18] in implementing the WReN. In this model, we first extract image features by a CNN. Each answer feature is then composed with each context image feature to form a set of ordered pairs. The order pairs are further fed to an MLP and summed. Finally, a softmax layer takes features from each candidate answer and makes a prediction. In WReN-DRT, we apply DRT on the extracted image features before the relational module.

For all DRT extensions, nodes in the same level share parameters and the representations for nodes' labels are fixed after initialization from corresponding 300-dimension GloVe vectors [PSM14]. Sentences used for assembling DRT could be either retrieved or learned by an encoder-decoder. Here we report results using retrieval.

3.1.6.2 Experimental Setup

We split the RAVEN dataset into three parts, 6 folds for training, 2 folds for validation, and 2 folds for testing. We tune hyper-parameters on the validation set and report the model accuracy on the test set. For loss design, we treat the problem as a classification

Method	Acc	`Center`	`2x2Grid`	`3x3Grid`	`L-R`	`U-D`	`O-IC`	`O-IG`
LSTM	13.07%	13.19%	14.13%	13.69%	12.84%	12.35%	12.15%	12.99%
WReN	14.69%	13.09%	28.62%	28.27%	7.49%	6.34%	8.38%	10.56%
CNN	36.97%	33.58%	30.30%	33.53%	39.43%	41.26%	43.20%	37.54%
ResNet	53.43%	52.82%	41.86%	44.29%	58.77%	60.16%	63.19%	53.12%
LSTM+DRT	13.96%	14.29%	15.08%	14.09%	13.79%	13.24%	13.99%	13.29%
WReN+DRT	15.02%	15.38%	23.26%	29.51%	6.99%	8.43%	8.93%	12.35%
CNN+DRT	39.42%	37.30%	30.06%	34.57%	45.49%	45.54%	45.93%	37.54%
ResNet+DRT	**59.56%**	**58.08%**	**46.53%**	**50.40%**	**65.82%**	**67.11%**	**69.09%**	**60.11%**
Human	84.41%	95.45%	81.82%	79.55%	86.36%	81.81%	86.36%	81.81%
Solver*	100%	100%	100%	100%	100%	100%	100%	100%

Table 3.2: Testing accuracy of each model against human subjects and the solver. Acc denotes the mean accuracy of each model, while other columns show model accuracy on different figure configurations. `L-R` denotes `Left-Right`, `U-D` denotes `Up-Down`, `O-IC` denotes `Out-InCenter`, and `O-IG` denotes `Out-InGrid`. *Note that the perfect solver has access to rule operations and searches on the symbolic problem representation.

task and train all models with the cross-entropy loss. All the models are implemented in PyTorch [PGC17] and trained with ADAM [KB14] before early stopping or a maximum number of epochs is reached.

3.1.6.3 Performance Analysis

Table Table 3.2 shows the testing accuracy of each model trained on RAVEN, against the human performance and the heuristics-based solver. Neither human subjects nor the solver experiences an intensive training session, and the solver has access to the rule operations and searches the answer based on a symbolic representation of the problem. In contrast, all the computer vision models go over an extensive training session, but only on the training set.

In general, human subjects produce better testing accuracy on problems with simple figure configurations such as `Center`, while human performance reasonably deteriorates on problem instances with more objects such as `2x2Grid` and `3x3Grid`. Two interesting observations:

1. For figure configurations with multiple components, although each component in `Left-Right`, `Up-Down`, and `Out-InCenter` has only one object, making the reasoning similar to `Center` except that the two components are independent, human subjects become less accurate in selecting the correct answer.
2. Even if `Up-Down` could be regarded as a simple transpose of `Left-Right`, there exists some notable difference. Such effect is also implied by the "inversion effects" in cognition; for instance, inversion disrupts face perception, particularly sensitivity to spatial relations [CM09, LMM01].

In terms of model performance, a counter-intuitive result is: computer vision systems do not achieve the best accuracy across all other configurations in the seemingly easiest figure configuration for human subjects (`Center`). We further realize that the LSTM model and the WReN model perform only slightly better than random guess (12.5%). Such results contradicting to [BHS18] might be attributed to the diverse figure configurations in RAVEN. Unlike LSTM whose accuracy across different configurations is more or less uniform, WReN achieves higher accuracy on configurations consisting of multiple randomly distributed objects (`2x2Grid` and `3x3Grid`), with drastically degrading performance in configurations consisting of independent image components. This suggests WReN is biased to grid-like configurations (majority of PGM) but not others that require compositional reasoning (as in RAVEN). In contrast, a simple CNN model with MLP doubles the performance of WReN on RAVEN, with a tripled performance if the backbone is ResNet-18.

We observe a consistent performance improvement across different models after incorporating DRT, suggesting the effectiveness of the structure information in this visual reasoning problem. While the performance boost is only marginal in LSTM and WReN, we notice a marked accuracy increase in the CNN- and ResNet-based models (6.63% and 16.58% relative increase respectively). However, the performance gap between artificial vision systems and humans are still significant (up to 37% in `2x2Grid`), calling for further research to bridge the gap.

3.1.6.4 Effects of Auxiliary Training

Barrett *et al*. [BHS18] mentioned that training WReN with a fine-tuned auxiliary task could further give the model a 10% performance improvement. We also test the influence of auxiliary training on RAVEN. First, we test the effects of an auxiliary task to classify the rules and attributes on WReN and our best performing model ResNet+DRT. The setting is similar to [BHS18], where we perform an OR operation on a set of multi-hot vectors describing the rules and the attributes they apply to. The model is then tasked to both correctly find the answer and classify the rule set with its governing attributes. The final loss becomes

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \beta \mathcal{L}_{\text{rule}}, \tag{3.4}$$

where $\mathcal{L}_{\text{target}}$ denotes the cross-entropy loss for the answer, $\mathcal{L}_{\text{rule}}$ the multi-label classification loss for the rule set, and β the balancing factor. We observe no performance change on WReN but a serious performance downgrade on ResNet+DRT (from 59.56% to 20.71%).

Since RAVEN comes with structure annotations, we further ask whether adding a structure prediction loss could help the model improve performance. To this end, we cast the experiment in a similar setting where we design a multi-hot vector describing the structure of each problem instance and train the model to minimize

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{target}} + \alpha \mathcal{L}_{\text{struct}}, \tag{3.5}$$

where $\mathcal{L}_{\text{struct}}$ denotes the multi-label classification loss for the problem structure, and α the balancing factor. In this experiment, we observe a slight performance decrease in ResNet+DRT (from 59.56% to 56.86%). A similar effect is noticed on WReN (from 14.69% to 12.58%).

3.1.6.5 Test on Generalization

One interesting question we would like to ask is how a model trained well on one figure configuration performs on another similar figure configuration. This could be a measure of models' generalizability and compositional reasoning ability. Fortunately, RAVEN naturally provides us with a test bed. To do this, we first identify several related configuration regimes:

- Train on `Center` and test on `Left-Right`, `Up-Down`, and `Out-InCenter`. This setting directly challenges the compositional reasoning ability of the model as it requires the model to generalize the rules learned in a single-component configuration to configurations with multiple independent but similar components.
- Train on `Left-Right` and test on `Up-Down`, and vice-versa. Note that for `Left-Right` and `Up-Down`, one could be regarded as a transpose of another. Thus, the test could measure whether the model simply memorizes the pattern in one configuration.
- Train on `2x2Grid` and test on `3x3Grid`, and vice-versa. Both configurations involve multi-object interactions. Therefore the test could measure the generalization when the number of objects changes.

The following results are all reported using the best performing model, *i.e.*, ResNet+DRT.

`Center`	`Left-Right`	`Up-Down`	`Out-InCenter`
51.87%	40.03%	35.46%	38.84%

Table 3.3: Generalization test. The model is trained on `Center` and tested on three other configurations.

	`Left-Right`	`Up-Down`
`Left-Right`	41.07%	38.10%
`Up-Down`	39.48%	43.60%

Table 3.4: Generalization test. The row shows configurations the model is trained on and the column the model is tested on.

Table 3.3, Table 3.4 and Table 3.5 show the result of our model generalization test. We observe:

	2x2Grid	3x3Grid
2x2Grid	40.93%	38.69%
3x3Grid	39.14%	43.72%

Table 3.5: Generalization test. The row shows configurations the model is trained on and the column the model is tested on.

- The model dedicated to a single figure configuration does not achieve better test accuracy than one trained on all configurations together. This effect justifies the importance of the diversity of RAVEN, showing that increasing the number of figure configurations could actually improve the model performance.
- Table 3.3 also implies that a certain level of compositional reasoning, though weak, exists in the model, as the three other configurations could be regarded as a multi-component composition of `Center`.
- In Table 3.4, we observe no major differences in terms of test accuracy. This suggests that the model could successfully transfer the knowledge learned in a scenario to a very similar counterpart, when one configuration is the transpose of another.
- From Table 3.5, we notice that the model trained on `3x3Grid` could generalize to `2x2Grid` with only minor difference from the one dedicated to `2x2Grid`. This could be attributed to the fact that in the `3x3Grid` configuration, there could be instances with object distribution similar to that in `2x2Grid`, but not vice versa.

3.1.7 Conclusion

We present a new dataset for Relational and Analogical Visual Reasoning in the context of Raven's Progressive Matrices (RPM), called RAVEN. Unlike previous work, we apply a systematic and structured tool, *i.e.*, Attributed Stochastic Image Grammar (A-SIG), to generate the dataset, such that every problem instance comes with rich annotations. This tool also makes RAVEN diverse and easily extendable. One distinguishing feature that tells apart RAVEN from other work is the introduction of the structure. We also recruit quality human subjects to benchmark human performance on the RAVEN dataset. These aspects fill two important missing points in previous works.

We further propose a novel neural module called Dynamic Residual Tree (DRT) that leverages the structure annotations for each problem. Extensive experiments show that models augmented with DRT enjoy consistent performance improvement, suggesting the effectiveness of using structure information in solving RPM. However, the difference between machine algorithms and humans clearly manifests itself in the notable performance gap, even in an unfair situation where machines experience an intensive training session while humans do not. We also realize that auxiliary tasks do not help performance on RAVEN. The generalization test shows the importance of diversity of the dataset, and also indicates current computer vision methods do exhibit a certain level of reasoning ability, though weak.

The entire work still leaves us many mysteries. Humans seem to apply a combination of the top-down and bottom-up method in solving RPM. How could we incorporate this into a model? What is the correct way of formulating visual reasoning? Is it model fitting? Is deep learning the ultimate way to visual reasoning? If not, how could we revise the models? If yes, how could we improve the models?

Finally, we hope these unresolved questions would call for attention into this challenging problem.

3.2 Approaching Visual Analogy Problem by Contrasting

"Thinking in pictures," [Gra06] *i.e.*, spatial-temporal reasoning, effortless and instantaneous for humans, is believed to be a significant ability to perform logical induction and a crucial factor in the intellectual history of technology development. Modern AI, fueled by massive datasets, deeper models, and mighty computation, has come to a stage where (super-)human-level performances are observed in certain specific tasks. However, current AI's ability in "thinking in pictures" is still far lacking behind. In this work, we study how to improve machines' reasoning ability on one challenging task of this kind: RPM. Specifically, we borrow the very idea of "contrast effects" from the field of psychology, cognition, and education to design and train a permutation-invariant model. Inspired by cognitive studies, we equip our model with a simple inference module that is jointly trained with the perception backbone. Combining all the elements, we propose the *Contrastive Perceptual Inference* network (CoPINet) and empirically demonstrate that CoPINet sets the new state-of-the-art for permutation-invariant models on two major datasets. We conclude that spatial-temporal reasoning depends on envisaging the possibilities consistent with the relations between objects and can be solved from pixel-level inputs.

3.2.1 Introduction

Among the broad spectrum of computer vision tasks are ones where dramatic progress has been witnessed, especially those involving visual information retrieval [KSH12, HZR16, RDG16, RHG15]. Significant improvement has also manifested itself in tasks associating visual and linguistic understanding [AAL15, JHM17a, JHM17b, HAR17]. However, it was only until recently that the research community started to re-investigate tasks relying heavily on the ability of "thinking in pictures" with modern AI approaches [Gra06, Arn69, Gal83], particularly spatial-temporal inductive reasoning [ZGJ19, HSB19, BHS18]; this line of work primarily focuses on Raven's Progressive Matrices (RPM) [Rav36, RC98]. It is believed that RPM is closely related to real intelligence [CJS90], diagnostic of abstract and structural reasoning ability [EKM84], and characterizes *fluid intelligence* [Spe27, Spe23, Hof95, JBJ08]. In such a test, subjects are provided with two rows of figures following certain *unknown* rules and asked to pick the correct answer from the choices that would best complete the third row with a missing entry; see Figure 3.6(a) for an example. As shown in early works [ZGJ19, BHS18], despite the fact that *visual elements* are relatively straightforward, there is still a notable performance gap between human and machine *visual reasoning* in this challenging task.

One missing ingredient that may result in this performance gap is a proper form of contrasting mechanism. Originated from perceptual learning [GG55, Gib14], it is well established in the field of psychology and education [CH89, GG01, HDW09, GP92, HIO11] that teaching new concepts by comparing with noisy examples is quite effective. [SG14] summarize that comparing cases facilitates transfer learning and problem-solving, as well as the ability to learn relational categories. [Gen83] in his structure-mapping theory points out that learners generate a structure alignment between two representation when they compare two cases. A more recent study from [SCO11] also shows that contrasting cases help foster an appreciation of a deep understanding of concepts.

We argue that such a *contrast effect* [Bow61], found in both humans and animals [Mey51, SH56, SV78, Law57, Ams62], is essential to machines' reasoning ability as well. With access to how the data is generated, a recent attempt [HSB19] finds that models demonstrate better generalizability if the choice of data and the manner in which it is presented to the model are made "contrastive." In this work, we try to address a more direct and challenging question, *independent* of how the data is generated: how to incorporate an explicit contrasting mechanism during model *training* in order to improve machines' reasoning ability? Specifically, we come up with two levels of contrast in our model: a novel contrast module and a new contrast loss. At the model level, we design a permutation-invariant contrast module that summarizes the common features and distinguishes each candidate by projecting it onto its residual on the common feature space. At the objective level, we leverage ideas in contrastive estimation [GH10, SE05, DL17] and propose a variant of Noise-Contrastive Estimation (NCE) loss.

Another reason why RPM is challenging for existing machine reasoning systems could be attributed to the demanding nature of the *interplay* between perception and inference. [CJS90] postulate that a proper understanding of one RPM instance requires not only an accurate encoding of individual elements and their visual attributes but also the correct induction of the hidden rules. In other words, to solve RPM, machine reasoning systems are expected to be equipped with *both* perception and inference subsystems; lacking either component would only result in a sub-optimal solution. While existing work primarily focuses on perception, we propose to bridge this gap with a simple inference module *jointly* trained with the perception backbone; specifically, the inference module reasons about which category the current problem instance falls into. Instead of training the inference module to predict the ground-truth category, we borrow the basis learning idea from [WSG10] and jointly learn the inference subsystem with perception. This basis formulation could also be regarded as a hidden variable and trained using a log probability estimate.

Furthermore, we hope to make a critical improvement to the model design such that it is truly *permutation-invariant*. The invariance is mandatory, as an ideal RPM solver should not change the representation simply because the rows or columns of answer candidates are swapped or the order of the choices alters. This characteristic is an essential trait missed by all recent works [ZGJ19, BHS18]. Specifically, [ZGJ19] stack all choices in the channel dimension and feed it into the network in one pass. [BHS18] add additional positional tagging to their WReN. Both of them *explicitly* make models permutation-sensitive. We notice in our experiments that removing the positional tagging in WReN decreases the performance by 28%, indicating that the model bypasses the intrinsic complexity of RPM by remembering the positional association. Making the model permutation-invariant also shifts the problem from classification to ranking.

Combining contrasting, perceptual inference, and permutation invariance, we propose the *Contrastive Perceptual Inference* network (CoPINet). To verify its effectiveness, we conduct comprehensive experiments on two major datasets: the RAVEN dataset [ZGJ19] and the PGM dataset [BHS18]. Empirical studies show that our model achieves human-level performance on RAVEN and a new record on PGM, setting new state-of-the-art for permutation-invariant models on the two datasets. Further ablation on RAVEN and PGM reveals how each component contributes to performance improvement. We also investigate how the model performance varies under different sizes of datasets, as a step towards an ideal machine reasoning system capable of low-shot learning.

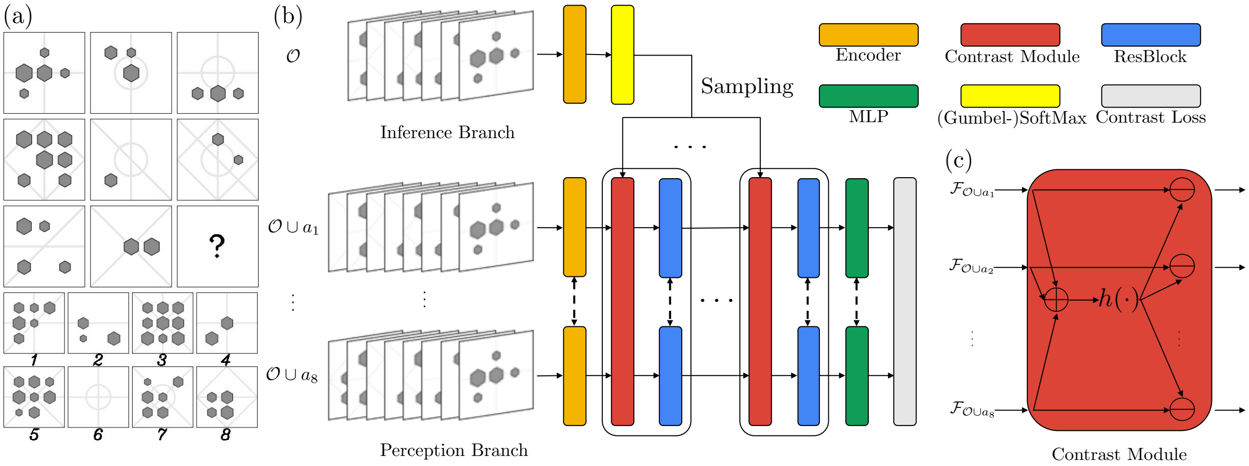


Figure 3.6: (a) An example of RPM. The hidden rule(s) in this problem can be denoted as $\{[\text{OR}, \text{line}, \text{type}]\}$, where an OR operation is applied to the type attribute of all lines, following the notations in [BHS18]. It is further noted that the OR operation is applied row-wise, and there is only one choice that satisfies the row-wise OR constraint. Hence the correct answer should be 5. (b) The proposed CoPINet architecture. Given a RPM problem, the inference branch samples a most likely rule for each attribute based only on the context $\mathcal{O}$ of the problem. Sampled rules are transformed and fed into each contrast module in the perception branch. Note that the combination of the contrast module and the residual block can be repeated. Dashed lines indicate that parameters are shared among the modules. (c) A sketch of the contrast module.

This work makes four major contributions:

- We introduce two levels of contrast to improve machines' reasoning ability in RPM. At the model level, we design a contrast module that aggregates common features and projects each candidate to its residual. At the objective level, we use an NCE loss variant instead of the cross-entropy to encourage contrast effects.
- Inspired by [CJS90], we incorporate an inference module to learn with the perception backbone jointly. Instead of using ground-truth, we regularize it with a fixed number of bases.
- We make our model permutation-invariant in terms of swapped rows or columns and shuffled answer candidates, shifting the previous view of RPM from classification to ranking.
- Combining ideas above, we propose CoPINet that sets new state-of-the-art on two major datasets.

3.2.2 Related Work

Contrastive Learning Teaching concepts by comparing cases, or contrasting, has proven effective in both human learning and machine learning. [Gen83] postulates that human's learning-by-comparison process is a structural mapping and alignment process. A later article [GM94] firmly supports this conjecture and shows finding the individual difference is easier for humans when similar items are compared. Recently, [SG14] conclude that learning by comparing two contrastive cases facilitates the distinction between two complex interrelated relational concepts. Evidence in educational research further strengthens the importance of contrasting—quantitative structure of empirical phenomena is less demanding to learn when contrasting cases are used [SCO11, CSS10, SM04]. All the literature calls for a similar treatment of contrast in machine learning. While techniques from [CHL05, WS09, WG15] are based on triplet loss using max margin to separate positive and negative samples, negative contrastive samples and negative sampling are proposed for language modeling [SE05] and word embedding [MSC13, KZS15], respectively. [GH10] discuss a general learning framework called Noise-Contrastive Estimation (NCE) for estimating parameters by taking noise samples into consideration, which [DL17] follow to learn an effective image captioning model. A recent work [HSB19] leverages contrastive learning in RPM; however, it focuses on data presentation while leaving the question of modeling and learning unanswered.

Computational Models on RPM The cognitive science community is the first to investigate RPM with computational models. Assuming access to a perfect state representation, structure-mapping theory [Gen83] and the high-level perception theory of analogy [CFH92, Mit93] are designed with heuristics to solve the RPM problem at a symbolic level [CJS90, LF17, LFU10, LTF09]. Another stream of research approaches the problem by measuring the image similarity with hand-crafted state representations [LLG12, MG14, MKG14b, MSD18, SG18a]. More recently, end-to-end data-driven methods with raw image input are proposed [ZGJ19, HSB19, BHS18, WS15]. [WS15] introduce an automatic RPM generation method. [BHS18] release the first large-scale RPM dataset and present a relational model [SRB17] designed for it. [SLV18] propose a pretrained β-VAE to improve the generalization performance of models on RPM. [ZGJ19] provide another dataset with structural annotations using stochastic image grammar [ZM07, PZ15, WXZ07]. [HSB19] take a different approach and study how data presentation affects learning.

3.2.3 Learning Perceptual Inference by Contrasting

The task of RPM can be formally defined as: given a list of observed images $\mathcal{O} = \{o_i\}_{i=1}^{8}$, forming a 3 × 3 matrix with a final missing element, a solver aims to find an answer $a_\star$ from an *unordered* set of choices $\mathcal{A} = \{a_i\}_{i=1}^{8}$ to best complete the matrix. Permutation invariance is a unique property for RPM problems: (1) According to [CJS90], the same set of rules is applied either row-wise or column-wise. Therefore, swapping the first two rows or columns should not affect how one solves the problem. (2) In any multi-choice task, changing the order of answer candidates should not affect how one solves the problem either. These properties require us to use a permutation-invariant encoder and reformulate the problem from a typical classification problem into a ranking problem. Formally, in a probabilistic formulation, we seek to find a model such that

$$p(a_\star|\mathcal{O}) \geqslant p(a'|\mathcal{O}), \quad \forall a' \in \mathcal{A}, a' \neq a_\star, \tag{3.6}$$

where the probability is invariant when rows or columns in $\mathcal{O}$ are swapped. This formulation also calls for a model that produces a density estimation for each choice, regardless of its order in $\mathcal{A}$. To that end, we model the probability with a neural network equipped with a permutation-invariant encoder for each observation-candidate pair $f(\mathcal{O} \cup a)$. However, we argue such a purely perceptive system is far from sufficient without contrasting and perceptual inference.

3.2.3.1 Contrasting

To provide the reasoning system with a mechanism of contrasting, we propose to explicitly build two levels of contrast: model-level contrast and objective-level contrast.

Model-level Contrast As the central notion of contrast is comparing cases [SG14, SCO11, CSS10, SM04], we propose an explicit model-level contrasting mechanism in the following form,

$$\text{Contrast}(\mathcal{F}_{\mathcal{O}\cup a}) = \mathcal{F}_{\mathcal{O}\cup a} - h\left(\sum_{a'\in\mathcal{A}} \mathcal{F}_{\mathcal{O}\cup a'}\right), \tag{3.7}$$

where $\mathcal{F}$ denotes features of a specific combination and $h(\cdot)$ summarizes the common features in all candidate answers. In our experiments, $h(\cdot)$ is a composition of BatchNorm [IS15] and Conv.

Intuitively, this explicit contrasting computation enables a reasoning system to tell distinguishing features for each candidate in terms of fitting and following the rules hidden among all panels in the incomplete matrix. The philosophy behind this design is to constrain the functional form of the model to capture both the commonality and the difference in each instance. It is expected that the very inductive bias on comparing similarity and distinctness is baked into the entire reasoning system such that learning in the challenging task becomes easier.

In a generalized setting, each $\mathcal{O} \cup a$ could be abstracted out as an object. Then the design becomes a general contrast module, where each object is distinguished by comparing with the common features extracted from an object set.

We further note that the contrasting computation can be encapsulated into a single neural module and repeated: the addition and transformation are shared and the subtraction is performed on each individual element. See Figure 3.6(c) for a sketch of the contrast module. After such operations, permutation invariance of a model will not be broken.

Objective-level Contrast To further enforce the contrast effects, we propose to use an NCE variant rather than the cross-entropy loss commonly used in previous works [ZGJ19, BHS18]. While there are several ways to model the probability in Equation 3.6, we use a Gibbs distribution in this work:

$$p(a|\mathcal{O}) = \frac{1}{Z} \exp(f(\mathcal{O} \cup a)), \tag{3.8}$$

where Z is the partition function, and our model $f(\cdot)$ corresponds to the negative potential function. Note that such a distribution has been widely adopted in image generation models [ZWM98, WXL18, XLZ16].

In this case, we can take the log of both sides in Equation 3.6 and rearrange terms:

$$\log p(a_\star|\mathcal{O}) - \log p(a'|\mathcal{O}) = f(\mathcal{O} \cup a_\star) - f(\mathcal{O} \cup a') \geqslant 0, \quad \forall a' \in \mathcal{A}, a' \neq a_\star. \tag{3.9}$$

This formulation could potentially lead to a max margin loss. However, we notice in our preliminary experiments that max margin is not sufficient; we realize it is inferior to make the negative potential of the wrong choices only *slightly lower*. Instead, we would like to further push the difference to *infinity*. To do that, we leverage the *sigmoid* function $\sigma(\cdot)$ and train the model, such that:

$$f(\mathcal{O} \cup a_\star) - f(\mathcal{O} \cup a') \to \infty \iff \sigma(f(\mathcal{O} \cup a_\star) - f(\mathcal{O} \cup a')) \to 1, \forall a' \in \mathcal{A}, a' \neq a_\star. \tag{3.10}$$

However, we notice that the relative difference of negative potential is still problematic. We hypothesize this deficiency is due to the lack of a baseline—without such a regularization, the negative potential of wrong choices could still be very high, resulting in difficulties in learning the negative potential of the correct answer. To this end, we modify Equation 3.10

into its sufficient conditions:

$$f(\mathcal{O} \cup a_\star) - b(\mathcal{O} \cup a_\star) \to \infty \iff \sigma(f(\mathcal{O} \cup a_\star) - b(\mathcal{O} \cup a_\star)) \to 1 \tag{3.11}$$

$$f(\mathcal{O} \cup a') - b(\mathcal{O} \cup a') \to -\infty \iff \sigma(f(\mathcal{O} \cup a') - b(\mathcal{O} \cup a')) \to 0, \tag{3.12}$$

where $b(\cdot)$ is a fixed baseline function and $a' \in \mathcal{A}, a' \neq a_\star$. For implementation, $b(\cdot)$ could be either a randomly initialized network or a constant. Since the two settings do not produce significantly different results in our preliminary experiments, we set $b(\cdot)$ to be a constant to reduce computation.

We then optimize the network to maximize the following objective as done in [GH10]:

$$\ell = \log(\sigma(f(\mathcal{O} \cup a_\star) - b(\mathcal{O} \cup a_\star))) + \sum_{a' \in \mathcal{A}, a' \neq a_\star} \log(1 - \sigma(f(\mathcal{O} \cup a') - b(\mathcal{O} \cup a'))). \tag{3.13}$$

Connection to NCE If we treat the baseline as the negative potential of a fixed noise model of the same Gibbs form and ignore the difference between the partition functions, Equation 3.11 and Equation 3.12 become the G function used in NCE [GH10]. But unlike NCE, we do not need to multiply the size ratio in the sigmoid function [DL17].

3.2.3.2 Perceptual Inference

As indicated in [CJS90], a mere perceptive model for RPM is arguably not enough. Therefore, we propose to incorporate a simple inference subsystem into the model: the inference branch should be responsible for inferring the hidden rules in the problem. Specifically, we assume there are at most N attributes in each problem, each of which is subject to the governance of one of M rules. Then hidden rules $\mathcal{T}$ in one problem instance can be decomposed into

$$p(\mathcal{T} | \mathcal{O}) = \prod_{i=1}^{N} p(t_i | \mathcal{O}), \tag{3.14}$$

where $t_i = 1 \dots M$ denotes the rule type on attribute n_i. For the actual form of the probability of rules on each attribute, we propose to model it using a multinomial distribution. This assumption is consistent with the way datasets are usually generated [ZGJ19, BHS18, WS15]: one rule is independently picked from the rule set for each attribute. In this way, each rule could also be regarded as a basis in a rule dictionary and jointly learned, as done in active basis [WSG10] or word embedding [MSC13, PSM14].

If we treat rules as hidden variables, the log probability in Equation 3.9 can be decomposed into

$$\log p(a|\mathcal{O}) = \log \sum_{\mathcal{T}} p(a|\mathcal{T}, \mathcal{O}) p(\mathcal{T}|\mathcal{O}) = \log \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T}|\mathcal{O})}[p(a|\mathcal{T}, \mathcal{O})]. \tag{3.15}$$

Note that writing the summation in the form of expectation affords sampling algorithms, which can be done on each individual attribute due to the independence assumption.

In addition, if we model $p(\mathcal{T}|\mathcal{O})$ as an inference branch $g(\cdot)$ and sample only once from it, the model can be modified into $f(\mathcal{O} \cup a, \hat{\mathcal{T}})$ with $\hat{\mathcal{T}}$ sampled from $g(\mathcal{O})$. Following the same derivation above, we now optimize the new objective:

$$\ell = \log(\sigma(f(\mathcal{O} \cup a_\star, \hat{\mathcal{T}}) - b(\mathcal{O} \cup a_\star))) + \sum_{a' \in \mathcal{A}, a' \neq a_\star} \log(1 - \sigma(f(\mathcal{O} \cup a', \hat{\mathcal{T}}) - b(\mathcal{O} \cup a'))). \tag{3.16}$$

To sample from a multinomial, we could either use hard sampling like Gumbel-SoftMax [JGP16, MMT16] or a soft one by taking expectation. We do not observe significant difference between the two settings.

The expectation in Equation 3.15 is proposed primarily to make the computation of the exact log probability controllable and tractable: while the full summation requires $O(M^N)$ passes of the model, a Monte Carlo approximation of it could be calculated in $O(1)$ time. We also note that if $p(\mathcal{T}|\mathcal{O})$ is highly peaked (*e.g.*, ground truth), the Monte Carlo estimate could be accurate as well. Despite the fact that we only sample once from an inference branch to reduce computation, we find in practice the Monte Carlo estimate works quite well.

3.2.3.3 Architecture

Combining contrasting, perceptual inference, and permutation invariance, we propose a new network architecture to solve the challenging RPM problem, named *Contrastive Perceptual Inference* network (CoPINet). The perception branch is composed of a common feature encoder and shared interweaving contrast modules and residual blocks [HZR16]. The encoder first extracts image features independently for each panel and sum ones in the corresponding rows and columns before the final transformation into a latent space. The inference branch consists of the same encoder and a (Gumbel-)SoftMax output layer. The sampled results will be transformed and concatenated channel-wise into the summation in Equation 3.7. In our implementation, we prepend each residual block with a contrast module; such a combination can be repeated while keeping the network permutation-invariant. The network finally uses an MLP to produce a negative potential for each observation and candidate pair and is trained using Equation 3.16; see Figure 3.6(b) for a graphical illustration of the entire CoPINet architecture.

3.2.4 Experiments

3.2.4.1 Experimental Setup

We verify the effectiveness of our models on two major RPM datasets: RAVEN [ZGJ19] and PGM [BHS18]. Across all experiments, we train models on the training set, tune hyper-parameters on the validation set, and report the final results on the test set. All of the models are implemented in PyTorch [PGC17] and optimized using ADAM [KB14]. While a good performance of WReN [BHS18] and ResNet+DRT [ZGJ19] relies on external supervision, such as rule specifications and structural annotations, the proposed model achieves better performance with only $\mathcal{O}$, $\mathcal{A}$, and $a_\star$. Models are trained on servers with four Nvidia RTX Titans. For the WReN model, we use a public implementation that reproduces results in [BHS18][2]. We implement our models in PyTorch [PGC17] and optimize using ADAM [KB14]. During training, we perform early-stop based on validation loss. We use the same network architecture and hyper-parameters in both RAVEN and PGM experiments.

3.2.4.2 Results on RAVEN

There are $70,000$ problems in the RAVEN dataset [ZGJ19], equally distributed in 7 figure configurations. In each configuration, the dataset is randomly split into 6 folds for training, 2 folds for validation, and 2 folds for testing. We compare our model with several simple baselines (LSTM [HS97], CNN [HW17], and vanilla ResNet [HZR16]) and two strong baselines (WReN [BHS18] and ResNet+DRT [ZGJ19]). Model performance is measured by accuracy.

General Performance on RAVEN In this experiment, we train the models on all $42,000$ training samples and measure how they perform on the test set. The first part of Table 3.6 shows the testing accuracy of all models. We also retrieve the performance of humans and a solver with perfect information from [ZGJ19] for comparison. As shown in the table, the proposed model CoPINet achieves the best performance among all the models we test. For the relational model WReN proposed in [BHS18], we run the tests on a permutation-invariant version, *i.e.*, one without positional tagging (NoTag), and tune the model also to minimize an auxiliary loss (Aux) [BHS18]. While the auxiliary loss could boost the performance of WReN as we will show later in the ablation study, we do not observe similar effects on CoPINet. As indicated in the detailed comparisons in Table 3.6, WReN is biased towards images of grid configurations and does poorly on ones demanding compositional reasoning, *i.e.*, ones with independent components. We further note that compared to previously proposed models (WReN [BHS18] and ResNet+DRT [ZGJ19]), CoPINet does not require additional

[2] `https://github.com/Fen9/WReN`

information such as structural annotations and meta targets and still shows human-level performance in this task. When comparing the performance of CoPINet and human on specific figure configurations, we notice that CoPINet is inferior in learning samples of grid-like compositionality but efficient in distinguishing images consisting of multiple components, implying the efficiency of the contrasting mechanism.

Ablation Study One problem of particular interest in building CoPINet is how each component contributes to performance improvement. To answer this question, we measure model accuracy by gradually removing each construct in CoPINet, *i.e.*, the perceptual inference branch, the contrast loss, and the contrast module. In the second part of Table 3.6, we show the results of ablation on CoPINet. Both the full model (CoPINet) and the one without the perceptual inference branch (CoPINet-Contrast-CL) could achieve human-level performance, with the latter slightly inferior to the former. If we further replace the contrast loss with the cross-entropy loss (CoPINet-Contrast-XE), we observe a noticeable performance decrease of around 4%, verifying the effectiveness of the contrast loss. A catastrophic performance

Method	Acc	Center	2x2Grid	3x3Grid	L-R	U-D	O-IC	O-IG
LSTM	13.07%	13.19%	14.13%	13.69%	12.84%	12.35%	12.15%	12.99%
WReN-NoTag-Aux	17.62%	17.66%	29.02%	34.67%	7.69%	7.89%	12.30%	13.94%
CNN	36.97%	33.58%	30.30%	33.53%	39.43%	41.26%	43.20%	37.54%
ResNet	53.43%	52.82%	41.86%	44.29%	58.77%	60.16%	63.19%	53.12%
ResNet+DRT	59.56%	58.08%	46.53%	50.40%	65.82%	67.11%	69.09%	60.11%
CoPINet	**91.42%**	**95.05%**	**77.45%**	**78.85%**	**99.10%**	**99.65%**	**98.50%**	**91.35%**
WReN-NoTag-NoAux	15.07%	12.30%	28.62%	29.22%	7.20%	6.55%	8.33%	13.10%
WReN-Tag-NoAux	17.94%	15.38%	29.81%	32.94%	11.06%	10.96%	11.06%	14.54%
WReN-Tag-Aux	33.97%	58.38%	38.89%	37.70%	21.58%	19.74%	38.84%	22.57%
CoPINet-Backbone-XE	20.75%	24.00%	23.25%	23.05%	15.00%	13.90%	21.25%	24.80%
CoPINet-Contrast-XE	86.16%	87.25%	71.05%	74.45%	97.25%	97.05%	93.20%	82.90%
CoPINet-Contrast-CL	90.04%	94.30%	74.00%	76.85%	99.05%	99.35%	98.00%	88.70%
Human	84.41%	95.45%	81.82%	79.55%	86.36%	81.81%	86.36%	81.81%
Solver	100%	100%	100%	100%	100%	100%	100%	100%

Table 3.6: Testing accuracy of models on RAVEN. Acc denotes the mean accuracy of each model. Same as in [ZGJ19], L-R denotes the Left-Right configuration, U-D Up-Down, O-IC Out-InCenter, and O-IG Out-InGrid.

downgrade of 66% is observed if we remove the contrast module, leaving only the network backbone (CoPINet-Backbone-XE). This drastic performance gap shows that the functional constraint on modeling an explicit contrasting mechanism is arguably a crucial factor in machines' reasoning ability as well as in humans'. The ablation study shows that all the three proposed constructs, especially the contrast module, are critical to the performance of CoPINet. We also study how the requirement of permutation invariance and auxiliary training affect the previously proposed WReN. As shown in Table 3.6, sacrificing the permutation invariance (Tag) provides the model a huge upgrade during auxiliary training (Aux), compared to the one without tagging (NoTag) and auxiliary loss (NoAux). This effect becomes even more significant on the PGM dataset, as we will show in subsubsection 3.2.4.3.

Dataset Size and Performance Even though CoPINet surpasses human performance on RAVEN, this competition is inherently unfair, as the human subjects in this study never experience such an intensive training session as our model does. To make the comparison fairer and also as a step towards a model capable of human learning efficiency, we further measure how the model performance changes as the training set size shrinks. To this end, we train our CoPINet on subsets of the full RAVEN training set and test it on the full test set. As shown on Table 3.8 and Table 3.7, the model performance varies roughly log-linearly with the training set size. One surprising observation is: with only half of the amount of the data, we could already achieve human-level performance. On a training set 16× smaller, CoPINet outperforms all previous models. And on a subset 64× smaller, CoPINet already outshines WReN.

3.2.4.3 Results on PGM

We use the neutral regime of the PGM dataset for model evaluation due to its diversity and richness in relationships, objects, and attributes. This split of the dataset has in total 1.42 million samples, with 1.2 million for training, $2,000$ for validation, and $200,000$ for testing.

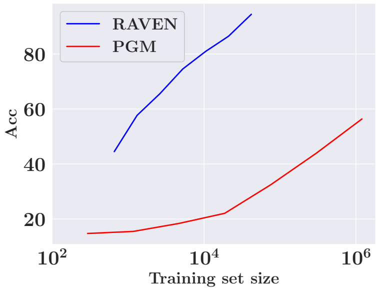


Table 3.7: CoPINet on RAVEN and PGM as the training set size shrinks.

Training set size	Acc
658	44.48%
1,316	57.69%
2,625	65.55%
5,250	74.53%
10,500	80.92%
21,000	86.43%

Table 3.8: Model performance under different training set sizes on RAVEN dataset. The full training set has 42,000 samples.

Training set size	Acc
293	14.73%
1,172	15.48%
4,688	18.39%
18,750	22.07%
75,000	32.39%
300,000	43.89%

Table 3.9: Model performance under different training set sizes on PGM dataset. The full training set has 1.2 million samples.

We train the models on the training set, tune the hyperparameters on the validation set, and evaluate the performance on the test set. We compare our models with baselines set up in [BHS18], *i.e.*, LSTM, CNN, ResNet, Wild-ResNet, and WReN. As ResNet+DRT proposed in [ZGJ19] requires structural annotations not available in PGM, we are unable to measure its performance. Again, all performance is measured by accuracy. Due to the lack of further stratification on this training regime, we only report the final mean accuracy.

General Performance on PGM In this experiment, we train the models on all 1.2 million training samples and report performance on the entire test set. As shown in Table 3.10, CoPINet achieves the best performance among all permutation-invariant models, setting a new state-of-the-art on this dataset. Similar to the setting in RAVEN, we make the previously proposed WReN permutation-invariant by removing the positional tagging (NoTag) and train it with both cross-entropy loss and auxiliary loss (Aux) [BHS18]. The auxiliary loss could boost the performance of WReN. However, in coherence with the study on RAVEN and a previous work [ZGJ19], we notice that the auxiliary loss does not help our CoPINet. It is worth noting that while WReN demands additional training supervision from meta targets to reach the performance, CoPINet only requires basic annotations of ground truth indices $a_\star$ and achieves better results.

Method	CNN	LSTM	ResNet	Wild-ResNet	WReN-NoTag-Aux	CoPINet
Acc	33.00%	35.80%	42.00%	48.00%	49.10%	**56.37%**

Table 3.10: Testing accuracy of models on PGM. Acc denotes the mean accuracy of each model.

Ablation Study We perform ablation studies on both WReN and CoPINet to see how the requirement of permutation invariance affects WReN and how each module in CoPINet contributes to its superior performance. The notations are the same as those used in the ablation study for RAVEN. As shown in the first part of Table 3.11, adding a proper auxiliary loss does provide WReN a 10% performance boost. However, additional supervision is required. Making the model permutation-sensitive gives the model a significant benefit by up to a 28% accuracy increase; however, it also indicates that WReN learns to shortcut the solutions by coding the positional association, instead of truly understanding the differences among distinctive choices and their potential effects on the compatibility of the entire matrix. The second part of Table 3.11 demonstrates how each construct contributes to the performance improvement of CoPINet on PGM. Despite the smaller enhancement of the contrast loss compared to that in RAVEN, the upgrade from the contrast module for PGM is still significant, and the perceptual inference branch keeps raising the final performance. In accordance with the ablation study on the RAVEN dataset, we show that all the proposed components contribute to the final performance increase.

Method	WReN-NoTag-NoAux	WReN-NoTag-Aux	WReN-Tag-NoAux	WReN-Tag-Aux
Acc	39.25%	49.10%	62.45%	77.94%
Method	CoPINet-Backbone-XE	CoPINet-Contrast-XE	CoPINet-Contrast-CL	CoPINet
Acc	42.10%	51.04%	54.19%	56.37%

Table 3.11: Ablation study on PGM.

Dataset Size and Performance Motivated by the idea of fairer comparison and low-shot reasoning, we also measure how the performance of the proposed CoPINet changes as the training set size of PGM varies. Specifically, we train CoPINet on subsets of the PGM training set and test it on the entire test set. As shown in Table 3.9 and Table 3.7, CoPINet performance on PGM varies roughly log-exponentially with respect to the training set size. We further note that when trained on a 16× smaller dataset, CoPINet already achieves results similar to CNN and LSTM.

3.2.5 Conclusion and Discussion

In this work, we aim to improve machines' reasoning ability in "thinking in pictures" by jointly learning perception and inference via contrasting. Specifically, we introduce the contrast module, the contrast loss, and the joint system of perceptual inference. We also require our model to be permutation-invariant. In a typical and challenging task of this kind, Raven's Progressive Matrices (RPM), we demonstrate that our proposed model—*Contrastive Perceptual Inference* network (CoPINet)—achieves the new state-of-the-art for permutation-invariant models on two major RPM datasets. Further ablation studies show that all the three proposed components are effective towards improving the final results, especially the contrast module. It also shows that the permutation invariance forces the model to understand the effects of different choices on the compatibility of an entire RPM matrix, rather than remembering the positional association and shortcutting the solutions.

While it is encouraging to see the performance improvement of the proposed ideas on two big datasets, it is the last part of the experiments, *i.e.*, dataset size and performance, that really intrigues us. With infinitely large datasets that cover the entirety of an arbitrarily complex problem domain, it is arguably possible that a simple over-parameterized model could solve it. However, in reality, there is barely any chance that one would observe all the domain, yet humans still learn quite efficiently how the hidden rules work. We believe this is the core where the real intelligence lies: learning from only a few samples and generalizing to the extreme. Even though CoPINet already demonstrates better learning efficiency, it would be ideal to have models capable of few-shot learning in the task of RPM. Without massive datasets, it would be a real challenge, and we hope this work could call for future research into it.

Performance, however, is definitely not the end goal in the line of research on relational and analogical visual reasoning: other dimensions for measurements include generalization, generability, and transferability. Is it possible for a model to be trained on a single configuration and generalize to other settings? Can we generate the final answer based on the given context panels, in a similar way to the top-down and bottom-up method jointly applied by humans for reasoning? Can we transfer the relational and geometric knowledge required in the reasoning task from other tasks? Questions like these are far from being answered. While [ZGJ19] show in the experiments that neural models do possess a certain degree of generalizability, the testing accuracy is far from satisfactory. In the meantime, there are a plethora of discriminative approaches towards solving reasoning problems in question answering, but generative methods and combined methods are lacking. The relational and analogical reasoning was initially introduced as a way to measure a human's intelligence, without training humans on the task. However, current settings uniformly reformulate it as a learning problem rather than a transfer problem, contradictory to why the task was started. Up to now, there has been barely any work that measures how knowledge on another task could be transferred to this one. We believe that significant advances in these dimensions would possibly enable Artificial Intelligence (AI) models to go beyond data fitting and acquire symbolized knowledge.

While modern computer vision techniques to solve Raven's Progressive Matrices (RPM) are based on neural networks, a promising ingredient is nowhere to be found: Gestalt psychology. Traces of the perceptual grouping and figure-ground organization are gradually faded out in the most recent wave of deep learning. However, the principles of grouping, both classical (*e.g.*, proximity, closure, and similarity) and new (*e.g.*, synchrony, element, and uniform connectedness) play an essential role in RPM, as humans arguably solve these problems by first figuring out groups and then applying the rules. We anticipate that modern deep learning methods integrated with the tradition of conceptual and theoretical foundations of the Gestalt approach would further improve models on abstract reasoning tasks like RPM.

3.3 Visual Language Reasoning with Commonsense Knowledge

Outside-knowledge visual question answering (OK-VQA) requires the agent to comprehend the image, make use of relevant knowledge from the entire web, and digest all the information to answer the question. Most previous works address the problem by first fusing the image and question in the multi-modal space, which is inflexible for further fusion with a vast amount of external knowledge. In this work, we call for an alternative paradigm for the OK-VQA task, which transforms the image into plain text, so that we can enable knowledge passage retrieval, and generative question-answering in the natural language space. This paradigm takes advantage of the sheer volume of gigantic knowledge bases and the richness of pre-trained language models. A **T**ransform-**R**etr**i**eve-**G**enerate framework (**TRiG**) framework is proposed[3], which can be plug-and-played with alternative image-to-text models and textual knowledge bases. Experimental results show that our **TRiG** framework outperforms all state-of-the-art supervised methods by at least 11.1% absolute margin.

3.3.1 Introduction

The visual question answering (VQA) task is to provide a natural language answer to a natural language question given an image[AAL15]. This task has been well studied in the research communities, and numerous cross-modal methods have achieved state-of-the-art performance [SNS19, YYC19, JMR20, GXT21, YLL20, LBP19, CLY19, LYL20, ZLH21, LGN20]. The knowledge-based visual question answering (KB-VQA) task requires more extensive learning since the questions can be answered only by referring to external general knowledge[SMY19,

[3] The code of this work will be made public.

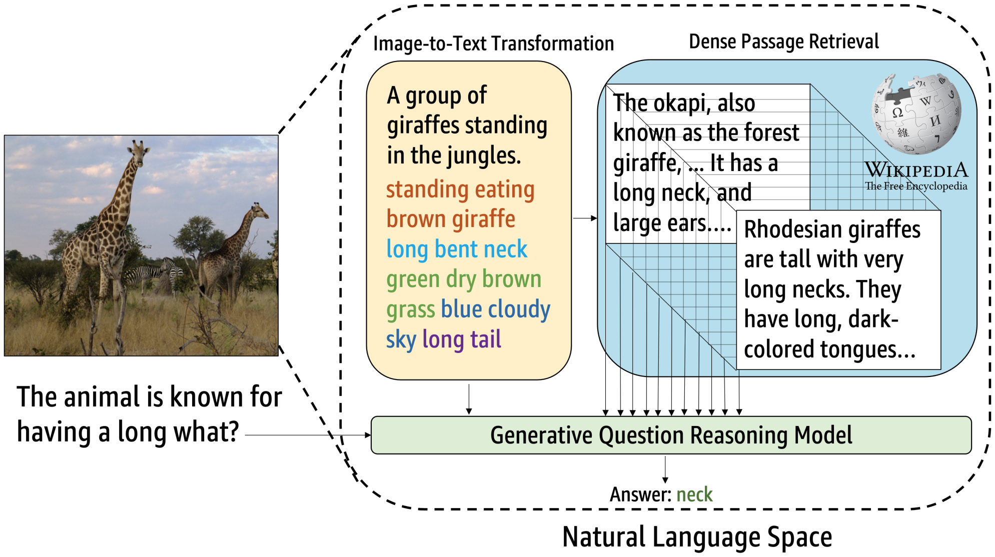


Image-to-Text Transformation
Dense Passage Retrieval
A group of giraffes standing in the jungles.
standing eating brown giraffe
long bent neck
green dry brown grass blue cloudy sky long tail
The okapi, also known as the forest giraffe, ... It has a long neck, and large ears....
Rhodesian giraffes are tall with very long necks. They have long, dark-colored tongues...
WIKIPEDIA
The Free Encyclopedia
The animal is known for having a long what?
Generative Question Reasoning Model
Answer: neck
Natural Language Space

Figure 3.7: An intuitive example of our TRiG framework on the OK-VQA problem. Our Framework transform all information into language space and performs retrieved-based question answering through generative language models.

WWS15, WWS17, LJZ18, SMY19]. Most KB-VQA datasets come with pre-defined knowledge bases, and each question is annotated with at least one supporting knowledge fact. Moreover, the recently proposed outside-knowledge visual question answering (OK-VQA) task is the most open in the sense that any external knowledge can be used to answer the questions.

Consider the example in Figure 3.7. As a human, one needs to first identify objects like *giraffes* and *trees* in the image, and associate the *giraffes* to the word *animal* in the question. Second, the human needs to apply his/her acquired commonsense knowledge about *giraffe*'s characteristics and answer the question that *giraffe* is known for having a *long neck*. For machine learning models to solve the same problem, there are several unique challenges. First, in order to answer such a question, one has to align the image, the question, and the vast amount of knowledge passages into one common space. One solution is to first fuse the image and question information in the multi-modal space with pre-trained vision-language models, and then inject knowledge into the multi-modal space. Most previous work on OK-VQA follow this paradigm, including directly injecting the knowledge embeddings [GZA20, STD21] and fusing the output of a vision-language model with the knowledge graph through graph convolutional network[MCP21]. However, this paradigm is at the cost of squeezing the rich representation of the textual knowledge, in the magnitude of hundreds of millions, into a much smaller multi-modal space. Comparing to knowledge corpus such as BookCorpus (800M words) and English Wikipedia (2,500M words), multi-modal pretraining datasets are much smaller such as Visual Genome with 0.01 million images and less than 2 million question-answer pairs[KZG17], which leads to less knowledge. Therefore, we argue that it is possible to transform everything into the language space first, and then take advantage of the tremendous amount of textual knowledge for question answering. Although this seems counter-intuitive, our work proves its advantage. In this paradigm, the challenge is to be able to transform the image into language with minimum information loss. In order to tackle this, we propose three-level image-to-text transformations which significantly outperform baselines that use only captions or object labels.

The second challenge of the OK-VQA task is how to effectively retrieve the most relevant knowledge passages from gigantic knowledge bases. Previous work has explored various retrieval methods such as term-based BM25[LZB21], and network-based ranking [LZB21, WLS21]. In the OK-VQA dataset, this task is problematic in that there is no ground-truth knowledge annotation for each question. The retrieval has to rely on either transfer learning from similar knowledge-retrieval tasks or weak supervision from pseudo signals such as whether the passage contains the answer tokens[QZY21]. Our preliminary study finds that there is no guarantee that a passage containing the ground-truth answer will essentially relate to the question or help the answer prediction. Such signals are very weak and may introduce more noise than useful information into the retrieval model. Instead, we adopt the state-of-the-art dense passage retrieval model (DPR) [KOM20] that is pre-trained on large question-answering dataset *Natural Questions* (NQ) [KPR19] as our knowledge retriever, which is shown to outperform the BM25 method in terms of retrieval coverage rate.

The third challenge of the OK-VQA task is to consolidate all the multi-source input, namely the question, visual context, and the retrieved knowledge passages, to predict answers. Since now everything is in the language space, the problem can be formulated as a multi-passage question answering problem. More specifically, the model needs to not only rank the retrieved passages but also predict an answer according to the ranked passages. Most existing work utilizes extractive methods to predict the answer span in the passage [RZL16, CFW17, CG17, RJL18, WNM19, LCT19, YXL19]. This is not applicable in the OK-VQA dataset because there is neither annotation of ground-truth passage nor answer span in any passage. Instead, we use the generative question answering model [IG20b] to avoid the defect in span prediction. Furthermore, we use beam-search for robust answer generation. Lastly, since the question-answering model is the last stage in the entire framework, any information distortion or loss in the image-to-text transformation and knowledge retrieval would propagate to the final question answering model. Therefore, it is important for the final question answering model to be more transparent and interpretable to diagnose the root cause of errors. We use cross-attention scores from the decoder of the generative model to rank and highlight the top supporting knowledge passages, which helps to interpret the results of the model.

To bridge the above-mentioned research gaps, we propose the **T**ransform-**R**etr**i**eve-**G**enerate (TRiG) framework for the OK-VQA task. At the high level, the framework aligns all the information (image, question, and knowledge) into the language space in order to take advantage of the rich semantics of textual knowledge. The framework starts with three-level image-to-text transformations, followed by dense passage retrieval to retrieve the most relevant knowledge passages. Further, the TRiG aggregates the information from all passages and generates an answer that is relatively easy to interpret. Our contributions are as follows:

- We propose a new paradigm shift for the OK-VQA task, from aligning all the information in the multi-modal space, to first transforming an image into plain text and performing knowledge retrieval and question answering all in language space.

- We propose a robust framework **T**ransform-**R**etr**i**eve-**G**enerate (TRiG), that achieves new state-of-the-art performance on the OK-VQA dataset and leading other supervised methods by 11.1%.

3.3.2 Related Work

Visual Question Answering (VQA) The conventional visual question answering (VQA) task aims to answer questions pertaining to a given image. Multiple VQA datasets have been proposed, such as Visual Genome QA[KZG16] VQA [AAL15], GQA[HM19], CLEVR[JHM17a], MovieQA[TZS16] and so on. Many works have shown state-of-the-art performance on VQA tasks, including task-specific VQA models with various cross-modality fusion mechanisms [SNS19, YYC19, KJZ18, YYX18, JMR20, GXT21, YLL20] and joint vision-language models that are pretrained on large-scale vision-language corpus and finetuned on VQA tasks [LBP19, CLY19, LYL20, ZLH21, LGN20, TB19, GCL20]. Please note that the conventional VQA task does not require external knowledge by definition, although studies show some VQA questions may require commonsense knowledge to answer correctly [AAL15].

Outside Knowledge-Based VQA (OK-VQA) Beyond the above paradigm, knowledge-based visual question answering (KB-VQA) is proposed where a visual question cannot be answered without external knowledge. Several knowledge-based VQA datasets are proposed, each providing its own knowledge bases and ground-truth supporting fact [SMY19, WWS17, LJZ18]. More recently, the dataset outside-knowledge visual question answering (OK-VQA)[MRF19] is proposed where the usage of outside knowledge is open to the entire web. Most existing work for OK-VQA rely on the pre-trained vision-language models as a major workhorse for question answering[GZA20, STD21, MCP21, WLS21, LZB21, YGW21]. In [GZA20, STD21], learned knowledge embeddings are injected into vision-language models to perform knowledge-aware question answering. Other work uses vision-language models as a knowledge-free VQA model first and later adjusts the predicted answers by fusion

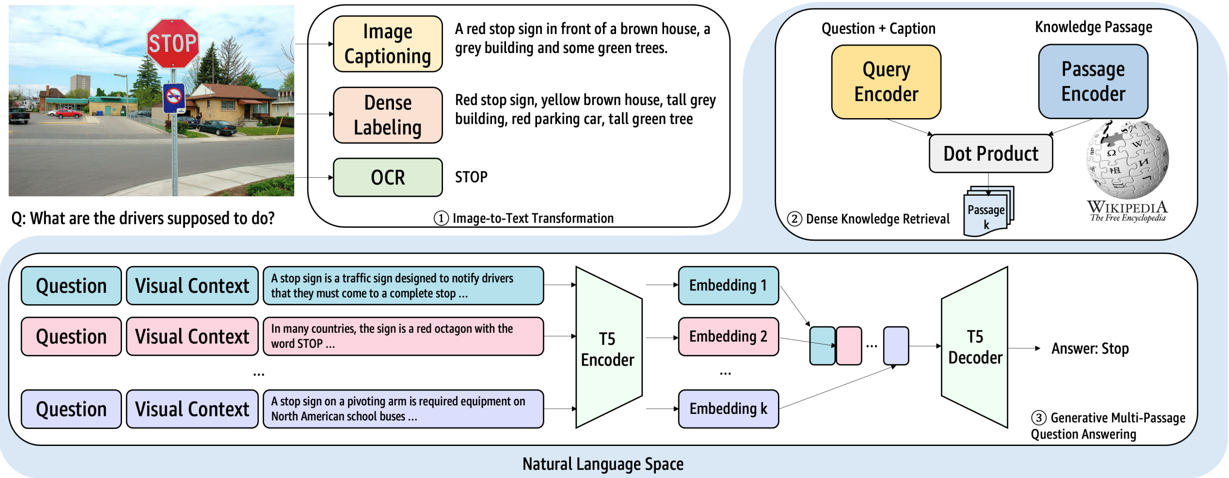

STOP
Q: What are the drivers supposed to do?
Image Captioning
A red stop sign in front of a brown house, a grey building and some green trees.
Dense Labeling
Red stop sign, yellow brown house, tall grey building, red parking car, tall green tree
OCR
STOP
① Image-to-Text Transformation
Question + Caption
Knowledge Passage
Query Encoder
Passage Encoder
Dot Product
Passage k
WIKIPEDIA The Free Encyclopedia
② Dense Knowledge Retrieval
Question
Visual Context
A stop sign is a traffic sign designed to notify drivers that they must come to a complete stop ...
Question
Visual Context
In many countries, the sign is a red octagon with the word STOP ...
...
Question
Visual Context
A stop sign on a pivoting arm is required equipment on North American school buses ...
T5 Encoder
Embedding 1
Embedding 2
...
Embedding k
T5 Decoder
Answer: Stop
③ Generative Multi-Passage Question Answering
Natural Language Space

Figure 3.8: The overview of our **TRiG** framework. (1) **T**: Our TRiG framework transforms all visual information into natural language space on three-levels: image-level captioning, object-level dense labeling and text OCR. (2) **R**: Our dense knowledge retriever retrieve top-k knowledge passages from Wikipedia that are relevant to the query. (3) **G**: Our generative question answering model encode all question-context-knowledge tuples and fuses the output to generate a final answer.

with knowledge graphs [MCP21] or answer validation with knowledge text [WLS21]. Some also propose to directly learn vision-language representation for dense knowledge retrieval [LZB21]. Different from the above, one recent work proposes to first convert the image into text caption and tags and then perform prompt-based QA on GPT-3 model purely in the language space [YGW21]. In addition, a concurrent work [GWH21] takes advantages of GPT-3 to retrieve implicit knowledge. However, the accessibility to this super-large-scale pre-trained language model is restricted, and it is challenging to interpret the QA result from the generative GPT-3 model.

Open-Domain Question Answering in NLP Open-domain question answering (Open-Domain QA) has been popular in the NLP community in recent years. The task is to answer a question with external knowledge bases without any given context paragraphs[RJL18]. There are mainly two streams of approaches, namely knowledge graph-based question answering [SDZ18, LCC19, WKM19, FCL20, LGX20, YRB21] and knowledge retrieval-based question answering[RZL16, CFW17, CG17, RJL18, WNM19, LCT19, YXL19, IG20b]. For retrieval-based methods, both elastic-search such as BM25[RZ09] and semantic search such as Dense Passage Retrieval (DPR)[KOM20] are utilized to retrieve most relevant knowledge snippets from knowledge bases. For question answering, most existing work adopt extractive methods to predict the span of an answer in knowledge snippets [RZL16, CFW17, CG17, RJL18, WNM19, LCT19, YXL19]. One most recent work proposes to use generative language models for knowledge-based QA, which achieves state-of-the-art performances [IG20b].

3.3.3 Methodology

In this section, we introduce the details of our **T**ransform-**R**etr**i**eve-**G**enerate (**TRiG**) framework. Shown in Figure 3.8, our framework contains three stages: (i) image-to-text transformation, (ii) knowledge passage retrieval, (iii) multi-passages open-domain question answer generation.

3.3.3.1 Image-to-Text Transformation

Contrary to existing work, we first transform the image into text and then perform all downstream tasks in the language space. In order to minimize the information loss in the process of transforming the image into plain text, three-levels of transformations are performed (Equation 3.17). First, image-level information is transformed to caption text with a state-of-the-art image captioning model[LYL20]. Second, object-level information is translated to object and attribute labels[AHB18, HYH21]. Lastly, according to [JKK21], some VQA questions can only be answered with optical character recognition (OCR). We use an off-the-shelf OCR model to detect all possible texts in the images [4].

We denote C_i, L_i, and O_i as the generated caption text, attribute and object text, and OCR text from image I_i respectively. In the rest of the section, we will denote the visual

[4] https://github.com/JaidedAI/EasyOCR

context $v_i = (C_i, L_i, O_i)$ for the corresponding image I_i. Please note that our proposed framework does not necessitate the use of the above-mentioned image-to-text transformation models only. One could choose to plug-and-play alternative methods into the framework.

$$\begin{aligned} C_i &= (w_0^c, \ldots, w_j^c) \leftarrow f_{ImageCaptioning}(I_i) \\ L_i &= \{(w_0^{attr}, w_0^{obj}), \ldots, (w_n^{attr}, w_m^{obj})\} \leftarrow f_{tagging}(I_i) \\ O_i &= \{w_0^{ocr}, \ldots, w_k^{ocr}\} \leftarrow f_{ocr}(I_i) \end{aligned} \tag{3.17}$$

3.3.3.2 Knowledge Passage Retrieval

After the image is transformed into plain-text representation, we use the text representation as the query to retrieve knowledge passages in the natural language space. In this work, we use the Wikipedia dump as the knowledge base, which contains over 21 million Wikipedia passages [LCT19]. We ensure that our framework is designed to be generic enough to support other textual KBs such as GenericsKB[BAC20] or the surface forms of graph knowledge bases such as ConceptNet[SCH17].

More specifically, given a textual query q_i of an image I_i and a knowledge base $\mathbb{K} = \{p_j\}$ where each p_j is a knowledge passage, the task is to retrieve top k knowledge passages $P_k = [p_1, p_2, \ldots, p_k]$ from $\mathbb{K}$ that are most relevant to the query q_i, where $k \ll |\mathbb{K}|$. In this work, we empirically use the query $q_i = (Q_i, C_i)$, where Q_i is the original question and C_i is the the generated caption of corresponding image I_i.

We use dense passage retrieval (DPR) to retrieve the knowledge passages [KOM20]. DPR encodes both query and passage with BERT layers that could better capture the semantic similarity between them than term-based retrieval methods such as TF*IDF and BM25[KOM20]. First, the query q_i and a passage p_k are encoded with two independent pre-trained BERT encoders [DCL18]. We take the embedding of the [CLS] token $\mathbf{x}_{q_i}$ and $\mathbf{x}_{p_i}$ in the BERT to represent q_i and p_k respectively. Second, a similarity scores $\text{sim}(q_i, p_k)$ is calculated by taking the dot product of the two encoded dense vectors of the query q_i and a passage p_k.

$$\mathbf{x}_{q_i} = E_Q(q_i), \mathbf{x}_{p_i} = E_P(p_k) \tag{3.18}$$

$$\text{sim}(q_i, p_k) = \mathbf{x}_{q_i}^T \cdot \mathbf{x}_{p_k} \tag{3.19}$$

Because of the tremendous amount of passages in the Wikipedia knowledge base, it is time-consuming to retrieve the top k passages for each query from the knowledge base with over 21 million passages. We leverage an open-sourced indexing engine FAISS[JDJ17], an extremely efficient library to speed up the clustering and indexing of large number of dense vectors. Given a query q_i, the dense passage retrieval module will return k passages $P_k = [p_1, p_2, \ldots, p_k]$ from the entire knowledge base $\mathbb{K}$ where $\text{sim}(q_i, p_1) > \text{sim}(q_i, p_2) > \cdots > \text{sim}(q_i, p_k)$ and $k \ll |\mathbb{K}|$. The retrieved passages P_k will be later used for downstream question-answering.

3.3.3.3 Generative Multi-Passages QA

After aligning the visual information, the question, and the external knowledge into the language space, we introduce our generative question-answering module. Our design of the model takes the following into consideration. First, although previous work on joint vision-language models formulates the task as an answer classification task [SCH17, MCP21, WLS21], our preliminary studies show that language models seem to be less flexible in classifying text into such high-dimensional answer space (over 100k) given a relatively small dataset. Second, although most previous language QA models follow a span-based answer prediction paradigm [RJL18, WNM19, LCT19, YXL19], it is impractical in our open-domain setting since there is no ground-truth supporting fact in our task, let alone the ground-truth answer span for prediction. On the other hand, recent work shows that a generative encoder-decoder network can achieve state-of-the-art performance on multiple open-domain QA datasets [RSR19], and it avoids span prediction and directly generates a free-form answer.

To achieve this goal, we use a transformer-like encoder-decoder model T5 as the backbone of our generative question answering module[RSR20]. It is impractical to include all top-k passages in one T5 model. We use T5 model to encode each $(question, visual\ context, knowledge)$ tuple independently and then fuse the k encoded representations to decode an answer following the idea in [IG20b].

Multi-Passages Question Answer Generation First, we feed the concatenated sequence of $(Q_i, v_i, p_{i,k})$ into a self-attentive encoder to get per-position hidden embeddings $\mathbf{z}^{Q_{i,k}}$, where q_i is the question, v_i is the visual context text and p_i is one passage respectively.

$$\begin{aligned} \mathbf{z}^{Q_{i,k}} &= E_{SelfAttn}(Q_i, v_i, p_{i,k}) \\ &= (z_0, \ldots, z_L) \end{aligned} \tag{3.20}$$

where z_i is the hidden embedding of the i-th token in the sequence, $\mathbf{z}^{Q_{i,k}} \in \mathbb{R}^{1 \times L \times h}$ is the hidden representation of the sequence, $L = |(Q_i, v_i, p_{i,k})|$ is the length of the sequence and h is the size of the hidden embedding.

Subsequently, we perform the same encoding operation on all k passages to derive k hidden representations:

$$\mathbf{z}^{Q_i} = \left(\mathbf{z}^{Q_{i,1}}, \ldots, \mathbf{z}^{Q_{i,k}}\right) \tag{3.21}$$

where we concatenate the k hidden embeddings to $\mathbf{z}^{Q_i} \in \mathbb{R}^{(k \cdot L) \times h}$. This operation is to fuse all the information from different question-context-passage interactions together in order to generate better answers. Then, we feed the concatenated hidden representation $\mathbf{z}^{Q_i}$ into a stacked self-attentive decoder to predict per-position word distribution over the vocabulary space $|V|$:

$$P(a_1), \ldots, P(a_l) = \sigma(D_{SelfAttn}(\mathbf{z}^{Q_i})) \tag{3.22}$$

where σ is a non-linear function such as softmax, l is the length of the answer, and $Q_i \in \mathbb{R}^{|V|}$ is the word distribution over the vocabulary of size $|V|$. Finally, we use teacher-enforcing to train the entire model with auto-regressive cross-entropy loss:

$$L_{ans} = -\frac{1}{N \cdot l \cdot |V|} \cdot \sum_{i=1}^{N} \sum_{j=1}^{l} \sum_{w=1}^{|V|} y_{i,j,w} \cdot \log(p(a_{i,j,w})) \tag{3.23}$$

Inference of the Multi-Passage Generative Model During training, teacher-enforcing is used to train the encoder-decoder model auto-regressively. During inference time, the answer tokens are generated iteratively by feeding the previous token a_{t-1} to the input of the next token a_t. We apply both greedy-decode and beam-search for the answer decoding. In greedy-decode, the best answer token is always selected with the highest probability at each decoding step. In beam search, a beam of size m is maintained during decoding, and m answer candidates are generated with ranked scores. We also take ensembles of the 6 TRiG models trained on different splits of the top-100 passages, where the best answer is selected by ranking the model answers with average log probability of all the generated tokens of the predicted answer: $a^* = \arg\max_n\{\frac{1}{l}\sum_j^l \ln P(a_{n,j})\}$ and n is the number of ensembles.

3.3.4 Experiments

In this section, we describe the implementation details of our method and report the experimental results.

3.3.4.1 Implementation Details

OK-VQA Dataset We use the OK-VQA dataset in this research work (version v1.1[5], license CC-BY 4.0[6]). It is one of the most challenging visual question answering datasets that is open to all external knowledge usage[MRF19]. The dataset contains 14,055 visual questions over 14,031 images from MSCOCO [LMB14]. The dataset split is 9,009 for training and 5,046 for testing. Each entry contains an image, a question, and 10 ground-truth answers annotated by human annotators.

Dense Passage Retrieval We use BERT-base encoders, E_Q and E_P, in the retrieval module and initialize them with the checkpoints pre-trained on the NQ dataset[KPR19]. Due to the extremely large size of the Wikipedia knowledge base, we choose the HNSW indexing algorithm instead of flat indexing for a much faster speed of queries with acceptable accuracy trade-off. For more details, please refer to the implementation of [JDJ17]. Each query is composed of the question Q_i and the corresponding caption C_i. The number of retrieved passages $k = 100$ for the best possible QA performance.

Generative Multi-Passages QA We use a transformer-based[VSP17] encoder-decoder T5-large[RSR20] model as the backbone. By default, the embedding size of the encoder is 768. The maximum length of the input tokens is restricted to be 300. Padding to the maximum length is applied for multiple questions batch training. Because the training of the generative model with 100 passages is memory-intensive, the batch size is set to be 1 for each GPU. To optimize the QA model, we apply the following techniques: (i) AdamW as the optimizer with a linearly scheduled learning rate starting from $1e - 4$; (ii) Warm-up of 2000 steps as the learning rate scheduler. We train the multi-passages QA model for 20000 optimization steps on an 8xA100 GPU cluster for 12 hours. During inference, both greedy-decode and beam-search are applied to get the best answers. Before evaluation,

[5] https://okvqa.allenai.org/download.html

[6] http://creativecommons.org/licenses/by/4.0/

a normalization step is performed on the generated answers, including lower-casing and removing articles, punctuation, and duplicated white space.

3.3.4.2 Empirical Results on OK-VQA

Performance of Knowledge Retrieval To evaluate the performance of the knowledge passage retrieval module, we consider a question that has a *hit* in its retrieved knowledge passages if at least one of its ground-truth answers appears in the retrieved passages. Then the *hit@k* is defined as the percentage of questions in the entire dataset who get a *hit* in their top *k* retrieved knowledge passages.

	OK-VQA Train	OK-VQA Test
Top-K	hit@k	hit@k
Top-5	42.72%	45.83%
Top-10	54.66%	57.88%
Top-20	68.76%	72.11%
Top-50	72.27%	80.49%
Top-100	83.76%	86.56%

Table 3.12: Hit@k of the dense passage retrieval (k = the number of retrieved knowledge passages).

From Table 3.12, we can observe that the answer retrieval rate *hit@k* increases along with the number of passages *k* from 42.7% to 83.7% as *k* increases from 5 to 100. A larger *k* increases the probability that each question has access to at least one relevant knowledge passage during inference. We experiment with different *k* for the downstream QA model, which will be discussed in subsubsection 3.3.4.3.

Performance of the Generative QA Model The OK-VQA dataset has 10 annotated answers for each question, and we consider both Exact Match and VQA Score as metrics to evaluate the generative QA model. The **Exact Match (EM)** is defined as the percentage of questions whose predicted answer exactly matches any of the 10 annotated answers. EM metric considers every answer as equally ground-truth the same. On the other hand, **VQA**

score defines a voting mechanism so that each annotated answer a_i is assigned a score s_i between 0 and 1[AAL15].

A generated answer $\hat{a}_i$ would get s_i score if it matches the annotated a_i. The VQA metric is an average of the weighted scores over the entire test set. Arguably, the voting mechanism of the VQA score may promote some ground-truth answers over others based on the annotators' consensus subjectively.

Comparison with Supervised-Learning SOTAs The performance of our proposed TRiG framework with state-of-the-art models is reported in Table 3.13. Please note that all the models in comparison are supervised-learning models. Several observations can be made from the table. First, most previous methods utilize the vision-language model as the backbone for question answering and then integrate it with external knowledge. Some represent the knowledge in the form of graph (KRISP[MCP21], ConceptBert[GZA20], RVLESK[STD21]) while others fuse the output of the vision-language model with textual knowledge representation (MAVEx[WLS21]) or implicit knowledge from a language QA mdoel[SAL21]. Second, a concurrent work, VRR[LZB21], transforms the image into caption text and performs span-based question answering on a trimmed knowledge base using Google search engine. Last and most importantly, all of the above methods achieve very similar VQA scores between 38.60 and 39.4, despite usage of diverse sources of knowledge bases such as ConceptNet[WLS21, MCP21, GZA20, STD21], Google Image[WLS21], Google Web Search[LZB21] and Wikipiedia[WLS21] and pretraining on other datasets such as VQA[GZA20, MCP21, STD21, WLS21] and Visual Genome[STD21].

Our proposed TRiG framework significantly outperforms all state-of-the-art supervised-learning methods with at least a 11.1% margin. Our TRiG framework differs from the existing methods as (i) instead of aligning representation of the vision-language QA model with external knowledge in the multimodal space, TRiG transforms the image into text information as accurately as possible and aligns all the information of the image, question,

Model	EM	VQA Score
SOTA Methods		
KRISP[MCP21]		32.31
ConceptBert[GZA20]		33.66
CBM[SAL21]		38.60
KRISP w/ VQA2.0 pretrained		38.70
MAVEx[WLS21]		38.70
RVLESK[STD21]		39.04
Weakly Supervised VRR[LZB21]		39.20
MAVEx w/(Ensemble 5) [WLS21]		39.40
Ours		
TRiG w/ Q+C+DL+O, G	**53.62%**	**49.24**
TRiG w/ Q+C+DL+O, BS	**53.59%**	**49.35**
TRiG w/ Q+C+DL+O, G, **E***	**54.73%**	**50.50**

Table 3.13: Comparison of supervised-learning methods on the OK-VQA dataset. In TRiG Model, **Q**: Question, **C**: Caption, **DL**: Dense Labels, **O**: OCR Text, **G**: Greedy Decode, **BS**: Beam-Search, **E***: Ensembles of the 6 TRiG models.

and knowledge in language space; (ii) the generative QA model in TRiG is not pre-trained on other multimodal datasets, which helps the model to start learning to reason over external knowledge, rather than inducing data bias from other multimodal datasets.

We would like to also highlight the Exact Match (EM) score of our TRiG models, which are higher than the VQA scores. As in Figure 3.10, we observe that sometimes the generative QA model predicts a reasonable answer but is not credited with the highest VQA score or not even any score according to annotators' voting.

Comparison with Prompt-Based SOTA We also compare our method with one very recent prompt-based method on the OK-VQA problem [YGW21]. By taking advantage of the super large-scale language model GPT-3 [BMR20], the proposed prompt-based method (PICa) surpasses all existing supervised methods with sophisticated prompting. As shown in Table 3.14, PICa achieves 43.3 VQA score with 16 prompts randomly selected from the training data. By carefully selecting 16 prompts based on the similarity between testing and training questions, PICa further achieves 46.5. With 5 ensembles of 16 prompts, PICa

Model	#Params	VQA Score
SOTA Prompt Method[YGW21]		
PICa w/16 RP C+T	175B	43.30
PICa w/16 SP C+T	175B	46.50
PICa w/16 SP C+T, 3×E	175B	47.70
PICa w/16 SP C+T, 5×E	175B	48.00
Ours		
TRiG w/ Q+C+DL+O, G	0.77B	**49.24**
TRiG w/ Q+C+DL+O, BS	0.77B	**49.35**
TRiG w/ Q+C+DL+O, G, $\mathbf{E}^*$	0.77B	**50.50**

Table 3.14: Comparison of Proposed TRiG with SOTA Prompt-Based Method on the OK-VQA Dataset. In[YGW21], **RP**: Random Prompt, **SP** Selected Prompt, **C**: Caption, **T**: Image-Tagging, **E**: Prompt Ensemble. In TRiG model, **Q**: Question, **C**: Caption, **DL**: Dense Labels, **O**: OCR Text, **G**: Greedy Decode, **BS**: Beam-Search, $\mathbf{E}^*$: Ensembles of the 6 TRiG models.

reaches 48.0 VQA score.

Our method (TRiG) outperforms PICa with greedy-decode 49.24, beam-search decoding 49.35 and ensemble 50.50. Both PICa and our method share the same idea of unifying the image, the visual question, and knowledge in language space and then performing question answering with language models. The significant performance gain of both methods (9-11.1% over SOTA) highlights the potential of this idea – if the image could be transformed into plain text information faithfully, then one could take advantage of the vast volume of external knowledge in text form and advanced language models pre-trained on rich variations of human natural language to yield better answer prediction.

We would like to also highlight that our method outperforms PICa by a margin of 2.50%, especially considering the among of parameters (175 billion over 0.77 billion of our model) and accessibility of the GPT-3 model. Moreover, we argue that our prediction results are relatively easier to interpret by selecting supporting knowledge passages, whereas in PICa the explanation is generated by GPT-3 in a black-box manner. We use the averaged cross-attention score of the generative model to select supporting facts[IG20a]. For concrete examples of such interpretability, please see the examples in Figure 3.10.

3.3.4.3 Ablation Study

Variant Visual Context Input We investigate the empirical differences among the combination of the visual contexts inputs to the generative QA model, namely image caption (C), object label (L and DL), and OCR (O).

Inputs	VQA Score
Question + K + C	42.54
Question + K + C + L	42.94
Question + K + C + L + O	43.53
Question + K + C + DL + O	**49.35**

Table 3.15: Ablation Study of the Different Variants of Text Input into the Generative QA Model (**K**: Knowledge passages, **C**: Caption, **L** = Bottom-Up Labels[AHB18], **DL** = Dense Labels, **O** = OCR Text).

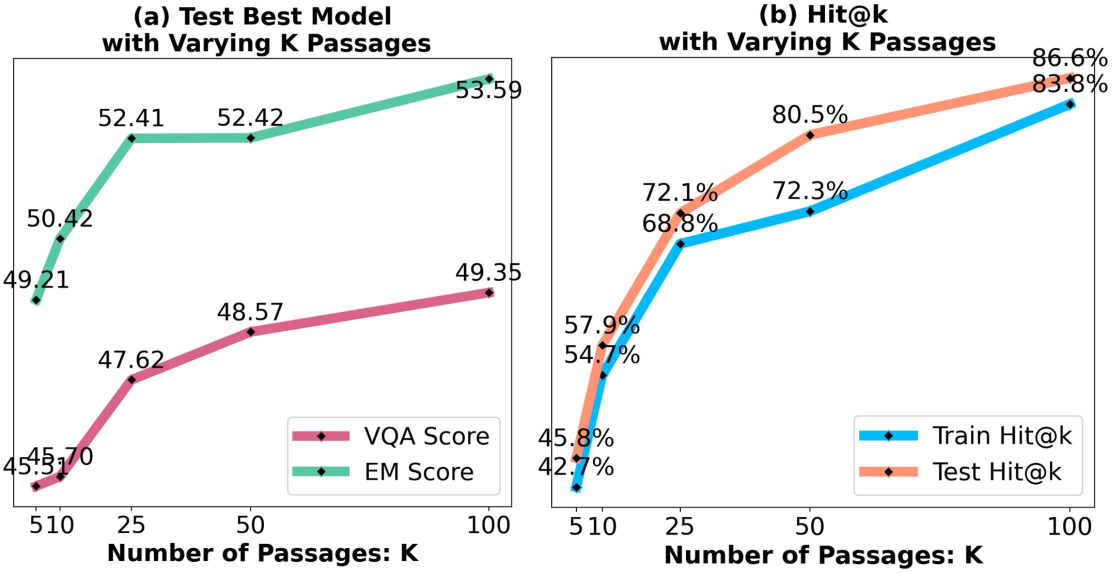
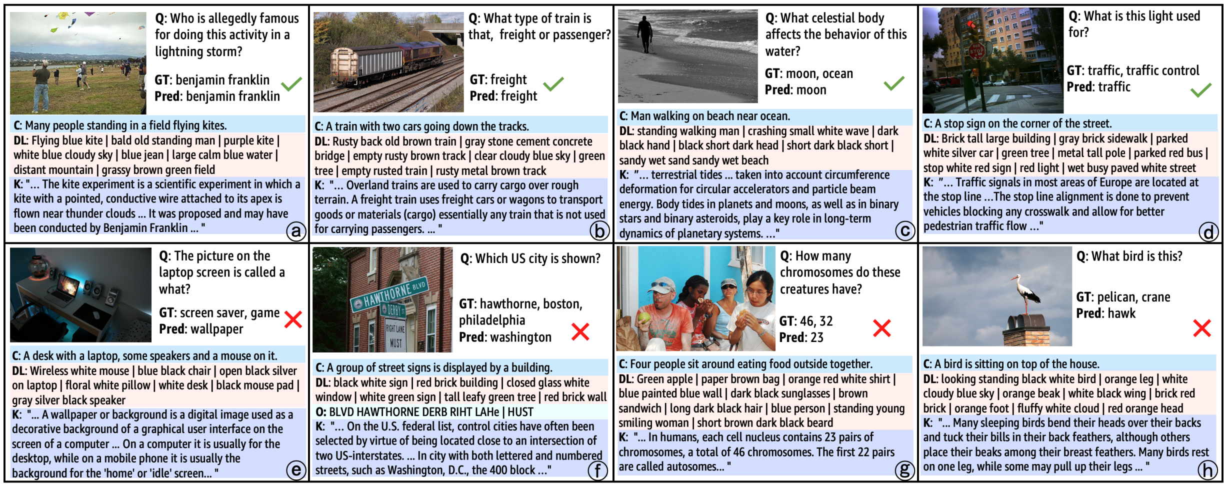


Figure 3.9: Testing the QA model with varying number of passages.

As in Table 3.15, we find that adding caption (C) to the input yields decent performance (42.5), suggesting that caption conveys basic information of the image. Adding sparse object labels and attributes (L) also helps a little (42.9). By adding OCR, the performance is further improved (43.5), which is in accord with previous findings that some questions in OK-VQA require understanding the text in the image through OCR[JKK21]. Interestingly, the largest gain is achieved by replacing sparse object and attribute labels with more semantically rich dense object labels (49.4), which again highlights that the faithfulness of image-to-text transformation is a crucial prerequisite for downstream QA in the language space.

Generative Multi-Passages QA with Varying K passages We also investigate how the generative QA model behaves with a different number of passages k. We apply our best model trained on 100 passages and test it with varying k passages. From Figure 3.9-(a), we can see that the testing performance of this model steadily increases along with the growing number of passages k. However, the improvement becomes marginal after k=25 (47.62 to 49.35), while the coverage *Hit@k* still increases by 15% as Figure 3.9-(b). This also supports our hypothesis that there may be a long-tail effect of the retrieval. Yet it is difficult to quantify as to which passages are essentially relevant to the question-answering.



Q: Who is allegedly famous for doing this activity in a lightning storm?

GT: benjamin franklin
Pred: benjamin franklin ✓

C: Many people standing in a field flying kites.

DL: Flying blue kite | bald old standing man | purple kite | white blue cloudy sky | blue jean | large calm blue water | distant mountain | grassy brown green field

K: "... The kite experiment is a scientific experiment in which a kite with a pointed, conductive wire attached to its apex is flown near thunder clouds ... It was proposed and may have been conducted by Benjamin Franklin ... "

Q: What type of train is that, freight or passenger?

GT: freight
Pred: freight ✓

C: A train with two cars going down the tracks.

DL: Rusty back old brown train | gray stone cement concrete bridge | empty rusty brown track | clear cloudy blue sky | green tree | empty rusted train | rusty metal brown track

K: "... Overland trains are used to carry cargo over rough terrain. A freight train uses freight cars or wagons to transport goods or materials (cargo) essentially any train that is not used for carrying passengers. ... "

Q: What celestial body affects the behavior of this water?

GT: moon, ocean
Pred: moon ✓

C: Man walking on beach near ocean.

DL: standing walking man | crashing small white wave | dark black hand | black short dark head | short dark black short | sandy wet sand sandy wet beach

K: "... terrestrial tides ... taken into account circumference deformation for circular accelerators and particle beam energy. Body tides in planets and moons, as well as in binary stars and binary asteroids, play a key role in long-term dynamics of planetary systems. ..."

Q: What is this light used for?

GT: traffic, traffic control
Pred: traffic ✓

C: A stop sign on the corner of the street.

DL: Brick tall large building | gray brick sidewalk | parked white silver car | green tree | metal tall pole | parked red bus | stop white red sign | red light | wet busy paved white street

K: "... Traffic signals in most areas of Europe are located at the stop line ...The stop line alignment is done to prevent vehicles blocking any crosswalk and allow for better pedestrian traffic flow ..."

Q: The picture on the laptop screen is called a what?

GT: screen saver, game
Pred: wallpaper ✗

C: A desk with a laptop, some speakers and a mouse on it.

DL: Wireless white mouse | blue black chair | open black silver on laptop | floral white pillow | white desk | black mouse pad | gray silver black speaker

K: "... A wallpaper or background is a digital image used as a decorative background of a graphical user interface on the screen of a computer ... On a computer it is usually for the desktop, while on a mobile phone it is usually the background for the 'home' or 'idle' screen... "

Q: Which US city is shown?

GT: hawthorne, boston, philadelphia
Pred: washington ✗

C: A group of street signs is displayed by a building.

DL: black white sign | red brick building | closed glass white window | white green sign | tall leafy green tree | red brick wall

O: BLVD HAWTHORNE DERB RIHT LAHe | HUST

K: "... On the U.S. federal list, control cities have often been selected by virtue of being located close to an intersection of two US-interstates. ... In city with both lettered and numbered streets, such as Washington, D.C., the 400 block ..."

Q: How many chromosomes do these creatures have?

GT: 46, 32
Pred: 23 ✗

C: Four people sit around eating food outside together.

DL: Green apple | paper brown bag | orange red white shirt | blue painted blue wall | dark black sunglasses | brown sandwich | long dark black hair | blue person | standing young smiling woman | short brown dark black beard

K: "... In humans, each cell nucleus contains 23 pairs of chromosomes, a total of 46 chromosomes. The first 22 pairs are called autosomes... "

Q: What bird is this?

GT: pelican, crane
Pred: hawk ✗

C: A bird is sitting on top of the house.

DL: looking standing black white bird | orange leg | white cloudy blue sky | orange beak | white black wing | brick red brick | orange foot | fluffy white cloud | red orange head

K: "... Many sleeping birds bend their heads over their backs and tuck their bills in their back feathers, although others place their beaks among their breast feathers. Many birds rest on one leg, while some may pull up their legs ... "

Figure 3.10: Examples of our TRiG model prediction together with the supporting passage. **Top**: four examples where TRiG model makes correct predictions. **Bottom**: four examples where TRiG model makes incorrect predictions. In each example: **Q**: question, **GT**: ground-truth answers, **Pred**: predicted answer, **C**: image caption, **DL**: dense labels, **O**: OCR text, **K**: top-1 supporting knowledge passage.

3.3.4.4 Discussion

Error Analysis To investigate the behavior of our TRiG model, we conduct error analysis with our best model using greedy-decoded predictions. The quantitative results are illustrated in Figure 3.11. We observe that answers with numerical values are harder to predict, where the model could get into a blunt generation (Figure 3.11-(a)). Furthermore, as the length of the answer increases, it is harder for the generative model to predict every token in the phrase correctly (Figure 3.11-(b)).

We also manually reviewed 50 examples where TRiG makes wrong predictions. Among these random examples, 50% of the errors are due to the information loss during image-to-text transformation, such as in Figure 3.10-(h), where the caption and dense labels failed to characterize the special features of the bird. We also found that 24% of the error are due to the failure in retrieving highly-relevant passages. The high Hit@k value doesn't guarantee the passages are indeed relevant to the question. Note that some examples failed due to multiple reasons including QA error (22%) or subjective human annotations (30%) as in Figure 3.10(g).

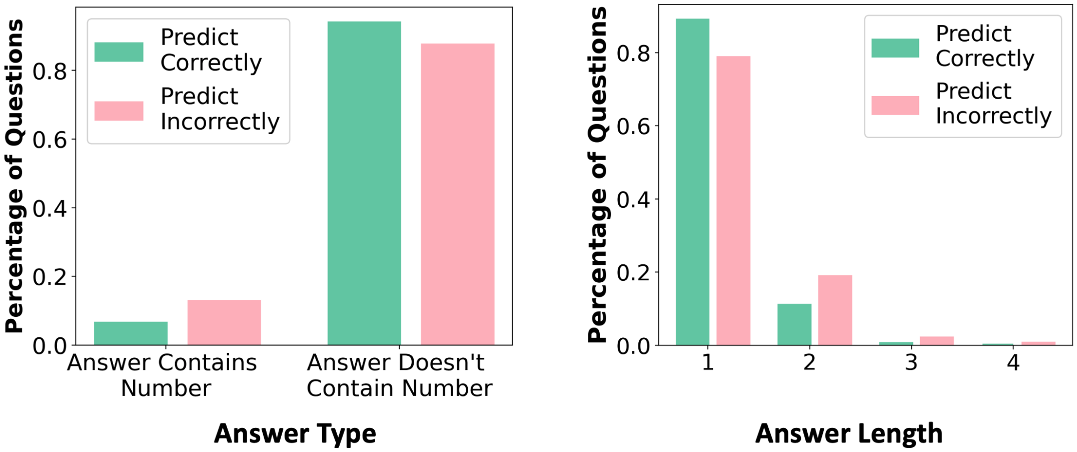


Figure 3.11: Performance of Generative QA model by different answer types. **Left**: whether numerical answers are harder to predict. **Right**: whether longer answers are harder to predict.

Interpretability To interpret the visual question-answering models, previous works attempt to supervise the VQA models with visual grounding annotations[ZNS19, DAZ17, DAZ17] or neural symbolic network[VDL19, YWG18, HAR17]. When it comes to knowledge-based VQA, it is all the more challenging to interpret the model in multimodal space because the knowledge has been transformed into a fused representation and loses its meaning.

Our TRiG framework alleviates this problem by providing transparent explanations in the language space. In the top row in Figure 3.10, the image-to-text transformations provide sufficient information for both the knowledge retrieval and QA model. Meanwhile, when Figure 3.10(e, f, g, h) make wrong predictions, the QA model is still predicting the answer according to the visual context and retrieved passages.

OK-VQA Evaluation Metrics Some researchers[LST21] also argue that the VQA score metric is subjective. In one OK-VQA example, a model will achieve 1.0 VQA score for the answer ***wetsuit*** but only 0.66 score for the answer ***wet suit***. In daily language, the usage of any of the semantically-similar answers is subtle and sometimes random. We also look at the top-3 answers of our TRiG model using beam search, and the model achieves significantly higher performance, *i.e.* **67.4** VQA score and **71.8%** EM. We call for better VQA metrics that probably compare two sets of answers instead of comparing only the top one answer or other alternatives such as AAS that automatically expands the ground-truth answer set for better matching[LST21].

3.3.5 Conclusion

In this work, we approach the OK-VQA task from a new perspective, where all the visual information is aligned into the language space to take advantage of the comprehensiveness in textual knowledge bases. Moreover, we propose a robust Transform-Retrieve-Generate (TRiG) framework that outperforms state-of-the-art supervised methods by 11.1%. One can plug-and-play with different image-to-text methods and textual knowledge bases into

TRiG for potential further improvement. Our work has limitations that the dense passage retrieval is not optimized for the OK-VQA task, due to the unavailability of ground-truth supporting facts. We consider this as one of our future work, as well as improving the quality of image-to-text transformation.

3.3.6 Supplementary Materials

3.3.6.1 Additional Details for Methodology

Hyper-parameters To better illustrate the implementation of the generative multi-passages QA model, we introduce some key hyper-parameters in Table 3.16.

Hyper-Parameter	Value
Max Input Length	300
Max Decoding Length	20
Early Stopping	True
Pad to Max Length	True
Max Number of Beams	3
Learning Rate	0.0001
LR Scheduler	Linear
Total Optimization Step	20000

Table 3.16: Hyper-parameters of the generative multi-passages QA model, not including hyper-parameters for T5 backbone.

Input Format Different from the default input format to the pre-train a T5 model, we use a alternative formatting for the input sequences. We concatenate the question, the visual context and one retrieved Wikipedia knowledge passage as the input sequence, without any special token such as "*[SEP]*" between them. The question has a prefix "*question:* " before it. The visual context is the concatenation of image caption, dense labels and OCR text. The knowledge passage consists of a Wikipedia title and a Wikipedia paragraph. The two are concatenated by putting a prefix "*title:* " and a prefix "*context:* " before them respectively.

Vocabulary We also want to highlight the effect of the different sizes of QA model vocabulary. As in Table 3.17, we notice a trend that models with larger vocabulary sizes achieve higher performance. In particular, models using the default vocabulary (PICa and TRiG) perform better on OK-VQA dataset.

Method	Size	VQA Score
KRISP w/o VQA2 pre-train	2,250	32.31
Weakly Supervised VRR (C)	11,060	36.78
RVLESK	14,456	39.04
PICa (5 Ensembles)	50,257	48.00
Ours (6 Ensembles)	32,128	**50.50**

Table 3.17: The vocabulary size and performances of different SOTA methods on OKVQA. (C) represents classification. Some numbers may not be public accessible and we only report the numbers directly from the authors.

3.3.6.2 Additional Details for Ablation Study

Answer Accuracy in Beam-Search In this work, we argue that the ground-truth answers of an OK-VQA question might be a semantically-similar cluster, such as (*swimsuit*, *bath suit*, *bikini*). This may also hold true for the question answering models, in terms of both classification models (top-k class prediction) and generative models (top-k beam prediction).

	Exact Match	VQA
Top-1	53.59%	49.35
Top-2	65.99%	61.61
Top-3	**71.78%**	**67.48**

Table 3.18: Ablation on Different k in Beam-Search Decoding.

We report the performance of our generative question answering model using top-1/2/3 beam-search decoding. As shown in Table 3.18, we can find that the both the Exact Match (EM) and VQA score increase as the k of beam-search increase. This suggests that while the top-one answers only achieve 49.35 VQA score, their semantically-similar candidates could reach as high as 67.48 VQA score. Therefore, we call out for new metrics that compare two sets of answers instead of top-one answer versus many ground-truth answers.

Backbone Model Size To further illustrate the effectiveness and the efficiency of our model, we also compare the performance with various backbone model size. In Table 3.19, we show the VQA scores of different model backbones and highlights the approximate size of them.

Method and Backbone	Size/# Params	VQA Score
MAVEx (VilBert)	1.02GB	39.04
VRR (RoBerta-Large)	1.33GB	39.20
PICa (GPT-3)	175B params	48.00
Ours (w/ T5-Base)	0.85GB	46.50
Ours (w/ T5-Large)	2.75GB	**50.50**

Table 3.19: Performances and size of the backbone models in different methods. Since GPT-3 is not fully accessible, we only indicate the number of parameters of it which is 175 billion.

3.3.6.3 Additional Details for Error Analysis

To further understand the behavior of our TRiG framework, we conducted several error analysis.

Question Keywords / Types First, we investigate whether the model is likely to predict correctly over some question keywords than others. As in Figure 3.12-top, we can observe that majority of the questions contain the keyword "*what*", where our model is more likely to make correct predictions. On the other hand, for questions containing keywords such as "*how*" and "*why*", our model is more likely to make mistakes. We hypothesize that the "*how*" and "*why*" questions usually entail longer answers, which is harder for the generative model to predict. For example, for the question *why is this sign here*? (a sign for animal protection), the ground-truth answers are (*protect animal*, *safety*, *don't feed animal*, *direct*).

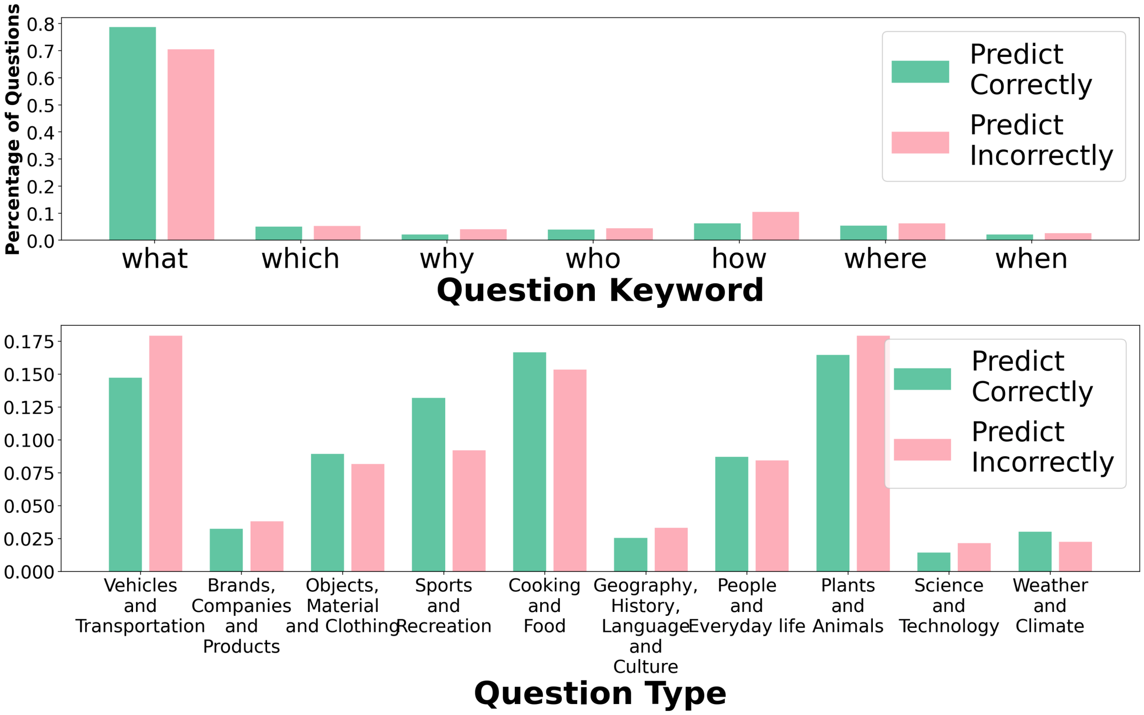
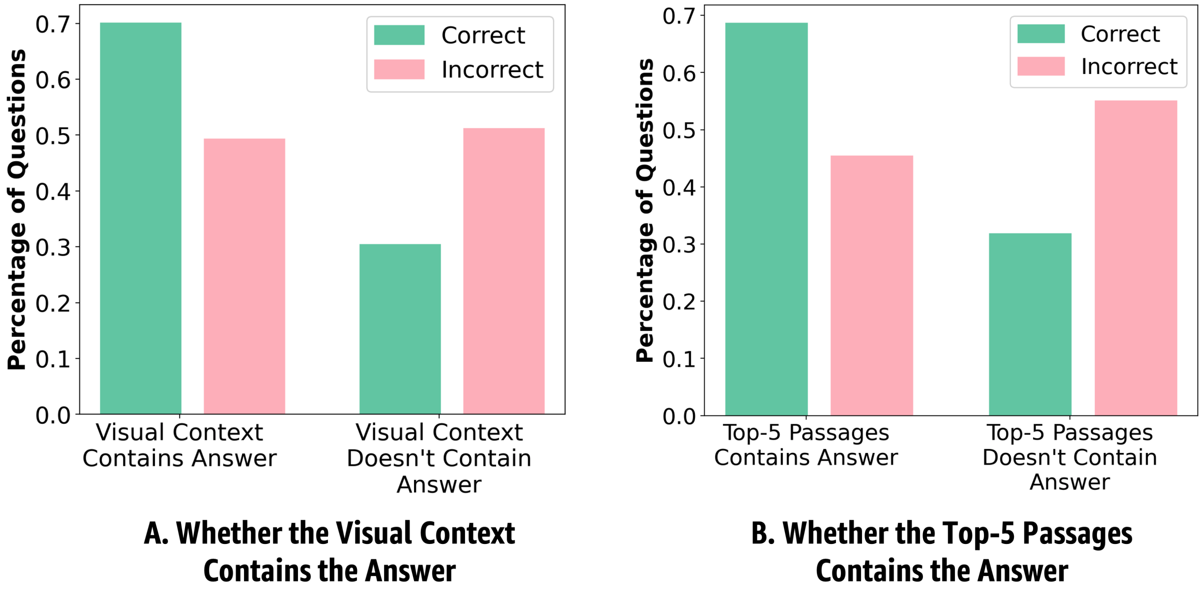


Figure 3.12: Distribution of correct/incorrect predictions. Top: Distributions of predictions over different question keywords. Bottom: Distribution of predictions over different question types.

Second, we report prediction distribution over 10 question types that are available from the OK-VQA dataset. As in Figure 3.12-bottom, we can see that our model is more likely to predict correctly on category of sports and recreation. On the contrary, our model makes more mistakes in Vehicles and Transportation and Plants and Animals.

The Impact of Visual Context and Knowledge Passages First, we would like to further investigate the effectiveness of the image-to-text transformation module, since it is the first stage in our TRiG framework. Shown in Figure 3.13-A, we find that if the visual contexts contain the ground-truth answers, the generative question answering model is more likely to generate a correct answer. In contrast, the model makes more mistakes if the visual contexts do not contain the ground-truth answers.

Second, we also investigate how the retrieved passages impact the generative question answering model. As is illustrated in Figure 3.13-B, we find that if the top-5 passages that contain the ground-truth answers, our generative question answering model is much more likely to predict correct answers. On the opposite side, if top-5 passages do not contain the ground-truth answers, it is more likely for the QA model to make a wrong prediction.



A. Whether the Visual Context Contains the Answer

B. Whether the Top-5 Passages Contains the Answer

Figure 3.13: Error Category Break-Down.

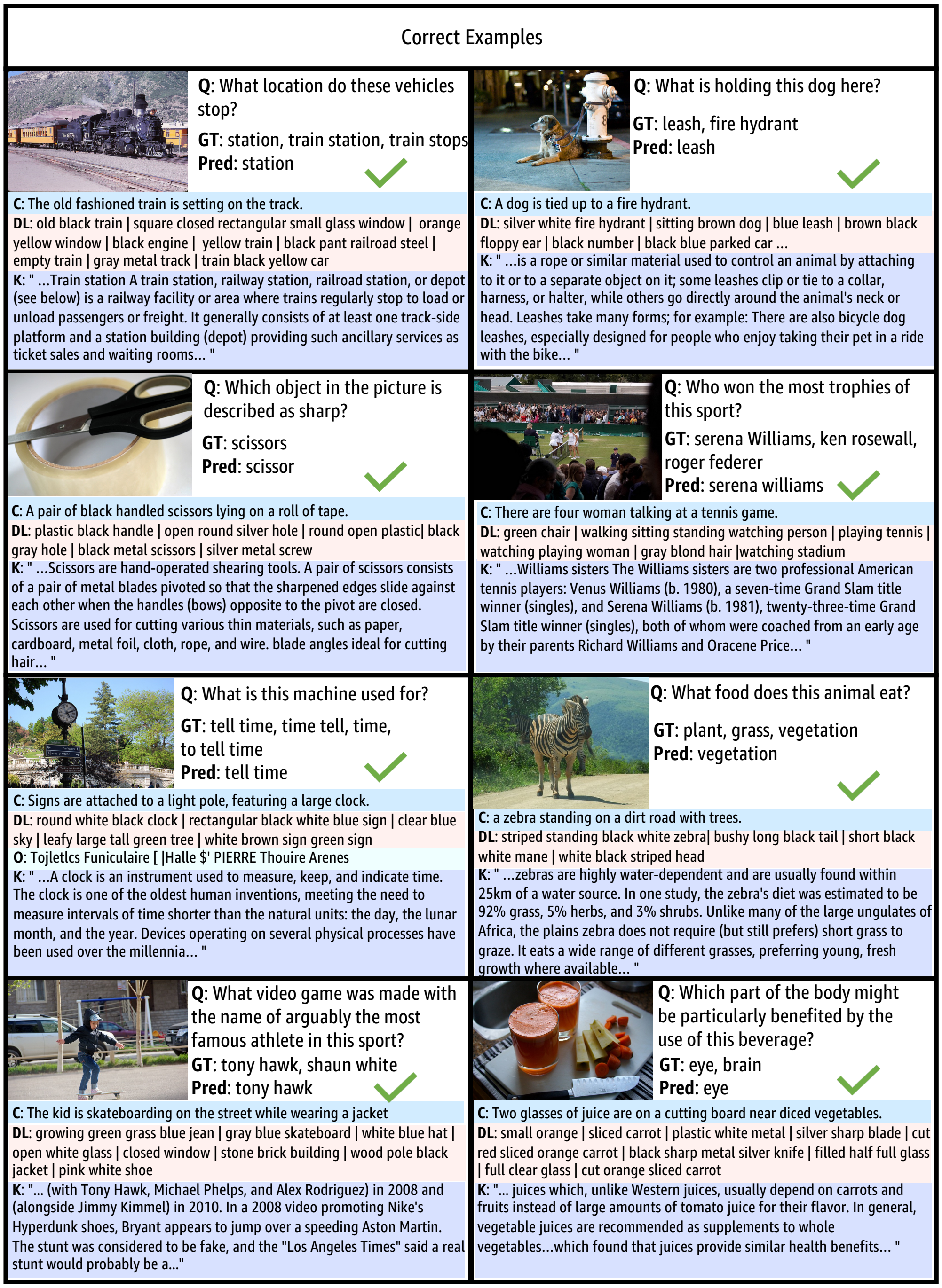
Manual Error Review We also conducted manual eye-balling on 50 random examples where the model has made wrong predictions. We look into each example with all the available information (question, caption, dense labels, OCR text, knowledge passages, ground-truth answers) and attribute each example to one or more error categories. A brief statistics is shown in Table 3.20. Please note that the percentages of error types are not mutually exclusive because some wrong cases may fall in multiple categories.

We can observe that the first contributing factor to the errors is in image-to-text transformation (50%). The second category is the answer annotation ambiguity (30%), where the predicted answers are reasonable according to human judgement, but do not match any ground-truth answers. There are also failures related to dense passage retrieval (24%) and

Category	Percentage
Image-to-Text	50%
Annotation	30%
Dense Passage Retrieval	24%
Generative QA	22%

Table 3.20: Ablation on Different k in Beam-Search Decoding.

generative QA model (22%). For more details of each error category, please see the examples in page 5-6.



Correct Examples

Q: What location do these vehicles stop?

GT: station, train station, train stops

Pred: station ✓

C: The old fashioned train is setting on the track.

DL: old black train | square closed rectangular small glass window | orange yellow window | black engine | yellow train | black pant railroad steel | empty train | gray metal track | train black yellow car

K: " ...Train station A train station, railway station, railroad station, or depot (see below) is a railway facility or area where trains regularly stop to load or unload passengers or freight. It generally consists of at least one track-side platform and a station building (depot) providing such ancillary services as ticket sales and waiting rooms... "

Q: What is holding this dog here?

GT: leash, fire hydrant

Pred: leash ✓

C: A dog is tied up to a fire hydrant.

DL: silver white fire hydrant | sitting brown dog | blue leash | brown black floppy ear | black number | black blue parked car ...

K: " ...is a rope or similar material used to control an animal by attaching to it or to a separate object on it; some leashes clip or tie to a collar, harness, or halter, while others go directly around the animal's neck or head. Leashes take many forms; for example: There are also bicycle dog leashes, especially designed for people who enjoy taking their pet in a ride with the bike... "

Q: Which object in the picture is described as sharp?

GT: scissors

Pred: scissor ✓

C: A pair of black handled scissors lying on a roll of tape.

DL: plastic black handle | open round silver hole | round open plastic| black gray hole | black metal scissors | silver metal screw

K: " ...Scissors are hand-operated shearing tools. A pair of scissors consists of a pair of metal blades pivoted so that the sharpened edges slide against each other when the handles (bows) opposite to the pivot are closed. Scissors are used for cutting various thin materials, such as paper, cardboard, metal foil, cloth, rope, and wire. blade angles ideal for cutting hair... "

Q: Who won the most trophies of this sport?

GT: serena Williams, ken rosewall, roger federer

Pred: serena williams ✓

C: There are four woman talking at a tennis game.

DL: green chair | walking sitting standing watching person | playing tennis | watching playing woman | gray blond hair |watching stadium

K: " ...Williams sisters The Williams sisters are two professional American tennis players: Venus Williams (b. 1980), a seven-time Grand Slam title winner (singles), and Serena Williams (b. 1981), twenty-three-time Grand Slam title winner (singles), both of whom were coached from an early age by their parents Richard Williams and Oracene Price... "

Q: What is this machine used for?

GT: tell time, time tell, time, to tell time

Pred: tell time ✓

C: Signs are attached to a light pole, featuring a large clock.

DL: round white black clock | rectangular black white blue sign | clear blue sky | leafy large tall green tree | white brown sign green sign

O: Tojletlcs Funiculaire [|Halle $' PIERRE Thouire Arenes

K: " ...A clock is an instrument used to measure, keep, and indicate time. The clock is one of the oldest human inventions, meeting the need to measure intervals of time shorter than the natural units: the day, the lunar month, and the year. Devices operating on several physical processes have been used over the millennia... "

Q: What food does this animal eat?

GT: plant, grass, vegetation

Pred: vegetation ✓

C: a zebra standing on a dirt road with trees.

DL: striped standing black white zebra| bushy long black tail | short black white mane | white black striped head

K: " ...zebras are highly water-dependent and are usually found within 25km of a water source. In one study, the zebra's diet was estimated to be 92% grass, 5% herbs, and 3% shrubs. Unlike many of the large ungulates of Africa, the plains zebra does not require (but still prefers) short grass to graze. It eats a wide range of different grasses, preferring young, fresh growth where available... "

Q: What video game was made with the name of arguably the most famous athlete in this sport?

GT: tony hawk, shaun white

Pred: tony hawk ✓

C: The kid is skateboarding on the street while wearing a jacket

DL: growing green grass blue jean | gray blue skateboard | white blue hat | open white glass | closed window | stone brick building | wood pole black jacket | pink white shoe

K: "... (with Tony Hawk, Michael Phelps, and Alex Rodriguez) in 2008 and (alongside Jimmy Kimmel) in 2010. In a 2008 video promoting Nike's Hyperdunk shoes, Bryant appears to jump over a speeding Aston Martin. The stunt was considered to be fake, and the "Los Angeles Times" said a real stunt would probably be a..."

Q: Which part of the body might be particularly benefited by the use of this beverage?

GT: eye, brain

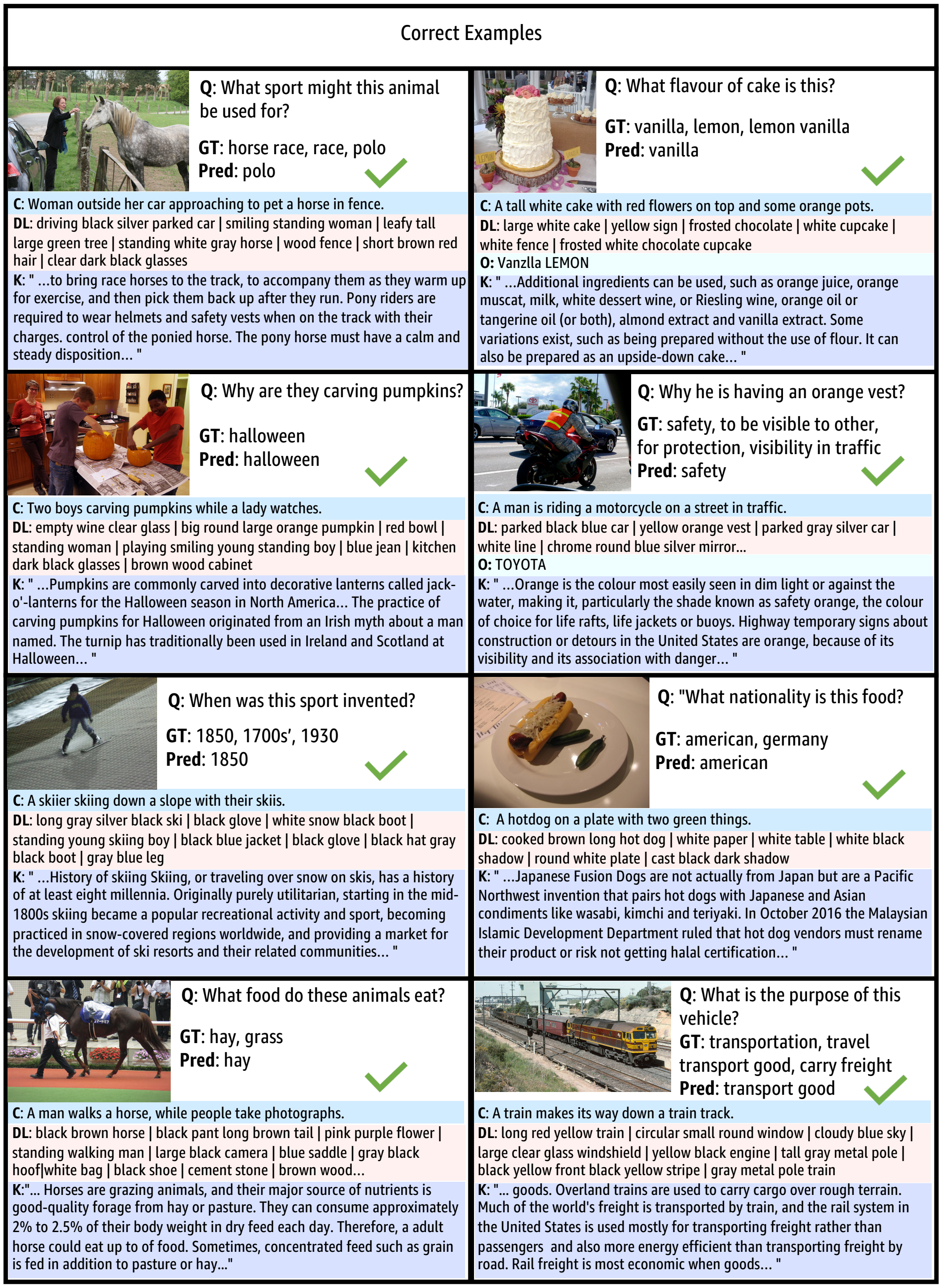
Pred: eye ✓

C: Two glasses of juice are on a cutting board near diced vegetables.

DL: small orange | sliced carrot | plastic white metal | silver sharp blade | cut red sliced orange carrot | black sharp metal silver knife | filled half full glass | full clear glass | cut orange sliced carrot

K: "... juices which, unlike Western juices, usually depend on carrots and fruits instead of large amounts of tomato juice for their flavor. In general, vegetable juices are recommended as supplements to whole vegetables...which found that juices provide similar health benefits... "

Figure 3.14



Correct Examples

Q: What sport might this animal be used for?

GT: horse race, race, polo
Pred: polo ✓

C: Woman outside her car approaching to pet a horse in fence.

DL: driving black silver parked car | smiling standing woman | leafy tall large green tree | standing white gray horse | wood fence | short brown red hair | clear dark black glasses

K: " ...to bring race horses to the track, to accompany them as they warm up for exercise, and then pick them back up after they run. Pony riders are required to wear helmets and safety vests when on the track with their charges. control of the ponied horse. The pony horse must have a calm and steady disposition... "

Q: What flavour of cake is this?

GT: vanilla, lemon, lemon vanilla
Pred: vanilla ✓

C: A tall white cake with red flowers on top and some orange pots.

DL: large white cake | yellow sign | frosted chocolate | white cupcake | white fence | frosted white chocolate cupcake

O: Vanzlla LEMON

K: " ...Additional ingredients can be used, such as orange juice, orange muscat, milk, white dessert wine, or Riesling wine, orange oil or tangerine oil (or both), almond extract and vanilla extract. Some variations exist, such as being prepared without the use of flour. It can also be prepared as an upside-down cake... "

Q: Why are they carving pumpkins?

GT: halloween
Pred: halloween ✓

C: Two boys carving pumpkins while a lady watches.

DL: empty wine clear glass | big round large orange pumpkin | red bowl | standing woman | playing smiling young standing boy | blue jean | kitchen dark black glasses | brown wood cabinet

K: " ...Pumpkins are commonly carved into decorative lanterns called jack-o'-lanterns for the Halloween season in North America... The practice of carving pumpkins for Halloween originated from an Irish myth about a man named. The turnip has traditionally been used in Ireland and Scotland at Halloween... "

Q: Why he is having an orange vest?

GT: safety, to be visible to other, for protection, visibility in traffic
Pred: safety ✓

C: A man is riding a motorcycle on a street in traffic.

DL: parked black blue car | yellow orange vest | parked gray silver car | white line | chrome round blue silver mirror...

O: TOYOTA

K: " ...Orange is the colour most easily seen in dim light or against the water, making it, particularly the shade known as safety orange, the colour of choice for life rafts, life jackets or buoys. Highway temporary signs about construction or detours in the United States are orange, because of its visibility and its association with danger... "

Q: When was this sport invented?

GT: 1850, 1700s', 1930
Pred: 1850 ✓

C: A skiier skiing down a slope with their skiis.

DL: long gray silver black ski | black glove | white snow black boot | standing young skiing boy | black blue jacket | black glove | black hat gray black boot | gray blue leg

K: " ...History of skiing Skiing, or traveling over snow on skis, has a history of at least eight millennia. Originally purely utilitarian, starting in the mid-1800s skiing became a popular recreational activity and sport, becoming practiced in snow-covered regions worldwide, and providing a market for the development of ski resorts and their related communities... "

Q: "What nationality is this food?

GT: american, germany
Pred: american ✓

C: A hotdog on a plate with two green things.

DL: cooked brown long hot dog | white paper | white table | white black shadow | round white plate | cast black dark shadow

K: " ...Japanese Fusion Dogs are not actually from Japan but are a Pacific Northwest invention that pairs hot dogs with Japanese and Asian condiments like wasabi, kimchi and teriyaki. In October 2016 the Malaysian Islamic Development Department ruled that hot dog vendors must rename their product or risk not getting halal certification... "

Q: What food do these animals eat?

GT: hay, grass
Pred: hay ✓

C: A man walks a horse, while people take photographs.

DL: black brown horse | black pant long brown tail | pink purple flower | standing walking man | large black camera | blue saddle | gray black hoof|white bag | black shoe | cement stone | brown wood...

K:"... Horses are grazing animals, and their major source of nutrients is good-quality forage from hay or pasture. They can consume approximately 2% to 2.5% of their body weight in dry feed each day. Therefore, a adult horse could eat up to of food. Sometimes, concentrated feed such as grain is fed in addition to pasture or hay..."

Q: What is the purpose of this vehicle?

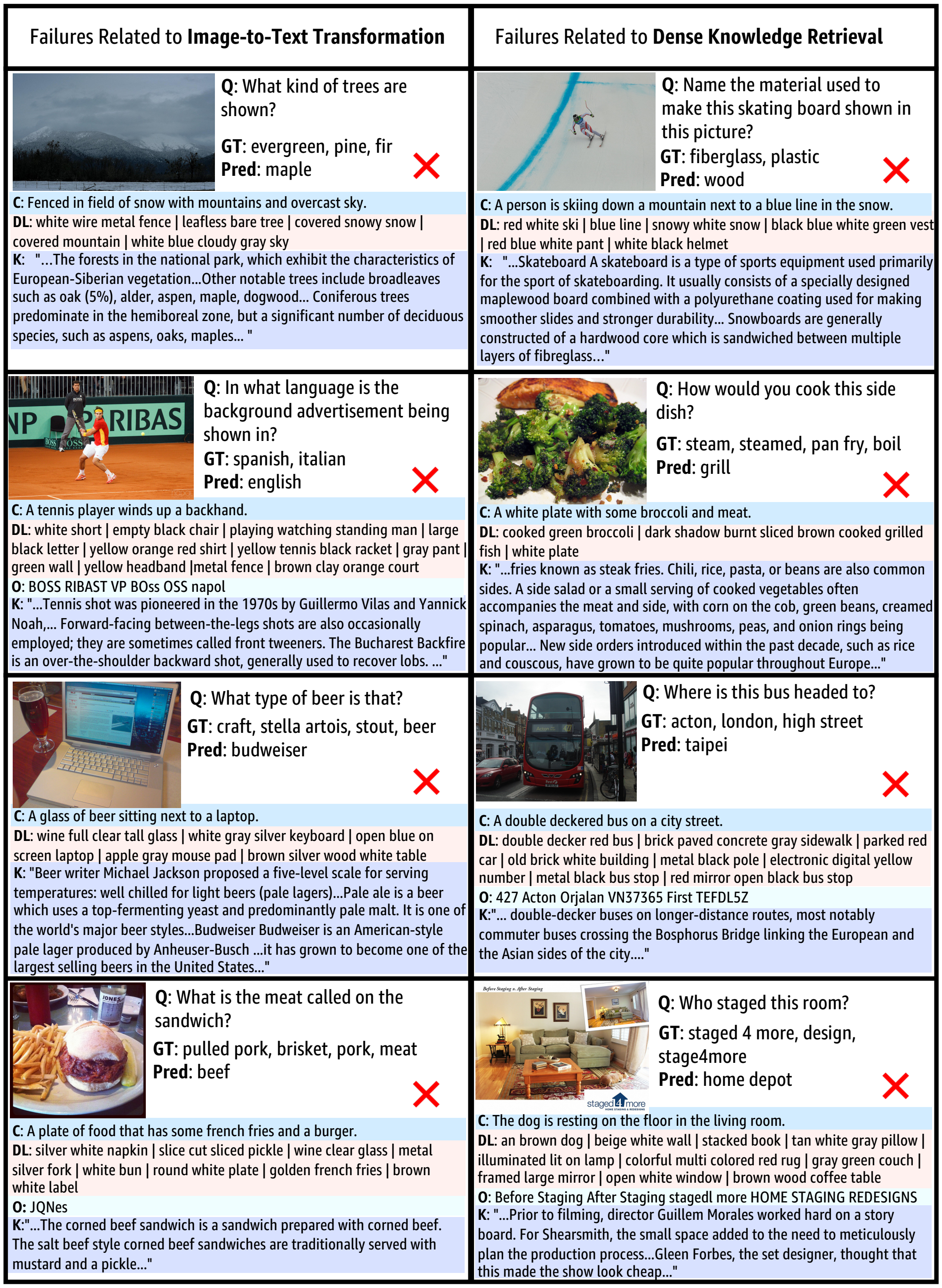
GT: transportation, travel transport good, carry freight
Pred: transport good ✓

C: A train makes its way down a train track.

DL: long red yellow train | circular small round window | cloudy blue sky | large clear glass windshield | yellow black engine | tall gray metal pole | black yellow front black yellow stripe | gray metal pole train

K: "... goods. Overland trains are used to carry cargo over rough terrain. Much of the world's freight is transported by train, and the rail system in the United States is used mostly for transporting freight rather than passengers and also more energy efficient than transporting freight by road. Rail freight is most economic when goods... "

Figure 3.15



Failures Related to **Image-to-Text Transformation**

Q: What kind of trees are shown?

GT: evergreen, pine, fir
Pred: maple ✗

C: Fenced in field of snow with mountains and overcast sky.
DL: white wire metal fence | leafless bare tree | covered snowy snow | covered mountain | white blue cloudy gray sky
K: "...The forests in the national park, which exhibit the characteristics of European-Siberian vegetation...Other notable trees include broadleaves such as oak (5%), alder, aspen, maple, dogwood... Coniferous trees predominate in the hemiboreal zone, but a significant number of deciduous species, such as aspens, oaks, maples... "

Q: In what language is the background advertisement being shown in?
GT: spanish, italian
Pred: english ✗

C: A tennis player winds up a backhand.
DL: white short | empty black chair | playing watching standing man | large black letter | yellow orange red shirt | yellow tennis black racket | gray pant | green wall | yellow headband |metal fence | brown clay orange court
O: BOSS RIBAST VP BOss OSS napol
K: "...Tennis shot was pioneered in the 1970s by Guillermo Vilas and Yannick Noah,... Forward-facing between-the-legs shots are also occasionally employed; they are sometimes called front tweeners. The Bucharest Backfire is an over-the-shoulder backward shot, generally used to recover lobs. ..."

Q: What type of beer is that?
GT: craft, stella artois, stout, beer
Pred: budweiser ✗

C: A glass of beer sitting next to a laptop.
DL: wine full clear tall glass | white gray silver keyboard | open blue on screen laptop | apple gray mouse pad | brown silver wood white table
K: "Beer writer Michael Jackson proposed a five-level scale for serving temperatures: well chilled for light beers (pale lagers)...Pale ale is a beer which uses a top-fermenting yeast and predominantly pale malt. It is one of the world's major beer styles...Budweiser Budweiser is an American-style pale lager produced by Anheuser-Busch ...it has grown to become one of the largest selling beers in the United States..."

Q: What is the meat called on the sandwich?
GT: pulled pork, brisket, pork, meat
Pred: beef ✗

C: A plate of food that has some french fries and a burger.
DL: silver white napkin | slice cut sliced pickle | wine clear glass | metal silver fork | white bun | round white plate | golden french fries | brown white label
O: JQNes
K:"...The corned beef sandwich is a sandwich prepared with corned beef. The salt beef style corned beef sandwiches are traditionally served with mustard and a pickle..."

Failures Related to **Dense Knowledge Retrieval**

Q: Name the material used to make this skating board shown in this picture?
GT: fiberglass, plastic
Pred: wood ✗

C: A person is skiing down a mountain next to a blue line in the snow.
DL: red white ski | blue line | snowy white snow | black blue white green vest | red blue white pant | white black helmet
K: "...Skateboard A skateboard is a type of sports equipment used primarily for the sport of skateboarding. It usually consists of a specially designed maplewood board combined with a polyurethane coating used for making smoother slides and stronger durability... Snowboards are generally constructed of a hardwood core which is sandwiched between multiple layers of fibreglass..."

Q: How would you cook this side dish?

GT: steam, steamed, pan fry, boil
Pred: grill ✗

C: A white plate with some broccoli and meat.
DL: cooked green broccoli | dark shadow burnt sliced brown cooked grilled fish | white plate
K: "...fries known as steak fries. Chili, rice, pasta, or beans are also common sides. A side salad or a small serving of cooked vegetables often accompanies the meat and side, with corn on the cob, green beans, creamed spinach, asparagus, tomatoes, mushrooms, peas, and onion rings being popular... New side orders introduced within the past decade, such as rice and couscous, have grown to be quite popular throughout Europe..."

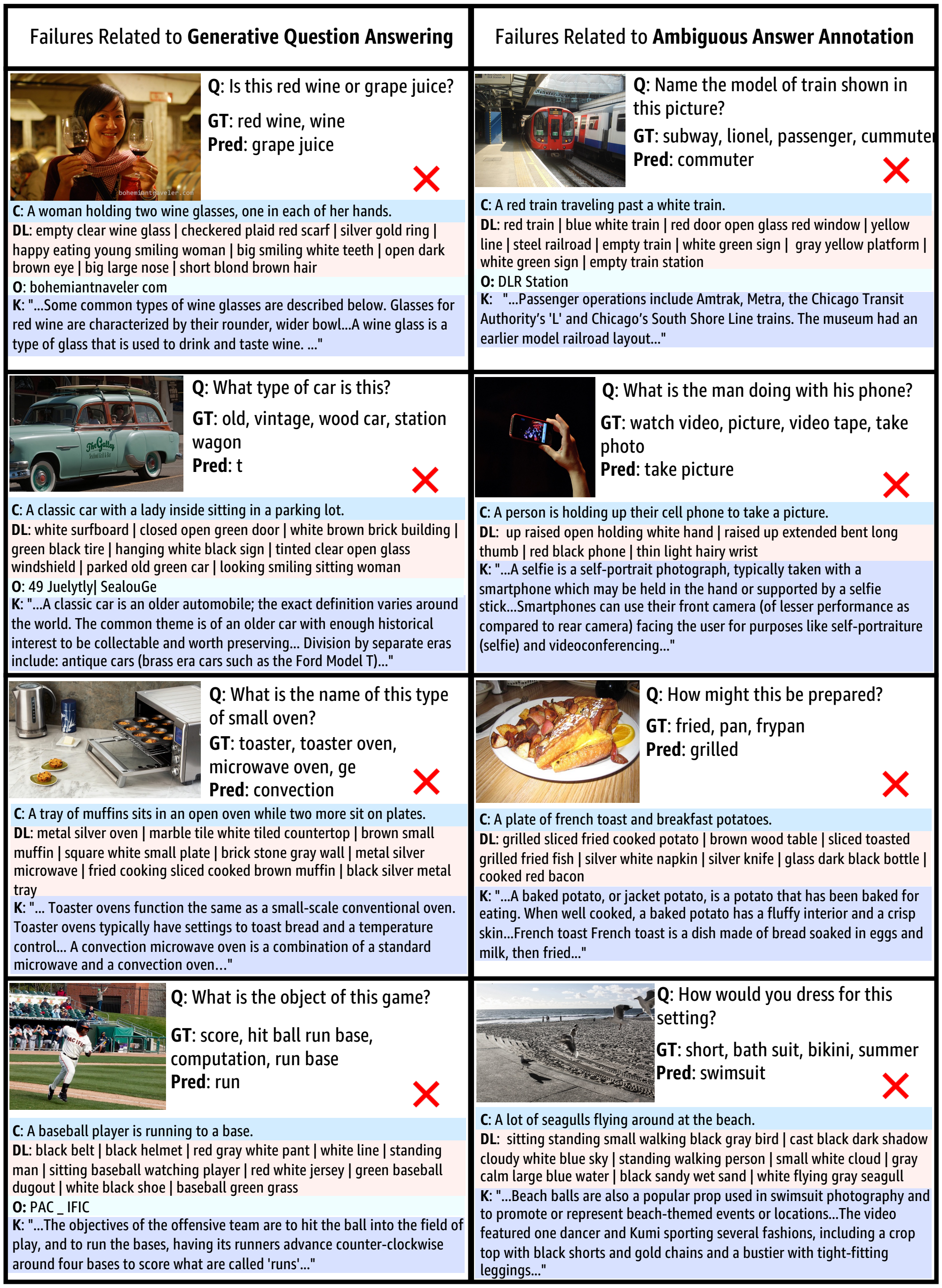
Q: Where is this bus headed to?
GT: acton, london, high street
Pred: taipei ✗

C: A double deckered bus on a city street.
DL: double decker red bus | brick paved concrete gray sidewalk | parked red car | old brick white building | metal black pole | electronic digital yellow number | metal black bus stop | red mirror open black bus stop
O: 427 Acton Orjalan VN37365 First TEFDL5Z
K:"... double-decker buses on longer-distance routes, most notably commuter buses crossing the Bosphorus Bridge linking the European and the Asian sides of the city...."

Q: Who staged this room?
GT: staged 4 more, design, stage4more
Pred: home depot ✗

C: The dog is resting on the floor in the living room.
DL: an brown dog | beige white wall | stacked book | tan white gray pillow | illuminated lit on lamp | colorful multi colored red rug | gray green couch | framed large mirror | open white window | brown wood coffee table
O: Before Staging After Staging stagedl more HOME STAGING REDESIGNS
K: "...Prior to filming, director Guillem Morales worked hard on a story board. For Shearsmith, the small space added to the need to meticulously plan the production process...Gleen Forbes, the set designer, thought that this made the show look cheap..."

Figure 3.16



Failures Related to **Generative Question Answering**

Q: Is this red wine or grape juice?

GT: red wine, wine
Pred: grape juice

C: A woman holding two wine glasses, one in each of her hands.
DL: empty clear wine glass | checkered plaid red scarf | silver gold ring | happy eating young smiling woman | big smiling white teeth | open dark brown eye | big large nose | short blond brown hair
O: bohemiantnaveler com
K: "...Some common types of wine glasses are described below. Glasses for red wine are characterized by their rounder, wider bowl...A wine glass is a type of glass that is used to drink and taste wine. ..."

Q: What type of car is this?

GT: old, vintage, wood car, station wagon
Pred: t

C: A classic car with a lady inside sitting in a parking lot.
DL: white surfboard | closed open green door | white brown brick building | green black tire | hanging white black sign | tinted clear open glass windshield | parked old green car | looking smiling sitting woman
O: 49 Juelytly| SealouGe
K: "...A classic car is an older automobile; the exact definition varies around the world. The common theme is of an older car with enough historical interest to be collectable and worth preserving... Division by separate eras include: antique cars (brass era cars such as the Ford Model T)..."

Q: What is the name of this type of small oven?
GT: toaster, toaster oven, microwave oven, ge
Pred: convection

C: A tray of muffins sits in an open oven while two more sit on plates.
DL: metal silver oven | marble tile white tiled countertop | brown small muffin | square white small plate | brick stone gray wall | metal silver microwave | fried cooking sliced cooked brown muffin | black silver metal tray
K: "... Toaster ovens function the same as a small-scale conventional oven. Toaster ovens typically have settings to toast bread and a temperature control... A convection microwave oven is a combination of a standard microwave and a convection oven..."

Q: What is the object of this game?

GT: score, hit ball run base, computation, run base
Pred: run

C: A baseball player is running to a base.
DL: black belt | black helmet | red gray white pant | white line | standing man | sitting baseball watching player | red white jersey | green baseball dugout | white black shoe | baseball green grass
O: PAC _ IFIC
K: "...The objectives of the offensive team are to hit the ball into the field of play, and to run the bases, having its runners advance counter-clockwise around four bases to score what are called 'runs'..."

Failures Related to **Ambiguous Answer Annotation**

Q: Name the model of train shown in this picture?
GT: subway, lionel, passenger, cummuter
Pred: commuter

C: A red train traveling past a white train.
DL: red train | blue white train | red door open glass red window | yellow line | steel railroad | empty train | white green sign | gray yellow platform | white green sign | empty train station
O: DLR Station
K: "...Passenger operations include Amtrak, Metra, the Chicago Transit Authority's 'L' and Chicago's South Shore Line trains. The museum had an earlier model railroad layout..."

Q: What is the man doing with his phone?

GT: watch video, picture, video tape, take photo
Pred: take picture

C: A person is holding up their cell phone to take a picture.
DL: up raised open holding white hand | raised up extended bent long thumb | red black phone | thin light hairy wrist
K: "...A selfie is a self-portrait photograph, typically taken with a smartphone which may be held in the hand or supported by a selfie stick...Smartphones can use their front camera (of lesser performance as compared to rear camera) facing the user for purposes like self-portraiture (selfie) and videoconferencing..."

Q: How might this be prepared?

GT: fried, pan, frypan
Pred: grilled

C: A plate of french toast and breakfast potatoes.
DL: grilled sliced fried cooked potato | brown wood table | sliced toasted grilled fried fish | silver white napkin | silver knife | glass dark black bottle | cooked red bacon
K: "...A baked potato, or jacket potato, is a potato that has been baked for eating. When well cooked, a baked potato has a fluffy interior and a crisp skin...French toast French toast is a dish made of bread soaked in eggs and milk, then fried..."

Q: How would you dress for this setting?
GT: short, bath suit, bikini, summer
Pred: swimsuit

C: A lot of seagulls flying around at the beach.
DL: sitting standing small walking black gray bird | cast black dark shadow | cloudy white blue sky | standing walking person | small white cloud | gray calm large blue water | black sandy wet sand | white flying gray seagull
K: "...Beach balls are also a popular prop used in swimsuit photography and to promote or represent beach-themed events or locations...The video featured one dancer and Kumi sporting several fashions, including a crop top with black shorts and gold chains and a bustier with tight-fitting leggings..."

Figure 3.17

CHAPTER 4

Planning: Interactive Environment and Hierarchical Reinforcement Learning

4.1 A Virtual Testbed for Physical and Interactive AI

We propose VRGym, a virtual reality (VR) testbed for realistic human-robot interaction. Different from existing toolkits and VR environments, the VRGym emphasizes on building and training both physical and interactive agents for robotics, machine learning, and cognitive science. VRGym leverages mechanisms that can generate diverse 3D scenes with high realism through physics-based simulation. We demonstrate that VRGym is able to (i) collect human interactions and fine manipulations, (ii) accommodate various robots with a ROS bridge, (iii) support experiments for human-robot interaction, and (iv) provide toolkits for training the state-of-the-art machine learning algorithms. We hope VRGym can help to advance general-purpose robotics and machine learning agents, as well as assisting human studies in the field of cognitive science.

4.1.1 Introduction

The past decade has witnessed a rapid development of categorical classification for objects, scenes, and actions, fueled by large datasets and benchmarks, discriminative features, and machine learning methods. Similarly, successes have also been achieved in many other domain-specific tasks, largely due to the ever-growing vast amount of labeled data and rapidly increasing computing power, combined with supervised learning methods (in particular, deep learning [HS06]). The performance of certain tasks has reached a remarkable level, even arguably better than human in control [DCH16, MKS15], grasp [MLN17, LLS15], object recognition [HZR15], learning from demonstration (LfD) [ACV09], and playing the game of go [SHM16] and poker [MSB17, BS18].

Despite the impressive progress, these data-driven feed-forward classification methods have well-known limitations, hindering the advancement towards a more general AI that can interact with human: (i) needing *large labeled training datasets*; (ii) often *task-specific* and view-dependent, which makes it difficult to generalize; (iii) lacking an *explicit representation and structure* to handle large variations exhibited in and outside of the training data.

In contrast, the hallmark of machine intelligence is the capability to rapidly adapt to new tasks and "achieve goals in a wide range of environments [LH07]". To achieve such intelligence, recent years have seen the increasing use of synthetic data and simulation platforms[1]. Advantages include: (i) the structure of the data is efficiently encoded *without the need for human labeling* as the simulation inherently comes with the ground truth; (ii) can accommodate different embodied agents (*e.g.*, humans, humanoid robots, or turtle-bots); and (iii) benchmark *generalization* in various tasks at a low cost.

Empowered by the gaming industries, tremendous amount of game contents, including scenes and objects, are made available for the virtual environment. Meanwhile, more sophisticated physics-based simulation engines and rendering techniques have enabled more realistic simulations. These characteristics allow a growing number of tasks to be performed using synthetic data in simulation platforms. Furthermore, some simulation platforms also become publicly available, such as AirSim [SDL18], AI2THOR [KMG17], Gibson [XZH18], *etc.*, promoting the further explorations and applications. In short, it is both the research and the engineering efforts that make it possible to achieve considerable successes in some AI tasks and applications.

[1] See a brief review in the supplementary.

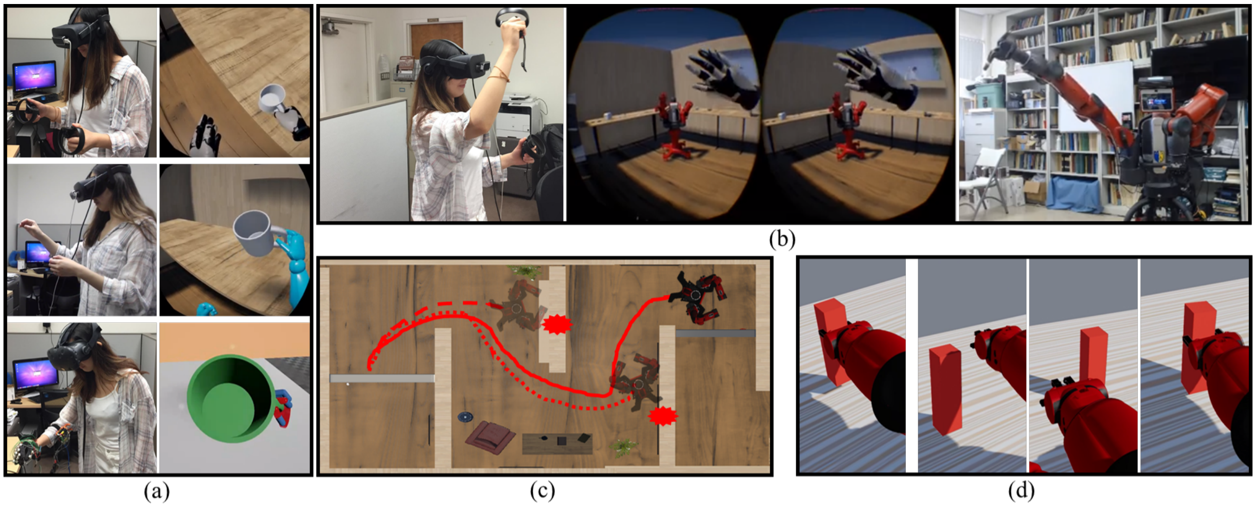


Figure 4.1: (a) VRGym integrates three types of input devices, providing human manipulation in an increasing resolution using Oculus Touch, LeapMotion, and a data glove, from top to bottom. (b) The VRGym-ROS bridge allows physical human/robot agent meet virtual agents inside a virtual world, providing the capability of social interactions. (c) The training of the robot navigation using reinforcement learning (RL) inside VRGym. The robot successfully navigates to the goal without collisions after about 10,000 episodes. (d) The learning of object manipulation using human demonstrations (leftmost) and inverse reinforcement learning (IRL) (right three) inside VRGym.

However, prior work often lacks the human involvement, especially in high-level tasks. For instance, although some virtual platforms (*e.g.*, OpenAI Gym [BCP16] and Mujoco [TET12]) allow to train a virtual robot to perform many manipulation tasks, they lack a human in-the-loop, thus cannot handle critical tasks like intention prediction and social interaction. Hence, having a simulation environment where a robot can interact realistically with a human and evolve incrementally could facilitate the robotics developments.

In this section, we propose VRGym—a virtual reality testbed, which combines VR with virtual training for both physical and interactive AI agents. By putting human in-the-loop, VRGym goes beyond the traditional synthetic data and simulation platforms by simulating a human-robot co-existing environment.

Specifically, VRGym tries to fill in the gap between the new advancement of VR and the need for training virtual agents to collaborate with human. In particular, we hope to address

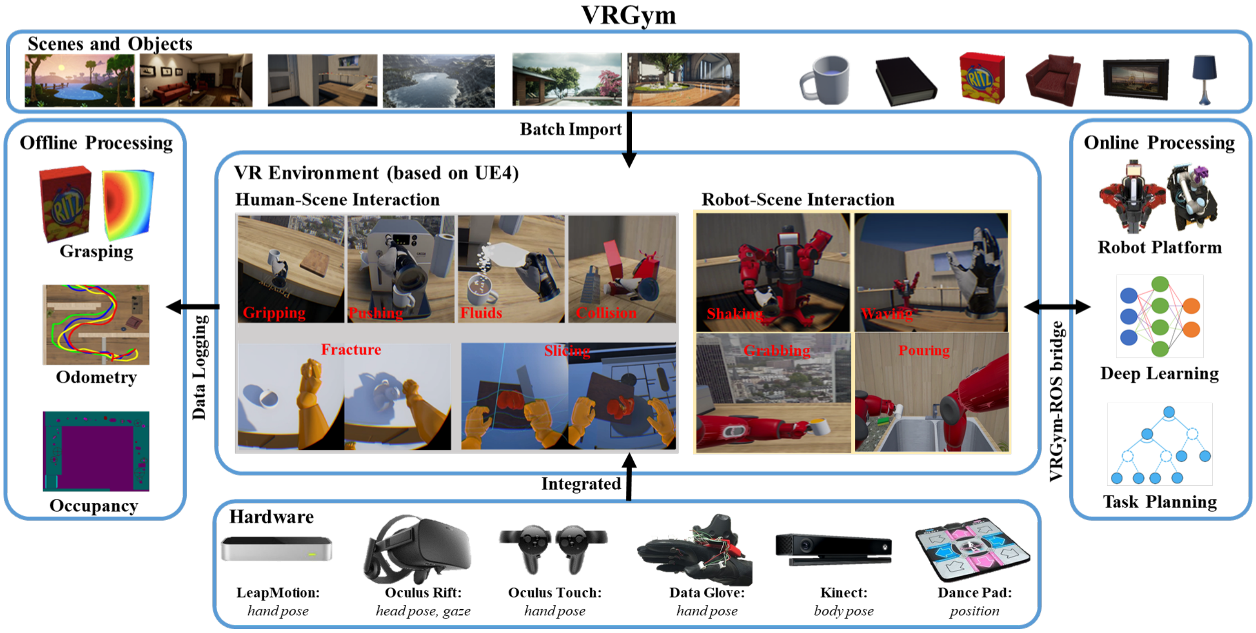


Figure 4.2: System architecture of VRGym, consist of three major components: (i) Hardware modules for human data input. (ii) Scene modules batch import various category of scenes as well as diverse objects, derived from different resources such as 3D modeling tools, scanned models, and automatically generated synthetic data. (iii) VR environment serves as an ideal testbed, where both a human and a robot can perform diverse tasks. The inherent physics-simulation engine enables realistic human-scene interactions and robot-scene interactions.

three critical issues. First, what is the best way to reflect human embodiment in VR; *i.e.*, how humans can genuinely interact with robots and how the robots can perceive related data that are sufficiently close to those in real life? Second, how to take advantages of current well-developed algorithms and models? Third, to which level of unique interactions the VR simulations can afford? To answer these questions, VRGym is designed to push the limits of current akin simulators by offering the following characteristics.

Fine-grained human embodiment representation Adding a real human in the simulation is not a trivial task. Most of the current simulation platforms only support either scripted or limited remote-controlled human models. In VRGym, we integrate a multi-sensor setup as alternatives to traditional VR input devices. Our setup is capable of providing a whole-body sensing and reflecting the measured data on a detailed human avatar. As a result, the simulation can account for both body and hand poses during interactions. Figure 4.1a shows different resolutions of manipulations in VRGym.

High compatibility with existing robotics systems and algorithms In VRGym, we build an efficient bi-directional communication interface with the Robot Operating System (ROS). Figure 4.1b depicts an example of how a person interacts with a robot in VRGym, supported by the VRGym-ROS bridge. As a result, all ROS-compatible resources can be used in VRGym with little effort, which allows easy setups, training, evaluations, and benchmark.

Multiple levels of interactions By providing the fine-grained human embodiment representation and the ROS integration, various interactions between humans and autonomous agents are made possible in different resolutions. VRGym supports interactions as simple as only providing visual/perception information and as sophisticated as learning complex robot grasping from human demonstrations. Figure 4.1c shows how an agent obtains a navigation policy using RL, and Figure 4.1d shows learning a grasp policy using IRL.

VRGym makes the three contributions:

- A comprehensive simulation platform that integrates UE4 built-in functions, *e.g.*, scene, physics-based simulation, rendering, basic human inputs, with customized developments, aiming to facilitate a variety of AI researches.
- A multi-sensor hardware and software setup that allows the whole body sensing and reflects human subjects to virtual embodiments with great details. The generated data can be seamlessly logged for online and offline training purposes.
- VRGym-ROS bridge enables a bi-directional data communication. Through this interface, AI researchers can take advantages of the existing robotics models and algorithms. Similarly, robotics researchers can utilize more sophisticated physics-based simulation.

4.1.2 VRGym System Architecture

Figure 4.2 illustrates the system architecture of the VRGym. VRGym offers a variety of realistic scenes and tasks for both humans and robots, and provide automatic logging of the data during agents performing tasks. This capability is provided by the integration of three main modules: (i) scene module which renders user-specified 3D scenes and objects, (ii) VR environment based on UE4 with physics-simulation engine, introducing various physical properties that enrich tasks and data, and (iii) VR hardware module that imports a human agent's state and command to the VRGym. We now further elaborate each module in the following subsections.

4.1.2.1 Scene Module

Scenes and objects are the building blocks for a simulation environment. In order to increase the variety of environments for VRGym, we develop several pipelines to import or create scenes into VRGym based on the users' specifications. The scene module enriches static environments for VRGym. Note that the ground truth of RGB image, depth image, surface normal, and object label come automatically with the scene module in real-time, enabling the training for machine learning models and robotics applications.

Specifically, VRGym can directly import the entire 3D scenes provided in large open-source datasets, either collected from the web [SYZ17, CDF17] or automatically generated from a given set of objects [YYT11, QZH18, JQZ18] (see top of Figure 4.2). Additionally, VRGym also supports manually constructed scenes (see Figure 4.4) for more specific tasks, where neither the open-source scene dataset or the automatically generated scenes could satisfy such constraints.

Similarly, individual objects can be imported to VRGym from mesh files, which can be obtained from open-source CAD datasets (*e.g.*, [CFG15, CWS15]). Customized or complex objects can be manually created or scanned using a RGB-D sensor to import to VRGym for

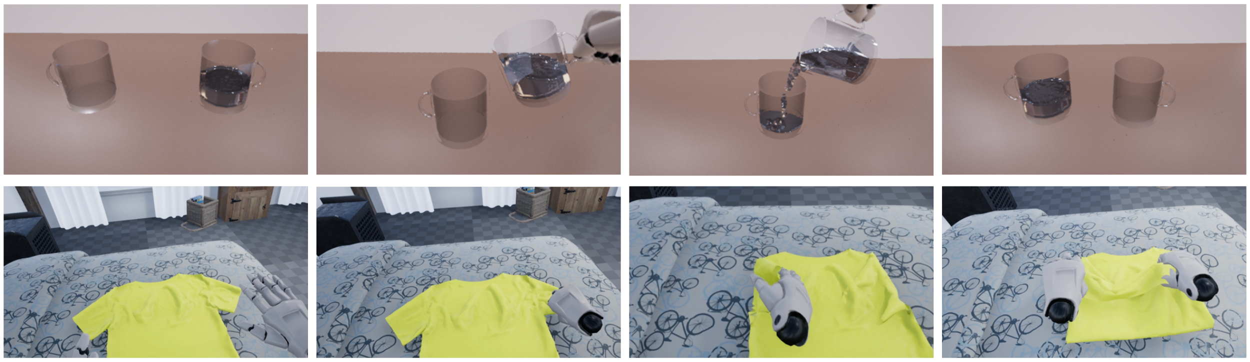


Figure 4.3: Examples of various physics-based simulation for diverse tasks in VRGym beyond merely rigid-body simulation in other 3D virtual environments. (Top) Pouring water. (Bottom) Folding clothes.

specific tasks. After the import, users can further adjust static meshes, textures, materials, and collision boundaries of the objects.

4.1.2.2 Real-time Physics-based Simulation

We choose UE4 as the simulation engine for VRGym for its advanced real-time physics-based simulation. Unlike previous 3D virtual environments that mostly focus on rigid body simulation or symbolic-level event simulation, VRGym integrates the advanced simulation provided by UE4 to enable a large set of various simulations, including rigid body, soft body, collision, fluid, cloth, slicing, and fracture. Some examples are shown in Figure 4.3 and the center of Figure 4.2. As a result, subtle object state or fluent [NC36] changes due to the virtual agent's actions are realistic and diversify. Integrating with such sophisticated physics-based simulations, VRGym not only increases the task complexity and improves the visual experience of human agents, but also affords more complicated task simulations for both virtual and physical robots.

4.1.3 Human Embodiment in VRGym

Compared to other similar 3D virtual environments, VRGym has another distinct feature; *i.e.*, introducing the capability to represent the physical human agent's embodiment in real-time as an avatar in the virtual environment. To reflect human movements and manipulations accurately, the physical human agent is tracked in real-time, resulting in a humanoid mesh that can deform accordingly based on the underlying tracked body skeleton and the hand poses.

Specifically, the setup includes: (i) A Kinect One RGB-D sensor to map human skeleton to the avatar in real-time through a customized-built Kinect plugin developed in UE4, (ii) an Oculus headset to record the head pose, (iii) a dance pad to navigate the avatar inside a large virtual world, and (iv) three types of input devices that provide manipulation information in different resolutions. Compared to other platforms, VRGym emphasizes the capability for users to interact with virtual environments. Depending on the needs, the user can use one of the three input devices for manipulation:

- Oculus Touch Controller offers an attachment-based approach; *i.e.*, the virtual object will automatically attach to the virtual controller/hand once the user triggers the grasp event. It enables a firm-grip manipulation, providing a firm but the least realistic grasp during the human-object interaction. Such manipulation is effective in the event-level tasks where the fine-grained hand pose is not required; *e.g.*, pick and place.
- The commercial hand pose sensing products (*e.g.*, LeapMotion) provide the vision-based gesture recognition. It is a low-cost and off-the-shelf solution that can be easily set up by mounting the sensor on the head-mounted display. However, it is difficult to have a firm grasp due to occlusions and sensor noises. Note that the hand tracking will fail if the hand is not within the view.
- An open-sourced glove-based device [LZX19] is also compatible with VRGym to provide the finest-grained manipulation. It requires a Vive Tracker to provide global positioning

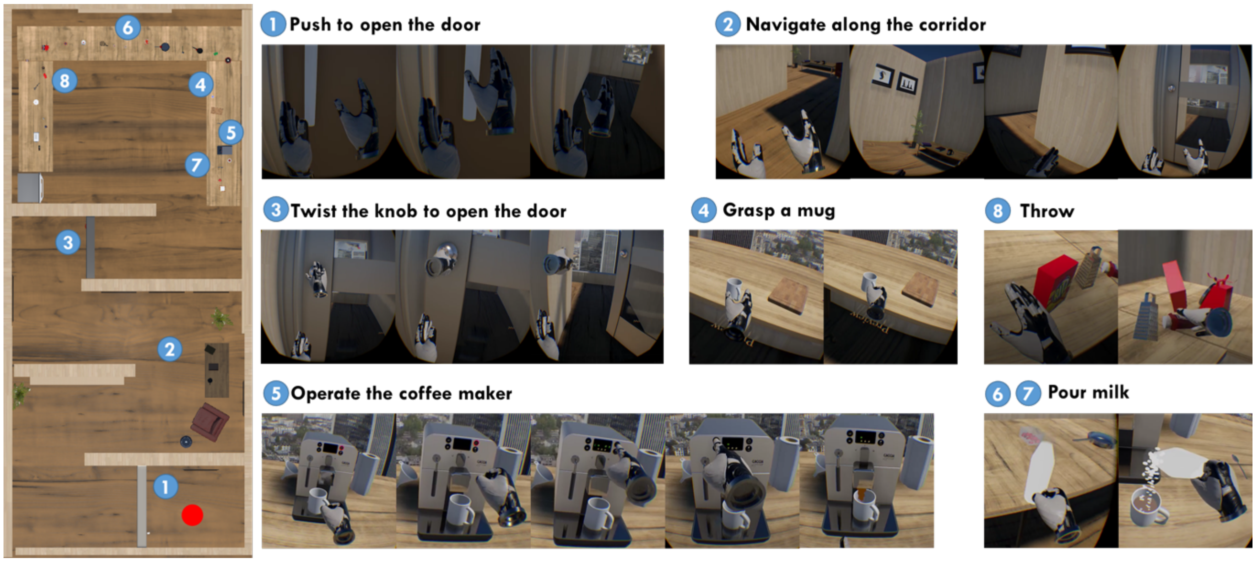

6
8
4
5
7
3
2
1
1 Push to open the door
2 Navigate along the corridor
3 Twist the knob to open the door
4 Grasp a mug
8 Throw
5 Operate the coffee maker
6 7 Pour milk

Figure 4.4: A human agent performs a series of actions in a virtual scene using Oculus Touch controllers. (Left) Action sequence from a top view of a virtual indoor environment. (Right) Sequences of the performed actions. Specifically, the human agent starts at the red dot as shown in the left, (1) pushes a door, (2) navigates along the hall, (3) twists a door to enter the kitchen, and (4)-(7) makes a cup of coffee. This process involves (i) large movements using the human embodiment provided in VRGym (navigating along the hallway), (ii) complex operations (operating the coffee maker), (iii) fine-grained manipulations (twisting the doorknob), and (iv) physics-rich controls (pouring milk).

of the hand, and an IMU network in the glove to measure the rotation of each phalanx and calculate the hand poses using forward kinematics. Although glove-based devices are costly compared to other alternatives, they allow reliable hand pose sensing, which is vital for the tasks with detailed, complex and subtle hand manipulations.

4.1.4 Software Interface Design

VRGym has two major software interfaces developed to enable training and benchmarking both physical and interactive AI agents. The first interface is the human data logging system that builds on top of the hardware setups to collect the data generated during the interactions between the avatar and the environment. Another interface, a VRGym-ROS bridge, is introduced to allow seamlessly import of robot models and robotics algorithms from ROS. The collected data together with the VRGym-ROS bridge could be used for a variety of AI applications; see examples in subsection 4.1.5.

To demonstrate the functions of these two interfaces, we consider a task-rich environment built for the VRGym. Figure 4.4 depicts an environment in VRGym that provides semantically-diverse tasks to the agent. Note that although such environment could be constructed in the real world to perform the demonstrated tasks, sensing and logging the detailed data generated during the interactions between the agent and objects would be extremely difficult in practice.

In such a typical virtual environment in VRGym, an agent (a human as an avatar or a virtual robot) is initially placed on the starting point, indicated as the red dot in Figure 4.4. The final goal for the agent is to reach the kitchen located at the far-end, and accomplish several sub-tasks. At the beginning, the agent has to push to open the first door and navigate along the corridor, requiring *large movements*. Then the agent must go through another door to enter into the kitchen, and the only solution is to twist the doorknob using complex *manipulations*. Inside the kitchen, the agent is required to make a cup of coffee with milk, which needs to grasp and move a mug, operate the coffee maker by pushing several buttons in a certain order. The entire procedure requires the *task planning* empowered by the *physics-based simulation*.

4.1.4.1 Human Data Logging

When a user performs a task, data generated by the interactions between an avatar and the environment can be directly logged with ground-truth labels in VRGym. In this section, we showcase two scenarios where the data is logged and used in other applications.

Grasping Finer-grained manipulation is made feasible in VRGym using a glove-based device [LZX19]; see Figure 4.5a for some results. By collecting a set of subjects' grasp data on a variety of objects, we can merge all the collected grasp data to form heat maps on

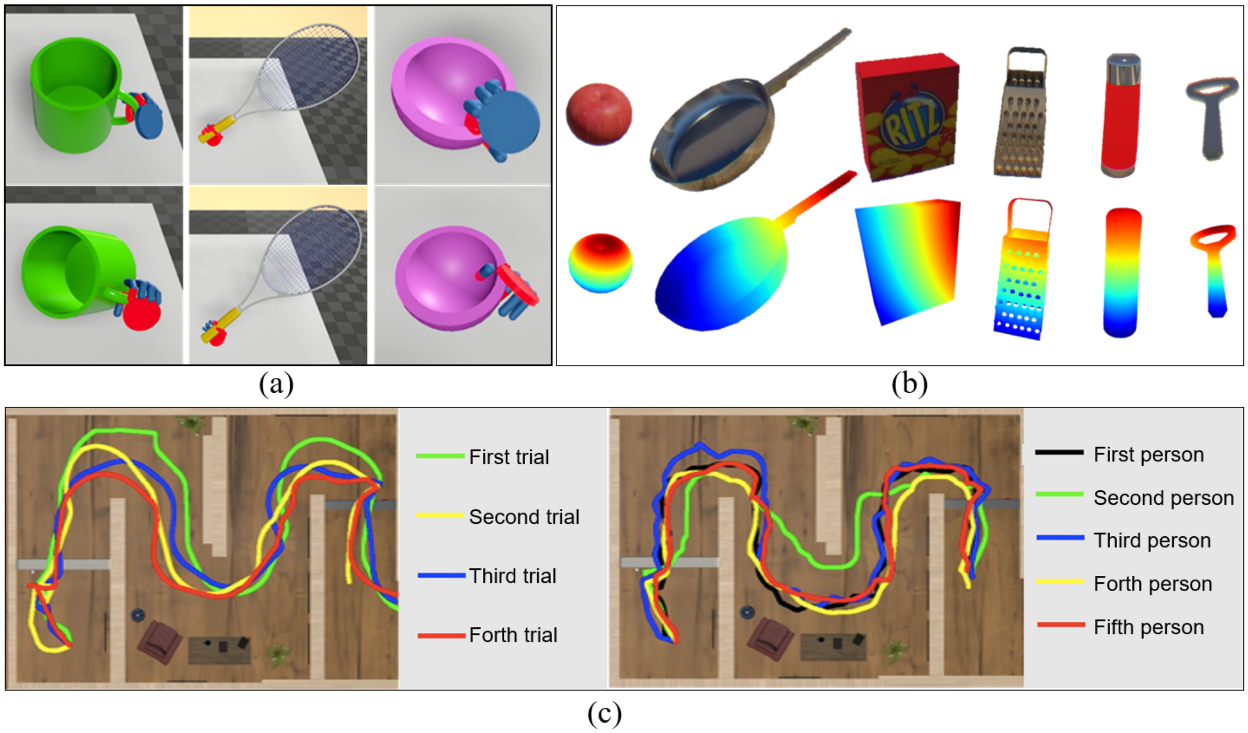


Figure 4.5: (a) Grasp a mug, a tennis racket, and a bowl. The red area indicates the contact force between the virtual hands and the object. (b) Visualization of the collected human grasp data. Top: a set of 3D objects. Bottom: the average grasp heat map generated by multiple subjects. (c) Visualization of footprint from different subjects.

different objects to visualize the likelihood of grasp points on man-made objects. Specifically, the grasp data shown in Figure 4.5b is the averaging data of heat maps collected from 10 human subjects, where the hotter the area is, the denser the grasp points are, and the more likely a human agent would grasp around that area.

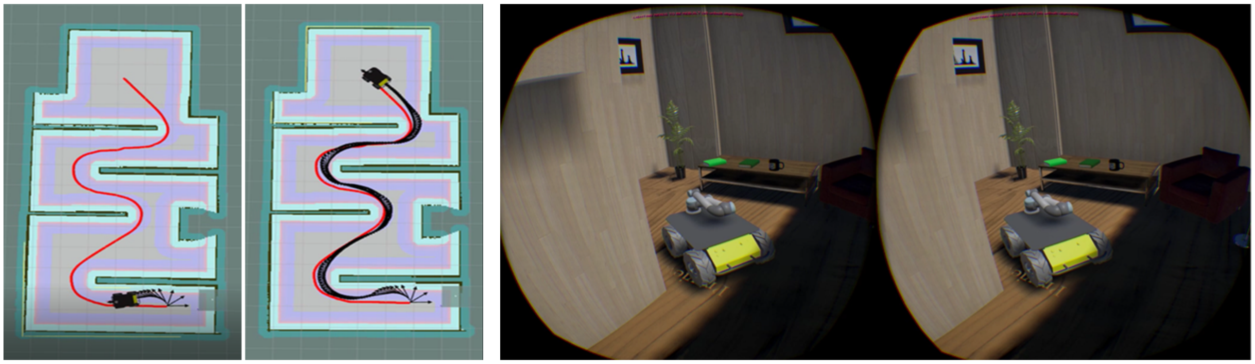
Footprints VRGym provides the function to log an agent's footprints or the odometry data. Figure 4.5c shows the recorded odometry data from 5 human subjects who have limited VR experience. Each of the participants navigates from the starting point to the kitchen room along the corridor using Oculus Touch controllers.

4.1.4.2 ROS Interface

The VRGym is compatible with the popular ROS framework through a customized VRGym-ROS communication bridge. This bridge allows the off-the-shelf ROS robot models to communicate with the simulations and human agents in VRGym with minimal efforts; *e.g.*, the diverse scenes rendered in VRGym can also be exported to the Gazebo simulator, which is highly compatible with ROS.

Implementation We develop a ROS interface, VRGym-ROS bridge, based on the TCP/IP protocol in order to enable VRGym to communicate with the existing popular robotics platforms. Through this interface, robot body parts can be easily imported to VR environments as mesh files and control signals, and a data stream can be seamlessly transferred between the VRGym and the robot platforms using ROS to communicate with either physical or virtual robots. We organize all data types (*i.e.*, ROS topics) in a unified JSON format and construct JSON parsers in both VRGym and ROS to further improve the compatibility. Each port in the protocol supports a stream of data, making it possible to present multiple agents from ROS into the VRGym. With the VRGym-ROS bridge, we present two examples of training and evaluating human robot interactions (HRI) inside VRGym in subsection 4.1.5, which incorporates direct human reactions and involvements. Such capability is largely missing in the current robotics simulators such as Gazebo or V-Rep. The benchmark in subsection 4.1.5 is also supported by this VRGym-ROS bridge.

Evaluation We evaluate the VRGym-ROS bridge on a navigation task (see Figure 4.4) using a Clearpath Husky robot. This navigation task is performed in VRGym, whereas the robot model is imported from ROS, making it possible to evaluate a number of SLAM algorithms and path planning approaches. In Figure 4.6a, the mapping result is obtained using the conventional GMapping package in ROS. The red curve indicates the planned path, whereas the black curve is the actual odometry of the Husky robot. Figure 4.6b shows the



(a) (b)

Figure 4.6: VRGym-ROS bridge. (a) The robot navigation in the scene imported into the Gazebo, exported from the VRGym. The red curve indicates the path planned by the robot's global planner. The black curve is the actual trajectory executed by the robot. (b) A Husky-UR5 robot is imported into VRGym from ROS to guide the way and open the door for a human agent.

user's view when the robot is moving. This VRGym-ROS bridge fills in the gap between the diverse scenes in VRGym and the existing fine-tuned algorithms provided in ROS.

Communication Bandwidth To evaluate the reliability and efficiency of the VRGym-ROS bridge, we conduct an experiment by sending packages with the size of 512Kb[2].

4.1.5 Experiments

In this section, we demonstrate the performance and capability of the VRGym from four different perspectives[3]. Two human robot interaction (HRI) applications are conducted, including a human intention prediction task and a social interaction task. Like other testbeds, we also benchmark the performance popular machine learning algorithms (*e.g.*, reinforcement learning and IRL) in the VRGym.

[2] See a detailed evaluation in supplementary.

[3] See a video demo at Vimeo.

4.1.5.1 Experiment 1: Intention Prediction

Predicting human intention is difficult when training on a physical robot since this task has very small error tolerance; wrong predictions may endanger both the human and the robot. It is particularly interesting to study human intention prediction in VRGym, since this problem involves complicated inference process that many types of data can be useful: human trajectories, human poses, object positions, object states, and first/third-person vision inputs, *etc.* Predicting intention is made possible in VRGym as our unique multi-sensor setup reflects human poses, and the odometry data provided by the data logging system indicates human's trajectories.

In the experiment, we analyze different human intention prediction algorithms to demonstrate the potential of VRGym as a testbed for both physical and virtual AI agents. Additionally, we show the unification of both the learning and the inference enabled by the VRGym. 20 subjects are recruited. The virtual environment is set up as a virtual kitchen, in which more than 20 objects are placed on top of three long tables. The layout of the kitchen is shown in Figure 4.7, where the agent starts from the entrance of the room (red dot) and performs the task with at least 4 steps: grasp a mug, operate the coffee maker, add milk, and add sugar. Note these tasks can perform in different orders. The resulting footprint from one subject is plotted in Figure 4.7. All subjects are required to perform a coffee-making task—making a cup of coffee using the available objects.

Figure 4.7e illustrates the comparisons among these three methods. The qualitative results are shown in Figure 4.7a-d to reveal the intention of the agent as the heat maps during the process of making coffee, where hotter color (red) indicates higher probability. This high-level semantic prediction is inferred given multiple human demonstrations as logged navigation and grasp data collected from the agent using VRGym.

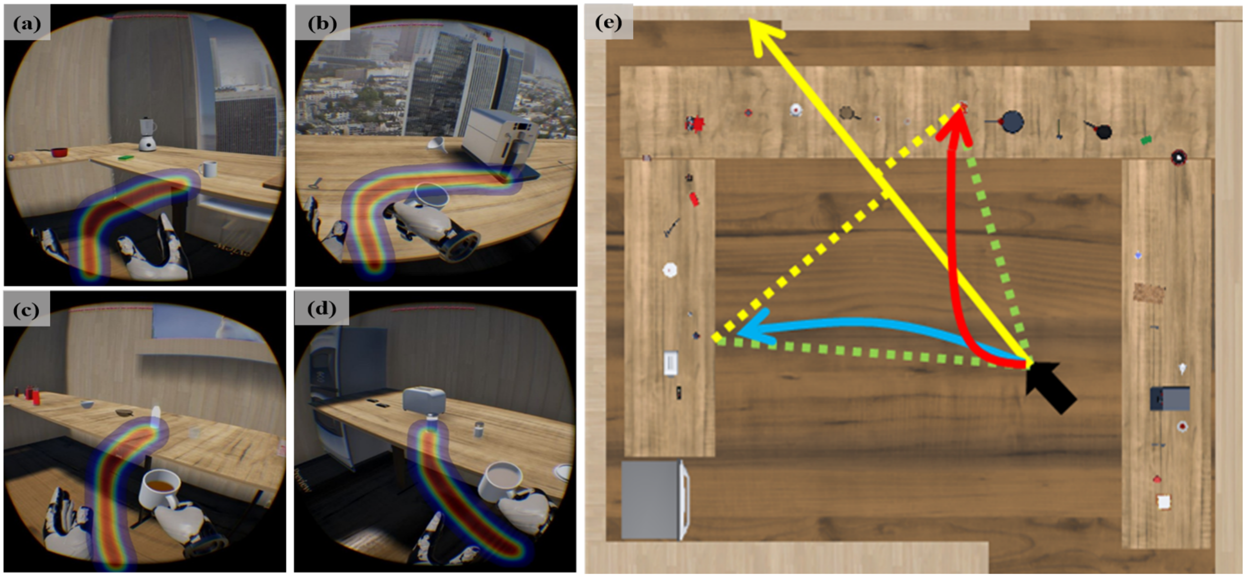


Figure 4.7: Intention predictions in a coffee-making task. (a) Grab a cup. (b) Use the coffee machine. (c) Pour milk. (d) Add sugar. (Right) Visualization of three intention prediction algorithms. Blue and Red: sampled paths from the grammar model [QHW17]. Green: straight-line distance. Yellow: prediction by shortest perpendicular distance (dashed lines) from objects to the ray direction (solid arrow) based on avatar's location.

4.1.5.2 Experiment 2: Social Interaction

Social interactions or social HRI is a vital topic enabling human-robot co-existing environment, since the robot needs to understand and respond properly to human's social behaviors, such as waving and hand-shaking. Although the current robot simulators (*e.g.*, Gazebo and V-Rep) provide a suite of features, one key element these simulation platforms still largely missing is the direct human involvement which is crucial for human-robot interaction studies.

Participants A total of 10 subjects were recruited. We implemented the algorithm proposed in [SGR17] for robot learning social affordance. The algorithm is briefly described as follows; we refer the readers to the original paper for more technical details.

Results Qualitative results are shown in Figure 4.8. Concretely, the robot starts hand-waving in response to the agent's hand waving (Figure 4.8a), illustrated by a virtual hand

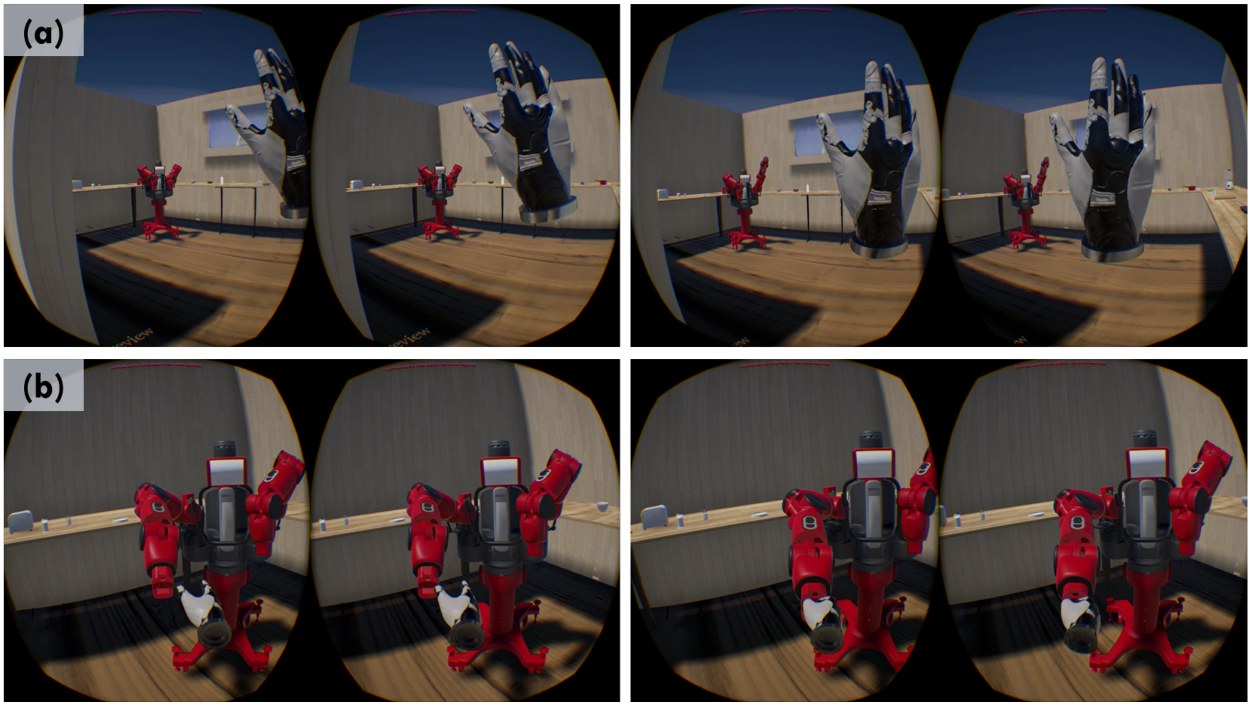


Figure 4.8: Human robot interactions in VRGym. A Baxter robot (a) waves hands and (b) shake hands with a virtual human agent.

model. The robot stretches out its manipulator to make a handshake with the virtual agent (Figure 4.8b). Technically, when the virtual Baxter inside the VRGym perceives the action signals from a virtual human such as *hand shaking* or *hand stretching out*, it sends the action signals to ROS through the VRGym-ROS bridge. In ROS, the motion planning will generate corresponding body parts transformations and send the computed transformation data back to the virtual Baxter inside VRGym, such that it will then act with the appropriate responses to the virtual human agent. In this sense, the proposed VRGym enables a new approach to study social human-robot interaction without using a costly physical robot or having a physical contact between a subject and robots, which in some cases could be dangerous.

4.1.5.3 Experiment 3: RL Algorithms Benchmark

We introduce a playground as a sub-module (Figure 4.9) inside the VRGym, aiming to train robots to navigate in a 3D maze-like indoor corridor. The overall goal is to teach the robot agent itself by trial and error to obtain a navigation policy, reaching the final goal of the maze. The learning strategy applied on the virtual robot follows the standard RL framework. A Baxter robot is integrated into the VRGym and controlled by off-the-shelf ROS packages.

Compared to other virtual playgrounds (*e.g.*, OpenAI Gym), the proposed VRGym differs in two primary aspects.

- *Sophisticated Interactions.* With the advanced physics-based simulator, the VRGym offers realistic interactions between the virtual agent and the virtual environment.
- *Physical RL Agent.* Since the VRGym is capable of importing both the physical and the virtual robot model to the virtual scene, it is feasible to transfer RL model trained inside the virtual environment directly to a physical robot agent.

We conduct four state-of-the-art deep RL algorithms to demonstrate the VRGym's capability in RL related tasks. These algorithms are **DDPG** [LHP16], **DQN** [MKS15], **Actor-**

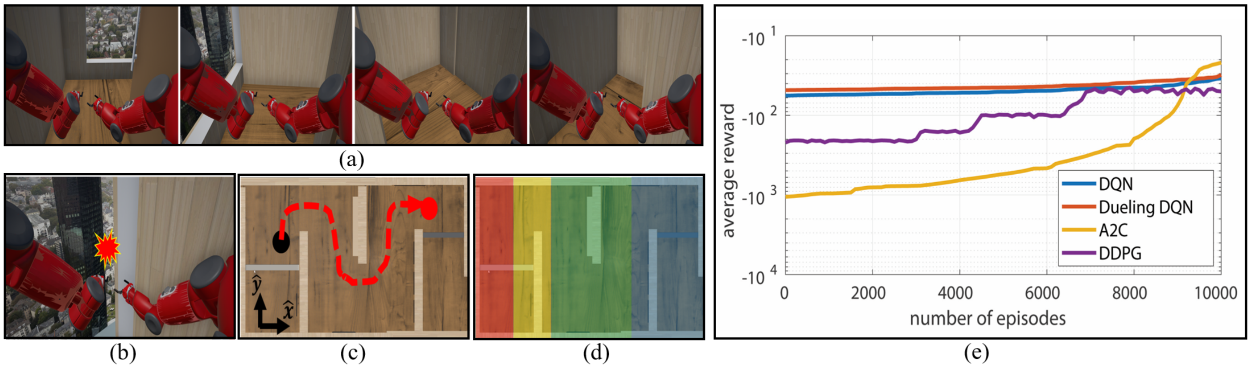


Figure 4.9: Settings for the RL training inside VRGym environment for an indoor maze navigation task. (a) First-person view of a virtual robot. (b) The robot collides with a wall, triggering negative rewards. (c) An eagle view of the indoor navigation task. (d) Rewards assigned in different color zones (red, yellow, green and blue) from low to high. (e) The performances of the tested RL algorithms.

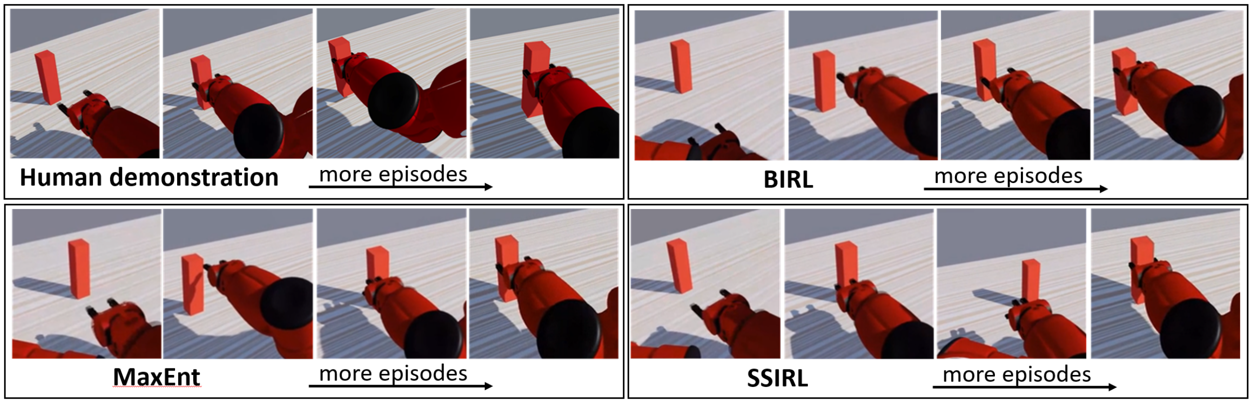


Figure 4.10: Learning human grasping demonstration with different IRL frameworks.

Critic [MBM16], and **Dueling DQN** [WSH16]. All four algorithms use the pixel-input from the first-person camera view. The quantitative comparison of the above four algorithms in VRGym is plotted in Figure 4.9e.

4.1.5.4 Experiment 4: IRL for Learning Grasp

Grasp is an imperative capability for an interactive agent. In this experiment, we adopt an inverse reinforcement learning (IRL) framework to enable a virtual robot learning to grasp from human demonstrations. This task primarily involves both the data logging function in VRGym and a ROS motion planer communicated by the VRGym-ROS bridge. The robot is expected to learn how to successfully grasp an unknown object based on the hand trajectories demonstrated by the human subjects, collected through tele-operations using the Oculus Touch Controller inside the VRGym.

The trajectories of the human demonstrations are logged and used to infer the model and its parameters. Later, with the learned model and its parameters, the robot can be executed using the motion planner in ROS to grasp an unknown objects in the virtual environment.

Three IRL algorithms are implemented in the VRGym: Bayesian-IRL [RA07], Maximum Entropy-IRL [ZMB08], and Semi-supervised-IRL [VGL12]. Qualitative results are shown in Figure 4.10.

4.1.6 Conclusion

In this section, we introduce the VRGym as a promising simulation platform for training and evaluating autonomous agents to build the physical and interactive AI. VRGym can represent a fine-grained human embodiment as a virtual avatar using a range of hardware setups for body and manipulation sensing. Existing robotics systems and algorithms developed in ROS can also be integrated to VRGym through a customized VRGym-ROS bridge. Multiple evaluations indicate that the VRGym has a robust performance at the system level and in the communication with ROS. Our experiments have demonstrated that four different robotics interactive tasks can be successfully trained using RL and IRL inside VRGym. Specifically, we showcase how the data logged from the VRGym is useful in several interaction tasks, combining with the functions (*e.g.*, motion planners, robot models) provided by ROS through the VRGym-ROS bridge. The successful implements of RL and IRL for robotics interactive tasks in VRGym also support the training capability offered by VRGym in training robots with advanced machine learning methods. We believe VRGym could have further potential applications and it will benefit future research on the physical and interactive AI.

4.2 Exploration-Efficient Hierarchical Reinforcement Learning

In this work, We adopt the Boltzmann Policy [SB98] to generalized Boltzmann policy (GBP) in order to leverage additional knowledge for exploration in a unified formulation. Furthermore, we integrate the GBP with a simple yet effective hierarchical reinforcement learning (HRL) [KNS16] framework. Both theoretical and empirical analyses are provided to justify the (i) exploration efficiency and (ii) dynamic balance between exploration and exploitation for the proposed method. Experiments are conducted in various 2D grid-world environments under the sparse-reward circumstance, in which discovering reward is difficult due to the delayed reward and long horizon. The results show significant improvements in exploration efficiency across the hierarchy comparing to the vanilla HRL framework.

4.2.1 Introduction

A reinforcement learning (RL) agent learns a policy to maximize long-term accumulated reward. Interacting with the environment, the agent collects experiences (exploration) and learns from them (exploitation). Such "trial and error" procedure is driven by a reward function which is usually crafted. [SB98]. Researchers have proposed classic solutions for RL, such as Q-Learning [Wat89], policy gradient [SMS00], *etc*. Recent success in deep learning also boosts the research of RL, providing a powerful functional approximation [MKS15].

RL models heavily hinge on the design of the reward function. On one hand, hand-crafted reward functions are oftentimes not suitable for complex tasks as a poorly designed reward function would potentially confuse the RL agent, dragging it into a dilemma. On the other hand, simple reward functions also baffle the RL agent; *e.g.*, a "0-1" reward can be extremely difficult for an RL agent to learn since the agent would receive the reward if and only if it reaches a long-term goal, before which the agent has no clue about the task. Such cases are so-called sparse-reward or reward-delay circumstances, requesting for a more efficient way to explore the state-action space.

Additionally, it is challenging to balance the exploration and the exploitation in RL. To alleviate this difficulty, one commonly adopted trick is to empirically set a schedule, such as ϵ-greedy in Q-Learning. Nevertheless, the drawback of a manual exploration schedule is fatal. The RL agent could hardly achieve the optimal policy if one does not have sufficient exploration, while the late stop of the exploration causes a plight to seldom converge to a deterministic or optimal policy. Therefore, it is advantageous that the RL agent, through learning, could properly balance the exploration and exploitation by itself without any manual intervenes.

In this work, we adopt Boltzmann Policy [SB98] to generalized Boltzmann policy (GBP) by taking the advantages of auxiliary information to (i) improve the exploration efficiency, especially under sparse-reward environments. Such auxiliary information can be interpreted

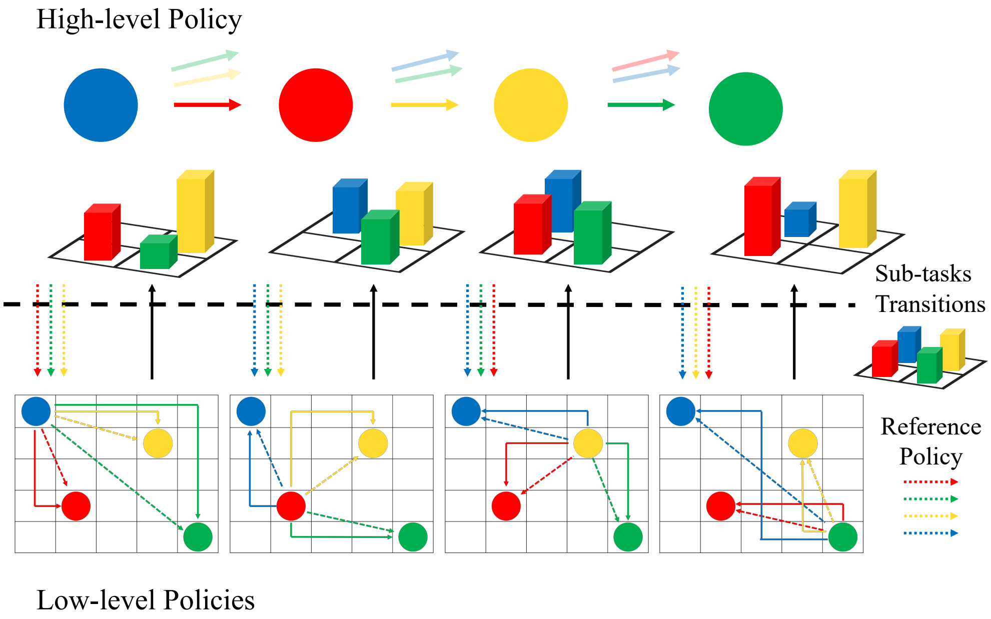


Figure 4.11: An illustration of the proposed method. The example task requires the agent to visit the key points in the order of blue, red, yellow, green to get the reward. The dashes and the bar-charts represents the auxiliary information in GBP for low- and high-level policies respectively. The dashes inspire the low-level policies to explore along meaningful directions while the bar-charts encourage the high-level policy to explore towards unfamiliar sub-tasks.

accordingly as heuristics e.g. euclidean distance, a pre-trained policy etc.. Furthermore, in order to balance the exploration and exploitation, we introduce a preference factor, a learnable parameter, to control the exploration and exploitation.

In the literature, the challenges caused by the sparse reward serve as one of the major motivations to propose the hierarchical reinforcement learning (HRL) framework [DH93, SPS99, Pre00, CBS05, PR98]. The main idea of HRL is to construct two (but could be more) layers of policies, responsible for different granularity of tasks. The high-level policy is designed to arrange the sketch of the task and to decompose a complex task into multiple but simple sub-tasks. Policies on the low-level layer are assigned to achieve various sub-tasks, controlling the details of executions. In such HRL frameworks, policies on both layers can be learned individually using typical RL methods, but the reward function is designed differently. Low-level policies receive an artificial reward, namely the intrinsic reward [CBS05], when a low-lever policy finishes its sub-task. The high-level policy will receive the actual reward from the environment—the extrinsic reward, when policies on both layers jointly complete the correct sketch of sub-tasks.

Although the HRL framework alleviates the reward-delay problem to some extent by decomposing it into sub-tasks, it still struggles if the task complexity increases due to (i) the absent or insufficient information exchanges between high- and low-level policies, resulting in (ii) the inefficient exploration across the hierarchy, wasting computational efforts on insignificant sub-tasks.

Such deficiencies widely exist in the current HRL approaches. Crucially, policies from different layers in the hierarchy explore individually without using the information from other layers to prune the space, leading to a low exploration efficiency.

To remedy such drawback, an exemplar of the hierarchy would enable the communications across the hierarchy: low-level policies should be able to receive hints guided by the high-level policy, while the high-level policy focuses on the transitions of sub-tasks. As shown in Figure 4.11, we introduce generalized Boltzmann policy (GBP) as the representation of the policy on different layers, enabling HRL frameworks with proper communications across the hierarchy.

Specifically, we improve HRLs by fusing GBP in two aspects: (i) GBP is capable of accommodating auxiliary information of the task, making it possible to inject information from another layer. Such auxiliary formulated as a reference policy, guiding the policy on a different layer to explore more deliberately, *i.e.*, narrowing down the exploration space. (ii) Because GBP affords a learnable mechanism to dynamically balance the exploration and the exploitation, it can further improve exploration efficiency in HRL. Intuitively, for high-level layer, GBP not only encourages the high-level policy to explore insufficiently learned sub-tasks, but also rolls out the sub-tasks which are not critical to the entire task with probabilities. For low-level layer, GBP integrates information from high-level policy into its reference policy, which guides the low-level exploration towards more meaningful directions.

See details in subsection 4.2.4.

This work makes two major contributions:

- We adopt the generalized Boltzmann policy (GBP), making use of auxiliary information to explore more efficiently. Meanwhile, GBP is able to dynamically balance the exploration and the exploitation.
- We introduce GBP to serve as an unified representation of policies on both high- and low-level layers in the hierarchical reinforcement learning (HRL), enriching the information exchange, thereby improving the exploration efficiency across the hierarchy.

4.2.2 Related Work

Reinforcement Learning with Auxiliary Information is designed to have an improved exploration strategy by utilizing existing knowledge to better guide the exploration.

Early work uses heuristics-based methods. [IRC98, MRI04] propose supervised RLs that assign weights to different exploration steps based on the prior knowledge obtained. [SBP04] proposes the supervised actor-critic model, which adds a supervisor and attempts to find a balance between the supervisor and the critic for policy learning. [AHL07] proposes several exploration methods based on the rule of "exploring the most unvisited state".

More recently, one important line of research uses curiosity-driven methods [SP12, BSO16]— to explore the spaces which are infrequently visited. Usually, it requires a proper division of the task space, making it tricky to assign the granularity for sub-spaces.

In contrast, the proposed GBP incorporates auxiliary information in a more versatile form, *i.e.*, a stochastic policy. There are other approaches that also adopt an exponential form of policy, *e.g.*, max-entropy inverse reinforcement learning [ZMB08] and soft Q-Learning [HTA17]. The GBP is motivated differently: we want to incorporate auxiliary information to guide exploration.

Hierarchical Reinforcement Learning (HRL) has been intensively studied in the RL community. One important line of work is the formulation of semi-MDPs and the options. [SPS99, CBS05, SLB10, Pre00] proposed a semi-MDP planning method by considering the sequential transitions within sub-tasks, and the option has been proven to be an effective method to build hierarchies, capable of discovering the differences between action-values and option-values. Although recent work on options achieves impressive performance [BHP17], there are still unresolved challenges. In such a formulation, the options and terminal conditions lie within the same environment for the agent to discover. This setup requires the agent to explore immersively in the environment to distinguish options and actions, which in turn generates the hierarchy [MMH04, MB01, SWB05, MMS02].

To improve this hierarchical learning scheme, several directions have been investigated. Efforts have been made to propose more explicit representations of hierarchies. For instance, [DH93] proposes a model using various levels of granularity, in which policies on higher levels make decisions for lower layers until tasks cannot be further decomposed. Similar work introduces other methods to decompose tasks [Die00, GKP03], including some recent work with promising results using deep neural networks [VOS17]. Despite the granular hierarchical representation, the true sub-goals are still left for the agent to discover. Such a setting could improve the learning efficiency of an HRL agent, but would still suffer from the inefficient exploration.

In another stream of work, [PR98] uses a high-level representation of the entire task to represent the joint policy as an automaton. [MB01, MRL05] integrate RL to fulfill the low-level details of high-level symbols; such symbols are usually manually engineered. In more recent works, [KNS16, TGZ17] follow the same idea to leverage high-level state representations to improve HRL training. Different from HAM framework, instead of using a planner for the high-level symbolic planning, these works use RL policies in both the high- and low-level layers, but only explore individually. Meanwhile, [AKL17, SXS18] propose the idea of "policy sketch" to guide the low-level learning. Different from HAM, the high-level policy rolls out the useless sketches, expediting the exploration. [BSO16] uses a count-based curiosity for a more efficient exploration of the policy on the high-level layer. However, the high-level representations used in such methods still only help decide which low-level policies to use, barely improving the exploration efficiency for each individual low-level policy.

We argue that an ideal HRL should enable proper bi-directional communications between high- and low-level of layers to explore more efficiently. In the proposed method, the high-level policy provides guidance for the low-level policies to efficiently finish sub-tasks through GBP, formulated as a reference policy guided exploration in a sub-task space. Meanwhile, the transition of low-level sub-tasks can be passed to high-level policy to encourage explorations of the sub-tasks that are crucial for the entire task.

4.2.3 Generalized Boltzmann Policy

4.2.3.1 Preliminaries and Notations

A standard **reinforcement learning (RL)** agent learns a policy to maximize its accumulated rewards through exploration. This process can be interpreted as a Markov decision process (MDP), formally defined as a five-tuple $(\mathcal{S}, \mathcal{A}, \mathbb{P}, r, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $r = r(s,a)$ the reward of taking the action a at the state s, and γ the discount factor. The agent's policy $\pi : \mathcal{S} \rightarrow \mathcal{P}(\mathcal{A})$ is a mapping from states to a probability distribution over actions. At time t, the action-value function $Q^{\pi}(s,a) = \mathbb{E}\left[r(s,a) + \gamma \cdot R(s_{t+1}, a_{t+1}) + \dots \;|s_t = s, a_t = a, \pi\right]$. The goal of RL is to find an optimal policy π^* that corresponds to the optimal Q function $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$.

Q-learning [Wat89], an off-policy RL method, is updated by the temporal-difference error (TD-error) δ_{td}

$$\delta_{\text{td}} = (r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{4.1}$$

The Q function can be parameterized as Q_θ with functional approximations using a neural network [MKS15], in which θ can be learned through gradient descent

$$\theta = \theta + \alpha\delta_{\text{td}}\nabla_\theta Q_\theta(s_t, a_t), \tag{4.2}$$

where α is the learning rate.

Policy gradient method [SMS00] is used to optimize a parameterized policy π_λ by the objective function

$$J(\lambda) = \mathbb{E}_{\tau\sim\pi_\lambda}(\tau)[\sum_{t=1}^{T} r(s_t, a_t)], \tag{4.3}$$

where r is the accumulated reward from time step t, and τ is the trajectory of state-action pairs. In order to maximize the objective function, *i.e.*, maximizing the accumulated rewards, gradient ascent can be applied

$$\lambda = \lambda + \eta\nabla_\lambda J(\lambda), \tag{4.4}$$

where η is the learning rate.

The **Boltzmann policy** is a particular type of the soft-max action preference policies, formally defined as

$$\pi(a|s;\theta) = \frac{1}{z}e^{\frac{1}{T}Q(s,a;\theta)}, \tag{4.5}$$

where z is the normalizing constant, T the temperature, and $Q(s, a; \theta)$ the Q function parameterized by θ.

4.2.3.2 Generalized Boltzmann Policy

Theoretically, the standard Boltzmann policy could eventually approach the optimal Q^* as the temperature T decreases. However, the annealing of T usually requires a delicate schedule, often set empirically for different tasks. In HRL, another challenge emerges—the

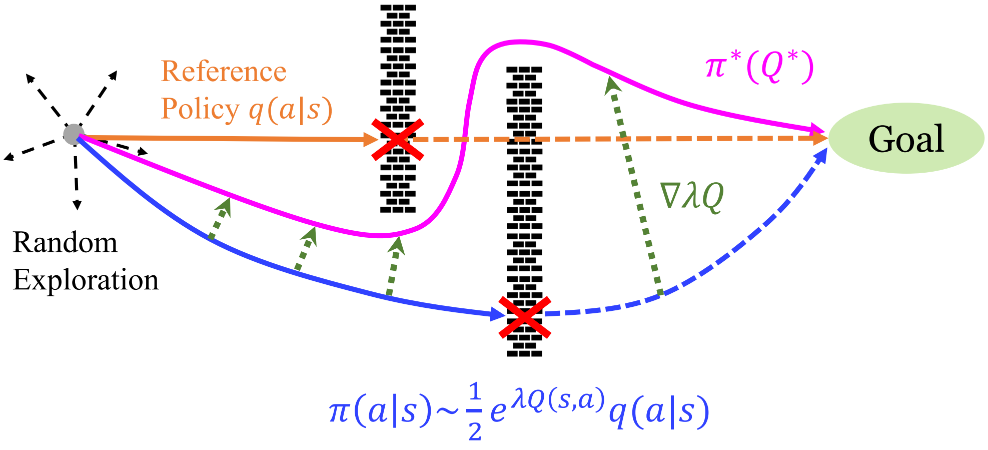


Figure 4.12: Illustration of the generalized Boltzmann policy (GBP). Instead of exploring randomly, the idea is to explore the space deliberately according to auxiliary information of the task (*i.e.*, the reference policy $q(a|s)$). Starting from this stochastic policy, we tilt the reference policy $q(a|s)$ to a more deterministic policy, which gradually converges to Q^*. Preference factor λ with respect to Q adjusts how deterministic the policy is.

Boltzmann policy cannot benefit from the learning of other levels of hierarchy, as it starts from a uniform distribution and explores inefficiently. To overcome these difficulties in the standard Boltzmann policy, we propose the **generalized Boltzmann policy**:

$$\pi(a|s; \lambda, \theta) = \frac{1}{Z} e^{\lambda Q(s,a;\theta)} q(a|s) \propto e^{\lambda Q(s,a;\theta) - U(s,a)}, \tag{4.6}$$

where $q(a|s) \propto e^{-U(s,a)}$ is a reference distribution that can be interpreted from two different views: (i) From the perspective of exploration, $q(a|s)$ can be treated as an initial exploration policy. It can be generated by auxiliary information (*e.g.*, prior knowledge) of the task, which potentially provides a good initial policy for exploration. (ii) We can also view $q(a|s)$ as a hidden cost of each action in a certain state. When $q(a|s)$ is a uniform distribution, the GBP degenerates to the standard Boltzmann policy and random explores at the beginning. In this case, each available action has the exactly the same cost. In contrast, in the proposed GBP, the policy could be optimized from any distribution as an initial exploration, even a pre-trained stochastic policy.

Same as the standard Boltzmann policy, we use $Q(s, a; \theta)$ as the action preference. Unlike the vanilla Boltzmann policy, we use a learnable preference factor λ to replace the temperature term $\frac{1}{T}$. It weights the learned action preference against the initial exploration policy $q(a|s)$. Rather than empirically setting the preference factor, we propose an algorithm to dynamically balance $Q(s, a; \theta)$ and $-U(s, a)$.

4.2.3.3 Learning

We describe the learning algorithm for (i) θ, the parameter of the state-action value function $Q_\theta(s, a)$, and (ii) λ, a learnable preference factor, replacing the pre-scheduled temperature factor $\frac{1}{T}$ in the vanilla Boltzmann policy.

Learning Action Value Function $Q_\theta(s, a)$ The action value function Q_θ is learned using Q-Learning (see Equation 4.1 and Equation 4.2). The action value function is represented by a deep neural network, making it possible to use stochastic gradient descent (SGD) to optimize Q_θ with respect to θ.

Learning Preference Factor λ We propose a learning algorithm for the preference factor λ that is fundamentally different from adjusting the temperature factor $\frac{1}{T}$ in the standard Boltzmann policy. The overall role of these two types of factors are the same: as the policy becomes better, λ should gradually increase to help the policy approach the deterministic optimal policy. Intuitively, we adjust λ by measuring how well the current Q function performs: λ should increase if the current Q function performs well; otherwise decrease. In this way, the algorithm can minimize the negative effect of a poorly learned Q function at an early stage during training, thereby achieving a good balance between the exploitation of the Q function and the exploration guided by the reference policy q, achieved by performing a policy gradient with respect to λ.

With Equation 4.3, the preference factor λ can be adjusted by maximizing the following objective function:

$$J(\lambda) = \mathbb{E}_{\tau \sim \pi_{\lambda,\theta}}(\tau)[\sum_{t=1}^{T} r(s_t, a_t)], \tag{4.7}$$

where $\pi(a|s; \lambda, \theta) = \frac{1}{Z} e^{\lambda Q(s,a;\theta)} q(a|s)$, and τ is the trajectory of state-action pairs.

Then the gradient of the preference factor λ is derived as

$$\nabla_\lambda J(\lambda) = \frac{1}{N} \sum_{i=1}^{N} \left[\left(\sum_{t=1}^{T} \nabla_\lambda \log \pi_{\lambda,\theta}(a_{i,t}|s_{i,t}) \right) \left(\sum_{t=1}^{T} r(s_{i,t}, a_{i,t}) \right) \right], \tag{4.8}$$

where N is the number of trajectories sampled from the current policy. The gradient of the action log-likelihood over preference factor λ can be derived as:

$$\begin{aligned}
&\nabla_\lambda \log \pi_{\lambda,\theta}(a_t|s_t)) = \frac{\partial}{\partial \lambda} \log \frac{1}{Z(\lambda)} e^{\lambda Q_\theta(s_t,a_t;)} q(a_t|s_t) \\
&= Q_\theta(s_t, a_t) - \frac{\partial}{\partial \lambda} \log Z(\lambda) = Q_\theta(s_t, a_t; g) - \frac{1}{Z(\lambda)} \frac{\partial}{\partial \lambda} Z(\lambda) \\
&= Q_\theta(s_t, a_t) - \sum_{a} p_{\lambda,\theta}(a|s_t) Q(s_t, a),
\end{aligned} \tag{4.9}$$

where $Q_\theta(s, a)$ is fixed while updating the preference factor λ, and λ is adjusted by gradient ascent (see Equation 4.4).

As shown in Equation 4.9, the gradient of the preference factor λ in GBP equals to the difference of the current Q value and its expectation with respect to the policy. The interpretation of the Equation 4.9 (see Figure 4.12) matches our intuition—λ should be increased if the current Q value is better than the current policy; otherwise it should be reduced. More concretely, Equation 4.9 steers the direction of the update, and the accumulated rewards along the sampled trajectories determines the volume of adjustment.

Convergence to Deterministic Policy In a proper learning algorithm, $Q(s, a)$ converges to $Q^*(s, a)$ if all state-action pairs are visited during exploration [Wat89]. Therefore, the preference factor λ is guaranteed to make the policy approach a deterministic policy, same as the temperature factor $\frac{1}{T}$ in the vanilla Boltzmann policy. By substituting the Q function in Equation 4.9 with Q^*, we have

$$\nabla_\lambda \log \pi_{\lambda,\theta} = Q^*(s_t, a_t) - \sum_a p_{\lambda,\theta}(a|s_t) Q^*(s_t, a) \geqslant 0. \tag{4.10}$$

Assuming that all the rewards are non-negative, we have $\nabla_\lambda J(\lambda) \geqslant 0$ according to Equation 4.8. In fact, when Q is optimal, the gradient will be strictly positive except that $Q^*(s_t, a_t) = \sum_a p_{\lambda,\theta}(a|s_t) Q^*(s_t, a)$, which indicates $p_{\lambda,\theta}(a|s_t)$ has already coincided with the optimal deterministic policy. Hence, λ will keep increasing, and the policy will converge to the optimal deterministic policy.

Note that although the parameters θ and λ need to be learned in GBP, they are not being updated simultaneously. In fact, they are updated within their own learning interval. As mentioned above, ideally, GBP can converge to a deterministic policy and acts as the same as Q^*. For a smoother learning progress, the update interval of θ should be reasonably longer than the update interval of λ. According to Equation 4.10, a better Q function makes the policy more deterministic. In contrast, an ill-learned Q function could potentially degenerate GBP to the reference policy $q(a|s)$.

4.2.4 HRL with Generalized Boltzmann Policy

4.2.4.1 Setup

Our method follows the setup of [KNS16] in which the sub-tasks are specified. A two-layer hierarchy is constructed: the high-level policy decides what sub-tasks to take and the low-level policies fill in the details of each sub-tasks respectively. The agent will be granted an intrinsic rewards once a low-level policy accomplishes a sub-task. We use a “0-1” intrinsic reward, but it is slightly different from the conventional definition [CBS05, SLB10]: the intrinsic reward is only used for learning the policies for sub-tasks instead of considering it with high-level planning. The high-level policy takes an empirically designed abstraction based on conducted sub-tasks as input (see subsubsection 4.2.5.1) and selects the next sub-task to be accomplished. On the low-level layer, there is a policy bank in where each policy directly takes the observation from the environment as input and outputs the legal actions of the task. A final "0-1" reward will be granted to the high-level policy as soon as it reaches the goal state of the task.

4.2.4.2 Overall Mechanism

Policies represented by GBP on both high- and low-level layers in the hierarchy collaborate together, achieving the final goal. Although both levels of policies use the same representation, the GBP, the underlying mechanism and notation of them are different. The information exchange is implemented via the reference policy in GBP. The learning process for both layers of policies follows subsubsection 4.2.3.3.

Policy on Low-level Layer The form of policies on low-level layers can be represented as

$$\begin{aligned}
&\pi^{\mathrm{low}}_{\lambda,\theta}(a|s^{\mathrm{low}};g) = \frac{1}{Z}e^{\lambda Q_\theta(s^{\mathrm{low}},a))}q^{\mathrm{low}}(a|s^{\mathrm{low}};g), \\
&q^{\mathrm{low}}(a|s^{\mathrm{low}};g)\propto e^{-U(s^{\mathrm{low}},a;g)},
\end{aligned} \tag{4.11}$$

where Z is the normalization constant. As mentioned in subsubsection 4.2.4.1, s^{low} is the feedback from the environment, a the available actions of the task, and $q^{low}(a|s^{low};g)$ the reference policy which guides exploration in low-level space. Usually, it is sub-optimal but better than random policy. Such stochastic policy can be described by a hidden cost $-U(s^{low},a;g)$. Intuitively, the goal of a sub-task g is given by the high-level policy, therefore the corresponding low-level policy should explore related directions towards the sub-goal g instead of picking random actions. Technically, any "hints" can be used to form such reference policy. For example, treating heuristics (used in our experiments) as the hidden cost to generate the reference policy according to Equation 4.11. Additionally, a pre-trained stochastic policy

```
Algorithm 2: High-level Training for Agent Toward $G$
Hyperparameters: Learning rates $\alpha, \eta$ for $\theta, \lambda$, Maximum episode $T$,
Maximum training step $k$, Training interval $\Delta$, $\lambda$ update interval $\delta$,
Recent episode window $\epsilon$
Input: High-level policy $\pi_{(\theta,\lambda)}$ to goal $G$, Memory $D = \varnothing$, Environment $\mathbf{E}$
Result: Learned high-level policy $\pi_{(\theta,\lambda)}$
Init: $i \leftarrow 0$
while $i < k$ do
    $s_t^{high} \leftarrow s_0^{high}$, $s_t^{low} \leftarrow s_0^{low}$, $t \leftarrow 0$
    while $s^{low}$ is NOT terminated and $t \leqslant T$ and $r_G \neq 0$ do
        $R_g = -\sum_{\max(j_g-\epsilon,i)}^{j_g} r_{j_g}$, see $j_g$ in algorithm 3
        $g_t \sim \frac{1}{Z} e^{\lambda Q_\theta(s_t^{\text{high}},g)} q^{\text{high}}(g|s_t^{\text{high}}; R_g)$, see Equation 4.12
        $(s_{t+1}^{low}, s_{t+1}^{high}, r_{g_t}, r_G) \leftarrow$ Execute algorithm 3 with $s_t^{low}, g_t$
        if $r_g \neq 0$ then
            $D \cup (s_t^{high}, g_t, s_{t+1}^{high}), r_G$, $s_t^{high} \leftarrow s_{t+1}^{high}$
        end
        $s_t^{low} \leftarrow s_{t+1}^{low}$
        if $i = \Delta$ then
            if $i = \delta$ then
                update $\lambda$ with Equation 4.4 and Equation 4.9
            end
            else
                update $\theta$ with Equation 4.1 and Equation 4.2
            end
        end
    end
end
```

can serve as the initial exploration. Note that the reference policies on the low-level layer are designed with a distance-related heuristics. An automatic pipeline for converting g into a reference policy is desirable. Eventually, the low-level policies are expected to learn the optimal policy for accomplishing the sub-tasks.

Policy on High-level Layer Taking different representation as the input, the high-level policy selects a proper sub-task for corresponding low-level policy to accomplish. We formulate it in the form of GBP:

$$\begin{aligned}\pi^{\text{high}}_{\lambda,\theta}(g|s^{\text{high}};R^{\text{low}}) = \frac{1}{Z}e^{\lambda Q_\theta(s^{\text{high}},g)q(g|s^{\text{high}};R^{\text{low}})},\\ q(g|s^{\text{high}};R^{\text{low}})\propto e^{-U(s^{\text{high}},g;R^{\text{low}})},\end{aligned} \tag{4.12}$$

where Z is the normalization constant. The input of high-level policy is an abstraction of a state reported by the environment. The state abstraction contains the information of visited sub-goals (see more detail in subsubsection 4.2.5.1). Ideally, such abstraction should be nominated automatically. In this work, we manually design the form of state abstraction. The reference policy $q^{high}(g|s^{high};R^{low})$ of the high-level policy is serving for the same purpose, *i.e.*, narrowing down the exploration space, but with different interpretation and implementation. The hidden cost $-U(s^{high},g;R^{low})$ can be given by the average intrinsic rewards achieved in a recent time window by the corresponding low-level policies. In short, the more likely one low-level policy can complete a sub-task, the less probable for a high-level policy to select this sub-task in a certain abstraction state.

The idea behind it is that we hope the high-level policy to explore its space with the unskilled sub-tasks policies. Such sub-tasks may be critical to the entire task, therefore the agent is required to master it. If these sub-tasks are parts of the optimal path to the goal state, the agent will acquire the final reward. Such experiences help to learn $Q_\theta(s^{high},g)$ in the high-level policy. As a result, the better the Q function is, the more deterministic (λ increases) the policy will be according to subsubsection 4.2.3.3. In other words, the high-level policy learns to conduct correct sub-tasks sequence to achieve the final reward.

4.2.4.3 Algorithm

The algorithm of the proposed method is described in algorithm 2 and algorithm 3. Specifically, both levels of policies are represented by GBP, learned using the method described in subsubsection 4.2.3.3. The *episode* denotes the maximum number of steps that is allowed in

```
Algorithm 3: Low-level Training for Agent Towards $g$
  Hyperparameters: Learning rates $\alpha, \eta$ for $\theta_g, \lambda_g$, Maximum episode $T$,
  Maximum training step $k$, Training interval $\Delta$, $\lambda$ update interval $\delta$
  Input: Low-level policy $\pi_{(\theta_g,\lambda_g)}$ to subgoal $g$, current low-level state $s_{0_g}^{low}$,
  Memory $D_g = \varnothing$, Environment $\mathbf{E}$
  Result: Learned low-level policy $\pi_{(\theta_g,\lambda_g)}$
  Init: $j_g \leftarrow 0$
  while $j_g < k$ do
      $s_t^{low} \leftarrow s_{0_g}^{low}$, $t \leftarrow 0$, $r_g \leftarrow 0$, $r_G \leftarrow 0$
      while $s^{low}$ is NOT terminated and $t \leqslant T$ and $r_g, r_G \neq 0$ do
          $a_t \sim \frac{1}{Z} e^{\lambda_g Q_{\theta_g}(s_t^{low}, a; g)} q^{low}(a|s_t; g)$
          $(s_{t+1}^{low}, s_{t+1}^{high}, r_g, r_G, g) \leftarrow \mathbf{E}(s_t^{low}, s_t^{high}, a_t)$
          $D_g \cup \{(s_t^{low}, a_t, s_{t+1}^{low}, r_g)\}$
          $s_t^{low} \leftarrow s_{t+1}^{low}$, $s_t^{high} \leftarrow s_{t+1}^{high}$, $t \leftarrow t + 1$
          if $j_g = \Delta$ then
              if $j_g = \delta$ then
                  update $\lambda_g$ with Equation 4.4 and Equation 4.9
              end
              else
                  update $\theta_g$ with Equation 4.1 and Equation 4.2
              end
          end
      end
      $j_g \leftarrow j_g + 1$
      yield $(s_t^{low}, s_t^{high}, r_g, r_G)$, see Python for yield
  end
```

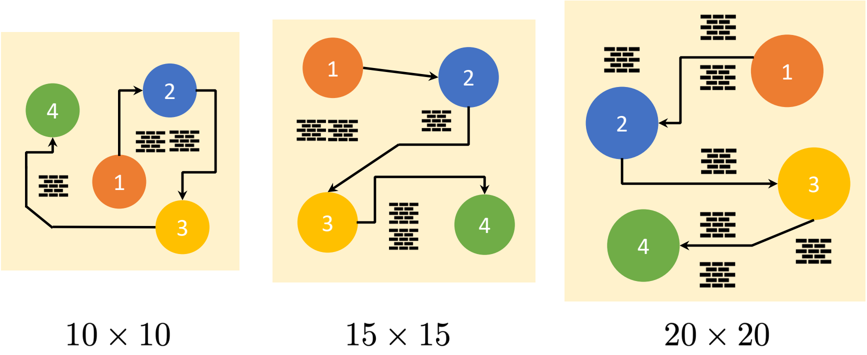
a single trail. *termination* is defined as a terminal non-goal state. Note that the *episode*, α, η on high- and low-level are different from each other. For simplicity, they are sharing the same symbols. Additionally, the λ-update period and Q-update period are not the same in high- and low-level policies' training.

4.2.5 Experiments

4.2.5.1 Experimental Setup

Environment We evaluate our method on a 2D grid world game in various scales. An illustrative example of this kind of game is shown in Figure 4.13. In order to achieve the final reward, the agent needs to conduct a sequence of sub-tasks in a correct order. For each environment, there is a unique path to obtain the final goal. Another challenge of this type of task is that the agent is not only asked to accomplish the task on symbolic level but also fulfill the details of the sub-task. In summary, the agent is required to go through certain landmarks in the maps with correct orders.

Game Setting In our setting, we manually define a certain number of sub-goals and the unique order of them to the final reward. The environment returns a state (**position**) as an input for the agent. The agent takes two different inputs. For high-level policy, the input is an abstraction state which is an **one-hot** vector, indicating the most recently and correctly achieved sub-goal. We assume that the high-level state transition will only take place when the agent correctly discovers the first remaining sub-goal. To level up the difficulty of the game, we bring in different types of obstacles and traps that the agent might need to jump on, or avoid to encounter (see supplementary for further descriptions). The size of valid actions set of our environment is as the same as the one in OpenAI Gym [BCP16]. Following the conventional RL setup, terminal conditions are set to be (i) the agent triggers the traps, or (ii) it exceeds the maximum steps limits for a single trail. Two types of rewards will be granted: (i) a “0-1” intrinsic reward for each low-level who accomplishes a sub-task; (ii) a “0-1” final reward can be achieved by the high-level policy if it successfully produces the correct sequence of sub-tasks.

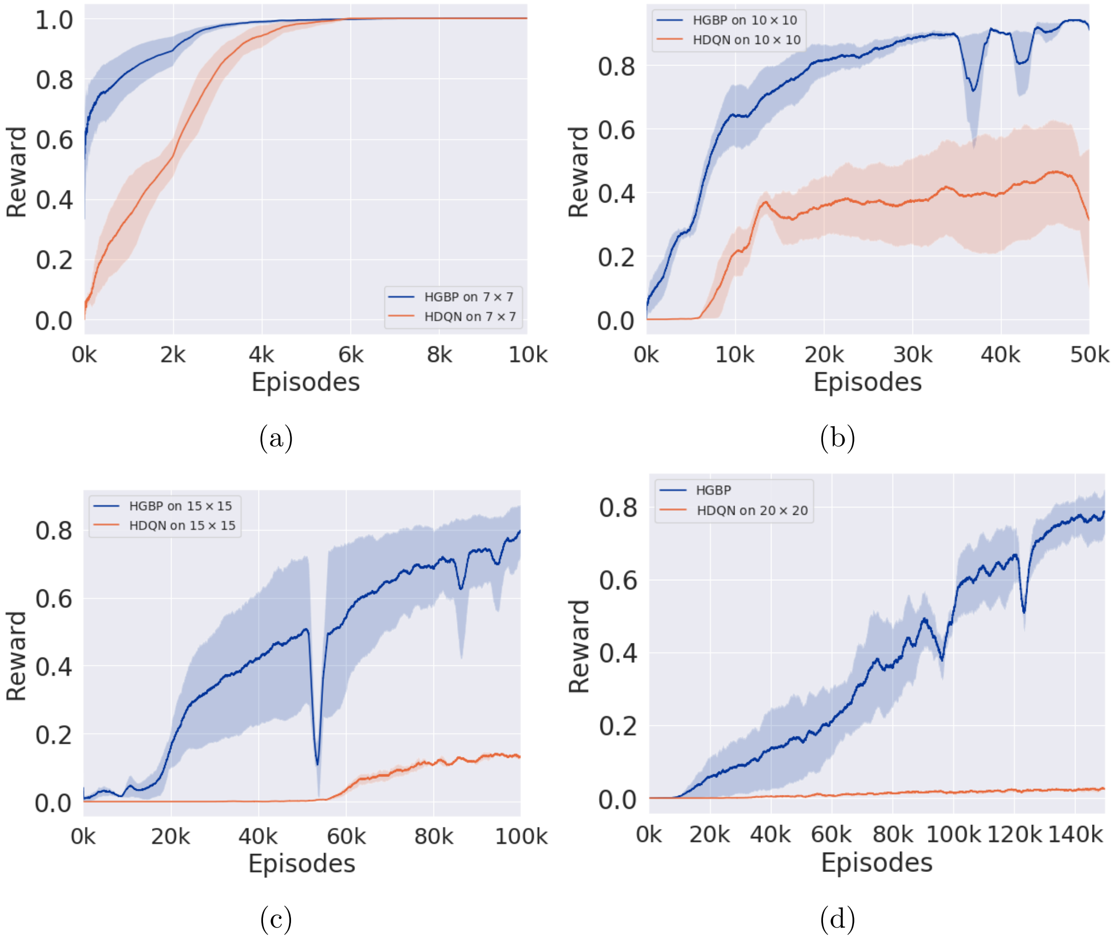


10×10 15×15 20×20

Figure 4.13: A simplified sketch of the tasks in different scales of grid-worlds: 7×7, 10×10, 15×15, 20×20, respectively. We conduct the experiments on these 4 environments where the agent needs to both achieve the final reward with correct order and accomplishing each sub-task.

4.2.5.2 Implementation Details

In our experiments, the state-action value function Q in GBP is implemented as a feed-forward neural network with ReLU as a nonlinear activation. For high-level policy, the input is the abstract one-hot state vector defined in section 4.2.5.1, and the output is the next sub-goal. For low-level policies, the input is the current position, and the output is the next action. The details of the network configurations are described in supplementary materials. We set the learning rate as 2×10^{-4} for learning high-level policies, and 6.25×10^{-5} for learning low-level policies. The discount factor γ for the low- and high-level learning are 0.95 and 0.5, respectively. For the high-level GBP learning, we update $Q(s^{\text{high},g})$ for 90 episodes, followed by the λ for 10 episodes. For the low-level GBP learning, we update $Q_g(s^{\text{low}})$ for 990 episodes, followed by the λ_g update for 10 episodes. The hidden cost $U(s^{\text{high}}, g)$ in the high-level policy is implemented as the average count of successful trials in finishing low-level policies in the most recent 200 episodes. The reference policy for low-level policies follows distance-related heuristics, for example the L2-distance.



(a)

(b)

(c)

(d)

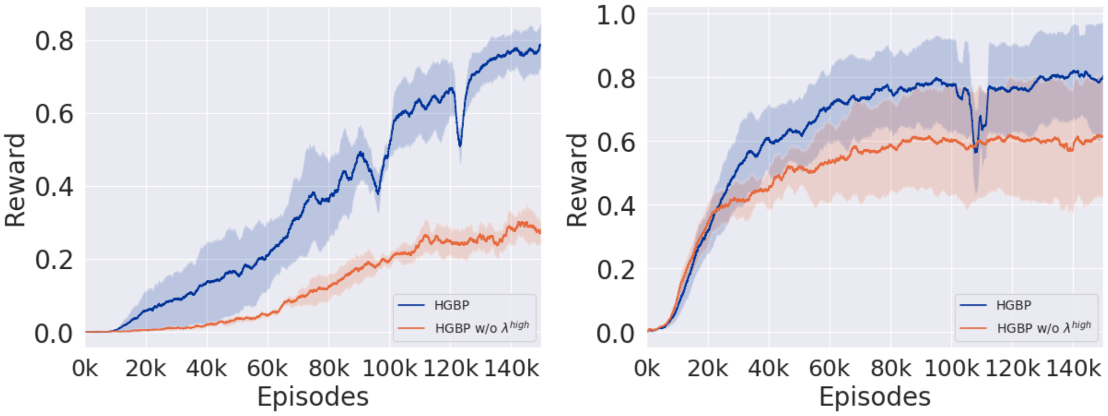
Figure 4.14: **The performance of the proposed model and the baselines** under different scales and difficulties of tasks. (a) 7 × 7 grid-world, (b) 10 × 10 grid-world, and (c) 15 × 15 grid-world, (d) 20 × 20 grid-world.

4.2.5.3 Empirical Results and Analysis

We evaluate the proposed method in 4 different 2D grid-world environments. As shown in Figure 4.13, scaling up the tasks increases the difficulty of them. Therefore, we can exam the exploration efficiency of our method on them. Besides, we compare the performance of our method with a popular HRL framework [KNS16] the same set of tasks. To further analyze our method, especially GBP, we provide ablation studies below.

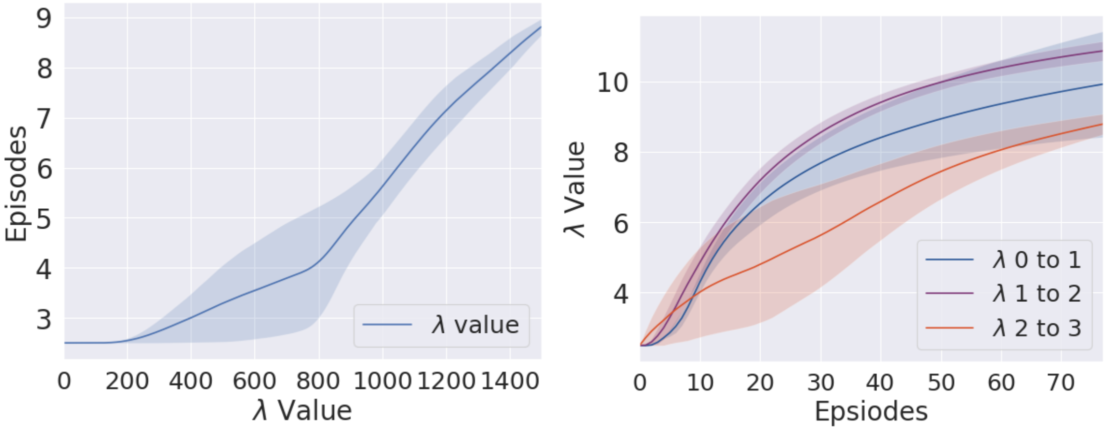
Performance Evaluation and Analysis The results for the proposed method are shown in Figure 4.14. We implement [KNS16] to be the baseline for comparison. The experimental results show that the proposed method has significant performance advantages in difficult environments. In the simplest task, the baseline framework achieves similar performance comparing to our method. However, as the tasks scale up, the proposed method outperforms the baseline by a larger and larger margin. Specifically, the reasons are two-fold: (i) the fast convergence of low-level learning, and (ii) the fast learning of the high-level policy. Fundamentally, the exploration across the hierarchy in our method is much more efficient than the baseline framework. (i) The reference policies on low-level layer in the hierarchy help the low-level policies to explore towards the sub-goal, therefore it collects experiences with rewards more effectively. In difficult tasks, such an initial exploration strategy can achieve the sparse reward with much higher probability. In contrast, the low-level policies in the baseline method can hardly reach the sub-goal states. (ii) The preference factor λ and the reference policy of the high-level policy encourage to learn unskilled but important sub-tasks and roll out sub-tasks that are not critical to the task. Hence, according to the results shown in Figure 4.15, the high level policy can learn the optimal path faster. For h-DQN model, it can be extremely difficult to propagate the value in such a sparse reward setup if the scale of the task is too large.

Ablation Study of Preference Factor λ In addition, Figure 4.16 shows the trend of the preference factor λ in the adopted GBP. In subsection 4.2.3, we introduce a preference factor λ and discuss the learning algorithm of it subsubsection 4.2.3.3. The results shown in Figure 4.16 affirm our hypothesis that the preference should increase over the learning progress, making the policy deterministic. The gradually smoothed curve of λ also empirically implies Equation 4.10. The gradient of λ decreases when the policies are gradually approaching optimal, slowing down the trend to increase λ.



(a) (b)

Figure 4.15: **The performance of the proposed method with or without** λ in GBP of the high-level policy. Results are shown in two (a),(b) tasks in different scales and difficulties.



(a) (b)

Figure 4.16: **Trend of preference factor** λ in both high- and low-level policies in the proposed method. (a) shows the trend of λ located on the high-level layer. (b) reflects the trend of 3 λ on the low-level policies for a certain sub-task.

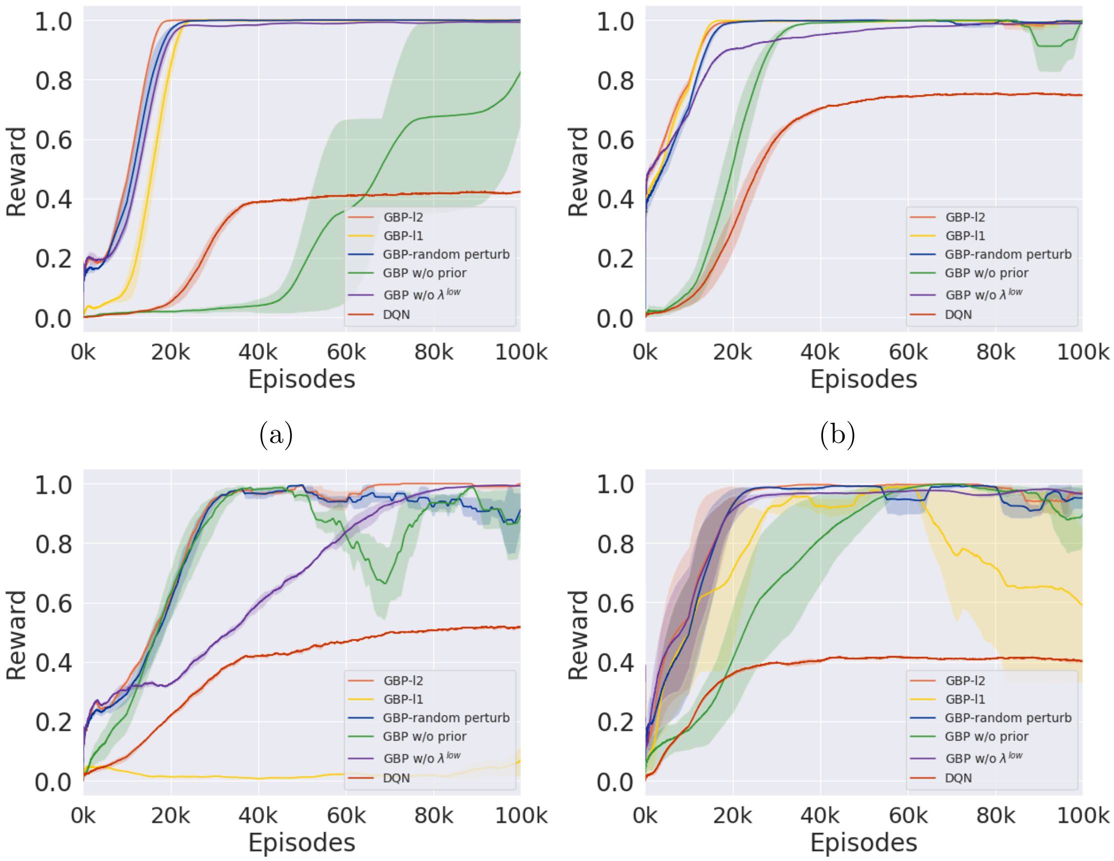
Ablation Studies of Different Reference Policy In the adopted GBP model, $U(s, a)$ is defined as the hidden cost for taking a specific action a in state s. In experiments, we implement the hidden cost $U(s^{low}, a; g)$ of the low-level policies with distance-related heuristics between the state s^{low} and the sub-goal location g, *e.g.*, L2-distance. Here, we discuss the performance of the low-level policies based on the experimental results using different heuristics to construct the reference policies. Four different heuristics are compared for constructing the reference policy. (i) Euclidean distance (L2-distance) (ii) Manhattan distance (L1-distance) (iii) L2-distance with random shuffle (iv) random exploration, *i.e.* uniform reference distribution. We also compare the above models with the ablation model which disables the learning of preference factor λ. For euclidean distance with random shuffle, the reference policy does not always reflect the heuristic. The reference policy will randomly pick an available action at time t according to Equation 4.13; the probability is set to 0.2. We also compare these heuristics and the ϵ-*greedy* exploration.

$$q(a|s) = \begin{cases} q(a|s) & \text{if } 1-p \\ \text{random action} & \text{if } p \end{cases} \tag{4.13}$$

As shown in Figure 4.17, in most of the cases, GBP with different kinds of heuristics shares similar performances. Comparing the ϵ-greedy strategy, GBP demonstrates its exploration efficiency, even with random exploration. One thing worth noticing is in Figure 4.17c, L1-distance seems not to perform well as in other cases. We argue that this is caused by the aggressiveness of L1-distance measure compared to L2-distance. We design some obstacles and traps on purpose to prevent reference policy from getting to the goal directly. Such setting can lead to aggressive exploration, being trapped in a dilemma. Note that the preference factor λ in the GBP encourages the policy to be deterministic, making faster convergence. We argue that this experiment justifies the robustness of the adopted GBP when potentially misleading reference policy is provided.

4.2.6 Conclusion

The experiments on various setup of tasks justify the effectiveness and efficiency of the proposed integration of GBP and HRL comparing to the vanilla framework. The empirical results show significant improvement comparing to a popular HRL framework. In conclusion,



(a)

(b)

(c)

(d)

Figure 4.17: **The performance of different heuristics for low-level policies** in the 20×20 grid-world. In most cases, GBP with heuristics significantly outperforms the normal DQN with ϵ-*greedy* as exploration strategy.

GBP has shown promising results to improve the HRL framework.

CHAPTER 5

Conclusion

In this dissertation, we introduce our contributions to developing early-stage techniques for an intelligent robot by exploring three domains: robot imitation learning, visual abstraction, and visual language reasoning, and robot planning.

In chapter 2, we study robot imitation learning by demonstrating a series of works. We proposed a prototype of a tactile glove for collecting human demonstration data. The tactile glove expands the spectrum from vision-only demonstration to multi-modal demonstration, including human pose and force sensing. We develop an unsupervised segmentation method to build hierarchical clusters of the collected human demonstration data. In order to learn from human demonstration, we proposed a method for imitation learning. It combines symbolic and neural network representations. The symbolic-based grammar planner can help to achieve better long-term planning, while the NN-based planner is more accurate in action prediction with tactile feedback. By using GEP, we fuse these two planners to achieve reasonable performance. We prove that the proposed imitation learning method successfully enables a humanoid robot to learn from the human demonstration in medicine bottle opening tasks. In addition, we investigate enhancing human's trust in robots' behavior. We conduct human studies and successfully show that the proposed imitation learning method gains human trust well.

We study visual reasoning in two different tasks in chapter 3. The first task, the human IQ test, namely RPM is a challenging task. In order to build a computer system that can tackle this task, we first propose a method that can systematically generate RPM. We also benchmark the generated RPM dataset, RAVEN. Furthermore, we introduce a DL-based method with a contrasting mechanism to solve the RPM. We show that our method successfully solves the RAVEN dataset, even with a relatively small amount of data. On the other hand, we also study the visual language reasoning problem. We choose to approach one of the most difficult tasks, outside knowledge visual question answering. We propose a method that transforms multi-modal domain information into a natural language domain for later reasoning and question-answering. This method successfully outperformed all SOTAs by the time it was made public.

In chapter 4, we introduce two attempts for a better robot planning system. We first develop a virtual testbed for robot agents. This environment is physically realistic. Additionally, robots can interact with virtual human demonstrators. Given these two advantages, our virtual platform is able to provide a more realistic robot training environment so that we can keep closing the gap between a simulator and the real world. We also demonstrate the idea of using RL-based methods for robot navigation tasks. This method makes use of heuristics and integrates it with hierarchical RL, showing promising performances.

To this end, we want to discuss some of the future research directions that are inspired by this dissertation:

Better Robot Imitation Learning: Current robot imitation learning methods heavily rely on cleaned data and environment. Most of them are also data-hungry. To relieve, we may want to apply techniques such as unsupervised learning or self-supervised learning with better representations. In addition, few-shot learning, even one-shot or zero-shot learning, could be the trend. As mentioned, we want the robot to learn with fewer examples. Therefore, few-shot techniques could help. In order to achieve it, we may need to apply large models pre-trained with large amount of knowledge. one could possibly build an imitation learning model that is able to quickly adapt to new tasks without forgetting learned knowledge.

More Efficient Multi-Modal Reasoning: The human IQ test actually inspires us to use smaller data to learn a relatively small model. However, it is still too restricted and simple compared to real-world scenarios. Tasks such as outside knowledge visual question answering require much more knowledge. Therefore, current methods all use huge models to tackle it. Unfortunately, the performance is yet unsatisfied, and using larger models is not sustainable. We argue that we should build a model for better efficiency of reasoning. It is possible to build a model that can think with an abstract representation. Just like how human does in IQ test, we just need a small amount of data to solve big tasks.

Author Contribution This dissertation is a combination of work that the author, Feng Gao, led and participated in during his Ph.D. study.

chapter 2 complies with four published papers in which Feng Gao is one of the authors.

- In [LXM17], Feng Gao contributes to the human demonstration data collection, experiments, and paper writing.
- In [XLE18], Feng Gao contributes to experiments and paper writing.
- In [EGX17], Feng Gao is the joint first author who takes the lead in key ideas, modeling, coding, robot platform development, experiments, and paper writing.
- In [EGL19], Feng Gao is the joint first author, and he is the student leader in modeling, coding, experiments, and robot platform development. He also participates in running human subject studies and paper writing.

chapter 3 complies with three published papers in which Feng Gao is one of the authors.

- In [ZGJ19], Feng Gao is the joint first author who contributes to the key idea, coding, experiment, and paper writing.
- In [ZJG19], Feng Gao contributes to experiments, evaluation, and paper writing.

- In [GPT22], Feng Gao is the first author who is in charge of key ideas, modeling, coding, experiments, and paper writing.

chapter 4 complies with one published paper and one pre-printed manuscript in which Feng Gao is one of the authors or main contributors.

- In [XLZ19], Feng Gao contributes to the coding, experiment, and paper writing.
- In section 4.2, Feng Gao is the main contributor who is in charge of key ideas, modeling, coding, experiments, and paper writing.

REFERENCES

[AAL15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. "Vqa: Visual question answering." In *International Conference on Computer Vision (ICCV)*, 2015.

[ACV09] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. "A Survey of Robot Learning from Demonstration." *Robotics and autonomous systems*, **57**(5):469–483, 2009.

[AHB18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. "Bottom-up and top-down attention for image captioning and visual question answering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.

[AHL07] Ngo Anh Vien, Nguyen Hoang Viet, SeungGwan Lee, and TaeChoong Chung. "Heuristic Search Based Exploration in Reinforcement Learning." In *Computational and Ambient Intelligence*, 2007.

[AKL17] Jacob Andreas, Dan Klein, and Sergey Levine. "Modular Multitask Reinforcement Learning with Policy Sketches." In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[Ams62] Abram Amsel. "Frustrative nonreward in partial reinforcement and discrimination learning: Some recent history and a theoretical extension." *Psychological review*, **69**(4):306, 1962.

[AN04] Pieter Abbeel and Andrew Y Ng. "Apprenticeship learning via inverse reinforcement learning." In *Proceedings of the twenty-first international conference on Machine learning (ICML)*, p. 1, 2004.

[Arn69] Rudolf Arnheim. *Visual thinking.* Univ of California Press, 1969.

[AYB18] Somak Aditya, Yezhou Yang, and Chitta Baral. "Explicit Reasoning over End-to-End Neural Architectures for Visual Question Answering." *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[BAC20] Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. "Genericskb: A knowledge base of generic statements." *arXiv preprint arXiv:2005.00660*, 2020.

[BBA16] Edoardo Battaglia, Matteo Bianchi, Alessandro Altobelli, Giorgio Grioli, Manuel G Catalano, Alessandro Serio, Marco Santello, and Antonio Bicchi. "ThimbleSense: a fingertip-wearable tactile sensor for grasp analysis." *IEEE Transactions on Haptics*, **9**(1):121–133, 2016.

[BCP16] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym.", 2016.

[BF08] Marcus A Brubaker and David J Fleet. "The kneed walker for human pose tracking." In *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR*, pp. 1–8. IEEE, 2008.

[BFH10] Marcus A Brubaker, David J Fleet, and Aaron Hertzmann. "Physics-based person tracking using the anthropomorphic walker." *International journal of computer vision (IJCV)*, **87**(1):140–155, 2010.

[BHP17] Pierre-Luc Bacon, Jean Harb, and Doina Precup. "The Option-Critic Architecture." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[BHS18] David Barrett, Felix Hill, Adam Santoro, Ari Morcos, and Timothy Lillicrap. "Measuring abstract reasoning in neural networks." In *Proceedings of International Conference on Machine Learning (ICML)*, pp. 511–520, 2018.

[BMR20] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners." *arXiv preprint arXiv:2005.14165*, 2020.

[Bow61] Gordon H Bower. "A contrast effect in differential conditioning." *Journal of Experimental Psychology*, **62**(2):196, 1961.

[BRM12] JP Bandera, JA Rodriguez, L Molina-Tanco, and A Bandera. "A survey of vision-based architectures for robot learning by imitation." *International Journal of Humanoid Robotics*, **9**(01):1250006, 2012.

[BS18] Noam Brown and Tuomas Sandholm. "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals." *Science*, **359**(6374):418–424, 2018.

[BSC18] Yonatan Bisk, Kevin J Shih, Yejin Choi, and Daniel Marcu. "Learning Interpretable Spatial Operations in a Rich 3D Blocks World." *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[BSF09] Marcus A Brubaker, Leonid Sigal, and David J Fleet. "Estimating contact dynamics." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2389–2396. IEEE, 2009.

[BSO16] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. "Unifying count-based exploration and intrinsic motivation." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[CBS05] Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. "Intrinsically motivated reinforcement learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2005.

[CDF17] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. "Matterport3D: Learning from RGB-D Data in Indoor Environments." In *International Conference on 3D Vision (3DV)*, 2017.

[CF98] Cristiano Castelfranchi and Rino Falcone. "Principles of trust for MAS: Cognitive anatomy, social importance, and quantification." In *Proceedings International Conference on Multi Agent Systems*, 1998.

[CFG15] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. "Shapenet: An information-rich 3d model repository.", 2015.

[CFH92] David J Chalmers, Robert M French, and Douglas R Hofstadter. "High-level perception, representation, and analogy: A critique of artificial intelligence methodology." *Journal of Experimental & Theoretical Artificial Intelligence*, **4**(3):185–211, 1992.

[CFW17] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. "Reading wikipedia to answer open-domain questions." *arXiv preprint arXiv:1704.00051*, 2017.

[CG17] Christopher Clark and Matt Gardner. "Simple and effective multi-paragraph reading comprehension." *arXiv preprint arXiv:1710.10723*, 2017.

[CH89] Richard Catrambone and Keith J Holyoak. "Overcoming contextual limitations on problem-solving transfer." *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **15**(6):1147, 1989.

[CHC16] Inrak Choi, Elliot W Hawkes, David L Christensen, Christopher J Ploch, and Sean Follmer. "Wolverine: A wearable haptic interface for grasping in virtual reality." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 986–993. IEEE, 2016.

[CHL05] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. "Learning a similarity metric discriminatively, with application to face verification." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[CJS90] Patricia A Carpenter, Marcel A Just, and Peter Shell. "What one intelligence test measures: a theoretical account of the processing in the Raven Progressive Matrices Test." *Psychological review*, **97**(3):404, 1990.

[CKP14] Yevgen Chebotar, Oliver Kroemer, and Jan Peters. “Learning Robot Tactile Sensing for Object Manipulation.” In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3368–3375. IEEE, 2014.

[CLL18] Qingxing Cao, Xiaodan Liang, Bailing Li, Guanbin Li, and Liang Lin. “Visual Question Reasoning on General Dependency Tree.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[CLY19] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. “Uniter: Learning universal image-text representations.” *arXiv preprint arXiv:1909.11740*, 2019.

[CM09] Kate Crookes and Elinor McKone. “Early maturity of face recognition: No childhood development of holistic processing, novel face encoding, or face-space.” *Cognition*, **111**(2):219–247, 2009.

[CPB06] Christina L Campbell, Richard Alan Peters, Robert E Bodenheimer, William J Bluethmann, Eric Huber, and Robert O Ambrose. “Superpositioning of behaviors learned through teleoperation.” *T-RO*, **22**(1):79–91, 2006.

[CR68] Fergus W Campbell and JG Robson. “Application of Fourier analysis to the visibility of gratings.” *The Journal of physiology*, **197**(3):551–566, 1968.

[CSS10] Catherine C Chase, Jonathan T Shemwell, and Daniel L Schwartz. “Explaining across contrasting cases for deep understanding in science: An example using interactive simulations.” In *Proceedings of the 9th International Conference of the Learning Sciences*, 2010.

[CWS15] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. “Benchmarking in Manipulation Research.” *IEEE Robotics & Automation Magazine*, **1070**(9932/15):36, 2015.

[DAZ17] Abhishek Das, Harsh Agrawal, Larry Zitnick, Devi Parikh, and Dhruv Batra. “Human attention in visual question answering: Do humans and deep networks look at the same regions?” *Computer Vision and Image Understanding*, **163**:90–100, 2017.

[DCH16] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. “Benchmarking deep reinforcement learning for continuous control.” In *International Conference on Machine Learning (ICML)*, 2016.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv:1810.04805*, 2018.

[DDS09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database." In *CVPR*, 2009.

[DH93] Peter Dayan and Geoffrey E Hinton. "Feudal reinforcement learning." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 1993.

[DH02] Y Derimis and G Hayes. "Imitations as a dual-route process featuring predictive and learning components: a biologically plausible computational model." *Imitation in animals and artifacts*, pp. 327–361, 2002.

[Die00] Thomas G Dietterich. "Hierarchical reinforcement learning with the MAXQ value function decomposition." *Journal of Artificial Intelligence Research*, **13**:227–303, 2000.

[DL17] Bo Dai and Dahua Lin. "Contrastive learning for image captioning." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[Dom15] Pedro Domingos. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.

[DSD08] Laura Dipietro, Angelo M Sabatini, and Paolo Dario. "A survey of glove-based systems and their applications." *Ieee transactions on systems, man, and cybernetics, part c (applications and reviews)*, **38**(4):461–482, 2008.

[Ear70] Jay Earley. "An efficient context-free parsing algorithm." *Communications of the ACM*, **13**(2):94–102, 1970.

[EGL19] Mark Edmonds, Feng Gao, Hangxin Liu, Xu Xie, Siyuan Qi, Brandon Rothrock, Yixin Zhu, Ying Nian Wu, Hongjing Lu, and Song-Chun Zhu. "A tale of two explanations: Enhancing human trust by explaining robot behavior." *Science Robotics*, **4**(37):eaay4663, 2019.

[EGX17] Mark Edmonds, Feng Gao, Xu Xie, Hangxin Liu, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. "Feeling the force: Integrating force and pose for fluent discovery through imitation learning to open medicine bottles." In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3530–3537. IEEE, 2017.

[EHN96] Kutluhan Erol, James Hendler, and Dana S Nau. "Complexity results for HTN planning." *Annals of Mathematics and Artificial Intelligence*, **18**(1):69–93, 1996.

[EKM84] R E Snow, Patrick Kyllonen, and B Marshalek. "The topography of ability and learning correlations." *Advances in the psychology of human intelligence*, pp. 47–103, 1984.

[Eva62] TG Evans. *A Heuristic Program to Solve Geometric Analogy Problems.* PhD thesis, MIT, 1962.

[Eva64] Thomas G Evans. "A heuristic program to solve geometric-analogy problems." In *Proceedings of the April 21-23, 1964, spring joint computer conference*, 1964.

[Fal19] Andrea Falcon. "Aristotle on Causality." In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Metaphysics Research Lab, Stanford University, spring 2019 edition, 2019.

[FCL20] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. "Scalable multi-hop relational reasoning for knowledge-aware question answering." *arXiv preprint arXiv:2005.00646*, 2020.

[FGM10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. "Object detection with discriminatively trained part-based models." *IEEE transactions on pattern analysis and machine intelligence*, **32**(9):1627–1645, 2010.

[FHL16] Lu Feng, Laura Humphrey, Insup Lee, and Ufuk Topcu. "Human-interpretable diagnostic information for robotic planning systems." In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[FJ13] Mohsen Falahi and Masoumeh Jannatifar. "Using orthogonal basis functions and template matching to learn whiteboard cleaning task by imitation." In *ICCKE.* IEEE, 2013.

[FN71] Richard E Fikes and Nils J Nilsson. "STRIPS: A new approach to the application of theorem proving to problem solving." *Artificial intelligence*, **2**(3-4):189–208, 1971.

[FOR18] Aleksandra Faust, Kenneth Oslund, Oscar Ramirez, Anthony Francis, Lydia Tapia, Marek Fiser, and James Davidson. "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5113–5120. IEEE, 2018.

[Fu74] King Sun Fu. *Syntactic methods in pattern recognition*, volume 112. Elsevier, 1974.

[Gal83] Francis Galton. *Inquiries into human faculty and its development.* Macmillan, 1883.

[GCL20] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. "Large-scale adversarial training for vision-and-language representation learning." *arXiv preprint arXiv:2006.06195*, 2020.

[Gen83] Dedre Gentner. "Structure-mapping: A theoretical framework for analogy." *Cognitive science*, **7**(2):155–170, 1983.

[GG55] James J Gibson and Eleanor J Gibson. "Perceptual learning: Differentiation or enrichment?" *Psychological review*, **62**(1):32, 1955.

[GG01] Dedre Gentner and Virginia Gunn. "Structural alignment facilitates the noticing of differences." *Memory & Cognition*, **29**(4):565–577, 2001.

[GH10] Michael Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[Gib14] James J Gibson. *The ecological approach to visual perception: classic edition.* Psychology Press, 2014.

[GKP03] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. "Efficient solution algorithms for factored MDPs." *Journal of Artificial Intelligence Research*, **19**:399–468, 2003.

[GM94] Dedre Gentner and Arthur B Markman. "Structural alignment in comparison: No difference without similarity." *Psychological science*, **5**(3):152–158, 1994.

[GMK99] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: Minds, brains, and how children learn.* William Morrow & Co, 1999.

[GP92] Mary L Gick and Katherine Paterson. "Do contrasting examples facilitate schema acquisition and analogical transfer?" *Canadian Journal of Psychology/Revue canadienne de psychologie*, **46**(4):539, 1992.

[GPT22] Feng Gao, Qing Ping, Govind Thattai, Aishwarya Reganti, Ying Nian Wu, and Prem Natarajan. "A Thousand Words Are Worth More Than a Picture: Natural Language-Centric Outside-Knowledge Visual Question Answering." *arXiv preprint arXiv:2201.05299*, 2022.

[Gra06] Temple Grandin. *Thinking in pictures: And other reports from my life with autism.* Vintage, 2006.

[GSL15] Ye Gu, Weihua Sheng, Meiqin Liu, and Yongsheng Ou. "Fine manipulative action recognition through sensor fusion." In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 886–891. IEEE, 2015.

[Gun17] David Gunning. "Explainable artificial intelligence (XAI)." *Defense Advanced Research Projects Agency (DARPA)*, 2017.

[GWH21] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. "KAT: A Knowledge Augmented Transformer for Vision-and-Language." *arXiv preprint arXiv:2112.08614*, 2021.

[GXT21] Dalu Guo, Chang Xu, and Dacheng Tao. "Bilinear graph networks for visual question answering." *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[GZA20] François Garderes, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. "Conceptbert: Concept-aware representation for visual question answering." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pp. 489–498, 2020.

[GZW07] Cheng-en Guo, Song-Chun Zhu, and Ying Nian Wu. "Primal sketch: Integrating structure and texture." *Computer Vision and Image Understanding (CVIU)*, **106**(1):5–19, 2007.

[HAD18] Ronghang Hu, Jacob Andreas, Trevor Darrell, and Kate Saenko. "Explainable neural computation via stack neural module networks." In *European Conference on Computer Vision (ECCV)*, 2018.

[HAR17] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. "Learning to Reason: End-to-End Module Networks for Visual Question Answering." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[HD94] Gillian M Hayes and John Demiris. *A robot controller using learning by imitation.* University of Edinburgh, Department of Artificial Intelligence, 1994.

[HDW09] Rubi Hammer, Gil Diesendruck, Daphna Weinshall, and Shaul Hochstein. "The development of category learning strategies: What makes the difference?" *Cognition*, **112**(1):105–119, 2009.

[HHT96] Keith J Holyoak, Keith James Holyoak, and Paul Thagard. *Mental leaps: Analogy in creative thought.* MIT press, 1996.

[HIO11] Etsuko Haryu, Mutsumi Imai, and Hiroyuki Okada. "Object similarity bootstraps young children to action-based verb extension." *Child Development*, **82**(2):674–686, 2011.

[HLD16] Bidan Huang, Miao Li, Ravin Luis De Souza, Joanna J Bryson, and Aude Billard. "A Modular Approach to Learning Manipulation Strategies from Human Demonstration." *Autonomous Robots*, **40**(5):903–927, 2016.

[HM18] Drew A Hudson and Christopher D Manning. "Compositional attention networks for machine reasoning." *arXiv preprint arXiv:1803.03067*, 2018.

[HM19] Drew A Hudson and Christopher D Manning. “Gqa: A new dataset for real-world visual reasoning and compositional question answering.” In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

[HMW14] Frank L Hammond, Yiğit Mengüç, and Robert J Wood. “Toward a modular soft sensor-embedded glove for human hand motion and tactile pressure measurement.” In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pp. 4000–4007. IEEE, 2014.

[Hof95] Douglas R Hofstadter. *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought.* Basic books, 1995.

[HOT06] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets.” *Neural Computation*, **18**(7):1527–1554, 2006.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory.” *Neural computation*, 1997.

[HS06] Geoffrey E Hinton and Ruslan R Salakhutdinov. “Reducing the dimensionality of data with neural networks.” *science*, **313**(5786):504–507, 2006.

[HS17] Bradley Hayes and Julie A Shah. “Improving robot controller transparency through autonomous policy explanation.” In *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2017.

[HSB19] Felix Hill, Adam Santoro, David GT Barrett, Ari S Morcos, and Timothy Lillicrap. “Learning to Make Analogies by Contrasting Abstract Relational Structure.” *arXiv:1902.00120*, 2019.

[HTA17] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. “Reinforcement Learning with Deep Energy-Based Policies.” In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[HW17] Dokhyam Hoshen and Michael Werman. “IQ of Neural Networks.” *arXiv preprint arXiv:1710.01692*, 2017.

[HYH21] Xiaotian Han, Jianwei Yang, Houdong Hu, Lei Zhang, Jianfeng Gao, and Pengchuan Zhang. “Image Scene Graph Generation (SGG) Benchmark.” *arXiv preprint arXiv:2107.12604*, 2021.

[HZR15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” In *CVPR*, 2015.

[HZR16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[IG20a] Gautier Izacard and Edouard Grave. "Distilling knowledge from reader to retriever for question answering." *arXiv preprint arXiv:2012.04584*, 2020.

[IG20b] Gautier Izacard and Edouard Grave. "Leveraging passage retrieval with generative models for open domain question answering." *arXiv preprint arXiv:2007.01282*, 2020.

[IRC98] Roberto Iglesias Rodriguez, Carlos Regueiro, J Correa, and S Barro. "Supervised reinforcement learning: Application to a wall following behaviour in a mobile robot." In *Tasks and Methods in Applied Artificial Intelligence*, 1998.

[IS15] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

[JBD00] Jiun-Yin Jian, Ann M Bisantz, and Colin G Drury. "Foundations for an empirically determined scale of trust in automated systems." *International Journal of Cognitive Ergonomics*, **4**(1):53–71, 2000.

[JBJ08] Susanne M Jaeggi, Martin Buschkuehl, John Jonides, and Walter J Perrig. "Improving fluid intelligence with training on working memory." *Proceedings of the National Academy of Sciences*, **105**(19):6829–6833, 2008.

[JDJ17] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-scale similarity search with GPUs." *arXiv preprint arXiv:1702.08734*, 2017.

[JGP16] Eric Jang, Shixiang Gu, and Ben Poole. "Categorical reparameterization with gumbel-softmax." *arXiv:1611.01144*, 2016.

[JHM17a] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[JHM17b] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross B Girshick. "Inferring and Executing Programs for Visual Reasoning." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[JKK21] Aman Jain, Mayank Kothyari, Vishwajeet Kumar, Preethi Jyothi, Ganesh Ramakrishnan, and Soumen Chakrabarti. "Select, Substitute, Search: A New Benchmark for Knowledge-Augmented Visual Question Answering." *arXiv preprint arXiv:2103.05568*, 2021.

[JLK11] Eunseok Jeong, Jaehong Lee, and DaeEun Kim. "Finger-gesture recognition glove using velostat." In *2011 11th International Conference on Control, Automation and Systems*, pp. 206–210. IEEE, 2011.

[JMR20] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. "In defense of grid features for visual question answering." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10267–10276, 2020.

[JQZ18] Chenfanfu Jiang, Siyuan Qi, Yixin Zhu, Siyuan Huang, Jenny Lin, Lap-Fai Yu, Demetri Terzopoulos, and Song-Chun Zhu. "Configurable 3D Scene Synthesis and 2D Image Rendering with Per-pixel Ground Truth Using Stochastic Grammars." *IJCV*, **126**(9):920–941, 2018.

[KAS15] Henk G Kortier, Jacob Antonsson, H Martin Schepers, Fredrik Gustafsson, and Peter H Veltink. "Hand pose estimation by fusion of inertial and magnetic sensing aided by a permanent magnet." *Transactions on Neural Systems and Rehabilitation Engineering*, **23**(5):796–806, 2015.

[KB12] Andrea Kleinsmith and Nadia Bianchi-Berthouze. "Affective body expression perception and recognition: A survey." *IEEE Transactions on Affective Computing*, **4**(1):15–33, 2012.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." *International Conference on Learning Representations (ICLR)*, 2014.

[KCC11] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. "Imitation learning of positional and force skills demonstrated via kinesthetic teaching and haptic input." *Advanced Robotics*, **25**(5):581–603, 2011.

[KCT11] Mrinal Kalakrishnan, Sachin Chitta, Evangelos Theodorou, Peter Pastor, and Stefan Schaal. "STOMP: Stochastic trajectory optimization for motion planning." In *International Conference on Robotics and Automation (ICRA)*, 2011.

[Kel99] Deborah Kelemen. "The scope of teleological thinking in preschool children." *Cognition*, **70**(3):241–272, 1999.

[KFM14] Kamil Kukliński, Kerstin Fischer, Ilka Marhenke, Franziska Kirstein, V Maria, Norbert Krüger, Thiusius Rajeeth Savarimuthu, et al. "Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control." In *ICUMT*. IEEE, 2014.

[KJZ18] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. "Bilinear Attention Networks." In *NeurIPS*, 2018.

[KK79] Gaetano Kanizsa and Gaetano Kanizsa. *Organization in vision: Essays on Gestalt perception*, volume 49. Praeger New York, 1979.

[KMG13] Maithilee Kunda, Keith McGreggor, and Ashok K Goel. "A computational model for solving problems from the Raven's Progressive Matrices intelligence test using iconic visual representations." *Cognitive Systems Research*, **22**:47–66, 2013.

[KMG17] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. "AI2-THOR: An interactive 3d environment for visual AI.", 2017.

[KMS11] Rebecca K Kramer, Carmel Majidi, Ranjana Sahai, and Robert J Wood. "Soft curvature sensors for joint angle proprioception." In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1919–1926. IEEE, 2011.

[KNC11] Petar Kormushev, Dragomir N Nenchev, Sylvain Calinon, and Darwin G Caldwell. "Upper-body kinesthetic teaching of a free-standing humanoid robot." In *ICRA*. IEEE, 2011.

[KNS16] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation." In *NIPS*, 2016.

[KOM20] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. "Dense passage retrieval for open-domain question answering." *arXiv preprint arXiv:2004.04906*, 2020.

[KPR19] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. "Natural questions: a benchmark for question answering research." *Transactions of the Association for Computational Linguistics*, **7**:453–466, 2019.

[KRD18] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. "Textual explanations for self-driving vehicles." In *European Conference on Computer Vision (ECCV)*, 2018.

[KSE08] Nidal S Kamel, Shohel Sayeed, and Grant A Ellis. "Glove-based approach to online signature verification." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(6):1109–1113, 2008.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2012.

[KSR14] Henk G Kortier, Victor I Sluiter, Daniel Roetenberg, and Peter H Veltink. “Assessment of hand kinematics using inertial and magnetic sensors.” *Journal of neuroengineering and rehabilitation*, **11**(1):1–15, 2014.

[KW16] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks.” *arXiv preprint arXiv:1609.02907*, 2016.

[KZG16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations.” *arXiv preprint arXiv:1602.07332*, 2016.

[KZG17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations.” *International journal of computer vision*, **123**(1):32–73, 2017.

[KZS15] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. “Skip-thought vectors.” In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[Law57] Reed Lawson. “Brightness discrimination performance and secondary reward strength as a function of primary reward amount.” *Journal of Comparative and Physiological Psychology*, **50**(1):35, 1957.

[LB04] Jeff Lieberman and Cynthia Breazeal. “Improvements on action parsing and action interpolation for learning through demonstration.” In *International Conference on Humanoid Robots*. IEEE, 2004.

[LBP19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. “Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.” *arXiv preprint arXiv:1908.02265*, 2019.

[LCC19] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. “Kagnet: Knowledge-aware graph networks for commonsense reasoning.” *arXiv preprint arXiv:1909.02151*, 2019.

[LCT19] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. “Latent retrieval for weakly supervised open domain question answering.” *arXiv preprint arXiv:1906.00300*, 2019.

[LF17] Andrew Lovett and Kenneth Forbus. “Modeling visual problem solving as analogical reasoning.” *Psychological Review*, **124**(1):60, 2017.

[LFU10] Andrew Lovett, Kenneth Forbus, and Jeffrey Usher. “A structure-mapping model of Raven’s Progressive Matrices.” In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2010.

[LGN20] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. “Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning.” *arXiv preprint arXiv:2012.15409*, 2020.

[LGX20] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. “Graph-based reasoning over heterogeneous external knowledge for commonsense question answering.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8449–8456, 2020.

[LH07] Shane Legg and Marcus Hutter. “Universal intelligence: A definition of machine intelligence.” *Minds and Machines*, **17**(4):391–444, 2007.

[LHP16] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. “Continuous control with deep reinforcement learning.” In *ICLR*, 2016.

[LJZ18] Pan Lu, Lei Ji, Wei Zhang, Nan Duan, Ming Zhou, and Jianyong Wang. “R-VQA: learning visual relation facts with semantic attention for visual question answering.” In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1880–1889, 2018.

[LKG14] Joshua D Langsfeld, Krishnanand N Kaipa, Rodolphe J Gentili, James A Reggia, and Satyandra K Gupta. “Incorporating failure-to-success transitions in imitation learning for a dynamic pouring task.” In *Workshop on Compliant Manipulation: Challenges and Control, Chicago, IL*, 2014.

[LKY15] Jin Huat Low, Phone May Khin, and Chen-Hua Yeow. “A pressure-redistributing insole using soft sensors and actuators.” In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2926–2930. IEEE, 2015.

[LLG12] Daniel R Little, Stephan Lewandowsky, and Thomas L Griffiths. “A Bayesian model of rule induction in Raven’s Progressive Matrices.” In *Annual Meeting of the Cognitive Science Society (CogSci)*, 2012.

[LLG15] Alex X Lee, Henry Lu, Abhishek Gupta, Sergey Levine, and Pieter Abbeel. “Learning Force-Based Manipulation of Deformable Objects from Multiple Demonstrations.” In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 177–184. IEEE, 2015.

[LLS15] Ian Lenz, Honglak Lee, and Ashutosh Saxena. "Deep learning for detecting robotic grasps." *The International Journal of Robotics Research*, **34**(4-5):705–724, 2015.

[LMB14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context." In *European conference on computer vision*, pp. 740–755. Springer, 2014.

[LMM01] Richard Le Grand, Catherine J Mondloch, Daphne Maurer, and Henry P Brent. "Neuroperception: Early visual experience and face processing." *Nature*, **410**(6831):890, 2001.

[Lom06] Tania Lombrozo. "The structure and function of explanations." *Trends in Cognitive Sciences*, **10**(10):464–470, 2006.

[Lom13] Tania Lombrozo. "Explanation and Abductive Inference." In *The Oxford Handbook of Thinking and Reasoning.* Oxford University Press, 2013.

[LS13] Gabriele Ligorio and Angelo Maria Sabatini. "Extended Kalman filter-based methods for pose estimation using visual, inertial and magnetic sensors: Comparative analysis and performance evaluation." *Sensors*, **13**(2):1919–1941, 2013.

[LS15] Byung Woo Lee and Hangsik Shin. "Feasibility study of sitting posture monitoring based on piezoresistive conductive film-based flexible force sensor." *IEEE Sensors Journal*, **16**(1):15–16, 2015.

[LST15] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. "Human-level concept learning through probabilistic program induction." *Science*, **350**(6266):1332–1338, 2015.

[LST21] Man Luo, Shailaja Keyur Sampat, Riley Tallman, Yankai Zeng, Manuha Vancha, Akarshan Sajja, and Chitta Baral. "'Just because you are right, doesn't mean I am wrong': Overcoming a bottleneck in development and evaluation of Open-Ended VQA tasks." In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2766–2771, 2021.

[LTF09] Andrew Lovett, Emmett Tomai, Kenneth Forbus, and Jeffrey Usher. "Solving geometric analogy problems through two-stage analogical mapping." *Cognitive science*, **33**(7):1192–1231, 2009.

[LW85] J David Lewis and Andrew Weigert. "Trust as a social reality." *Social forces*, **63**(4):967–985, 1985.

[LWH00] John Lin, Ying Wu, and Thomas S Huang. "Modeling the constraints of human hand motion." In *Proceedings workshop on human motion*, pp. 121–126. IEEE, 2000.

[LWP09] Liang Lin, Tianfu Wu, Jake Porway, and Zijian Xu. "A stochastic graph grammar for compositional object representation and recognition." *Pattern Recognition*, **42**(7):1297–1307, 2009.

[LXM17] Hangxin Liu, Xu Xie, Matt Millar, Mark Edmonds, Feng Gao, Yixin Zhu, Veronica J Santos, Brandon Rothrock, and Song-Chun Zhu. "A Glove-Based System for Studying Hand-Object Manipulation via Joint Pose and Force Sensing." In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6617–6624. IEEE, 2017.

[LYL20] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. "Oscar: Object-Semantics Aligned Pre-training for Vision-Language Tasks." *ECCV 2020*, 2020.

[LZB21] Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. "Weakly-Supervised Visual-Retriever-Reader for Knowledge-based Question Answering." *arXiv preprint arXiv:2109.04014*, 2021.

[LZX19] Hangxin Liu, Zhenliang Zhang, Xie Xu, Yixin Zhu, Yue Liu, Yongtian Wang, and Song-Chun Zhu. "High-Fidelity Grasping in Virtual Reality using a Glove-based System." In *ICRA*, 2019.

[LZZ19] Hangxin Liu, Chi Zhang, Yixin Zhu, Chenfanfu Jiang, and Song-Chun Zhu. "Mirroring without Overimitation: Learning Functionally Equivalent Manipulation Actions." In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.

[Mar82] David Marr. *Vision: A computational investigation into.* WH Freeman, 1982.

[Mar18] Gary Marcus. "Deep learning: A critical appraisal." *arXiv preprint arXiv:1801.00631*, 2018.

[MB01] Amy McGovern and Andrew G Barto. "Automatic discovery of subgoals in reinforcement learning using diverse density." In *Proceedings of International Conference on Machine Learning (ICML)*, 2001.

[MBM16] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning." In *ICML*, 2016.

[MBS16] Mostafa Mohammadi, Tommaso Lisini Baldi, Stefano Scheggi, and Domenico Prattichizzo. "Fingertip force estimation via inertial and magnetic sensors in deformable object manipulation." In *2016 IEEE Haptics Symposium (HAPTICS)*, pp. 284–289. IEEE, 2016.

[MCP21] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. "Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14111–14121, 2021.

[Mey51] Donald R Meyer. "The effects of differential rewards on discrimination reversal learning by monkeys." *Journal of Experimental Psychology*, **41**(4):268, 1951.

[MG14] Keith McGreggor and Ashok K Goel. "Confident Reasoning on Raven's Progressive Matrices Tests." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pp. 380–386, 2014.

[MGH09] Manuel Muhlig, Michael Gienger, Sven Hellbach, Jochen J Steil, and Christian Goerick. "Task-level imitation learning using variance-based movement optimization." In *ICRA*. IEEE, 2009.

[MGK16] Simon Manschitz, Michael Gienger, Jens Kober, and Jan Peters. "Probabilistic Decomposition of Sequential Force Interaction Tasks into Movement Primitives." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3920–3927. IEEE, 2016.

[Mit93] Melanie Mitchell. *Analogy-making as perception: A computer model.* MIT Press, 1993.

[MKG14a] Simon Manschitz, Jens Kober, Michael Gienger, and Jan Peters. "Learning to Sequence Movement Primitives from Demonstrations." In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4414–4421. IEEE, 2014.

[MKG14b] Keith McGreggor, Maithilee Kunda, and Ashok Goel. "Fractals and ravens." *Artificial Intelligence*, **215**:1–23, 2014.

[MKS15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning." *Nature*, **518**(7540):529–533, 2015.

[MLN17] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg. "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics." In *RSS*, 2017.

[MMH04] Shie Mannor, Ishai Menache, Amit Hoze, and Uri Klein. "Dynamic abstraction in reinforcement learning via clustering." In *Proceedings of International Conference on Machine Learning (ICML)*, 2004.

[MMS02] Ishai Menache, Shie Mannor, and Nahum Shimkin. "Q-cut—dynamic discovery of sub-goals in reinforcement learning." In *European Conference on Machine Learning*, 2002.

[MMT16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. "The concrete distribution: A continuous relaxation of discrete random variables." *arXiv:1611.00712*, 2016.

[MP17] Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[MRF19] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. "Ok-vqa: A visual question answering benchmark requiring external knowledge." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3195–3204, 2019.

[MRI04] David L. Moreno, Carlos V. Regueiro, Roberto Iglesias, and Senén Barro. "Using prior knowledge to improve reinforcement learning in mobile robotics." In *Towards Autonomous Robotics Systems*, 2004.

[MRL05] Bhaskara Marthi, Stuart J Russell, David Latham, and Carlos Guestrin. "Concurrent hierarchical reinforcement learning." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[MSB17] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. "Deepstack: Expert-level artificial intelligence in heads-up no-limit poker." *Science*, **356**(6337):508–513, 2017.

[MSC13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013.

[MSD18] Can Serif Mekik, Ron Sun, and David Yun Dai. "Similarity-Based Reasoning, Raven's Matrices, and General Intelligence." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1576–1582, 2018.

[MSG15] Steffen Müller, Christof Schröter, and H-M Gross. "Smart fur tactile sensor for a socially assistive mobile robot." In *International Conference on Intelligent Robotics and Applications*, pp. 49–60. Springer, 2015.

[MT97] Norman McCain, Hudson Turner, et al. "Causal theories of action and change." In *AAAI*, 1997.

[MTS18] David Mascharka, Philip Tran, Ryan Soklaski, and Arjun Majumdar. "Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[NC36] Isaac Newton and John Colson. *The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines*. Henry Woodfall; and sold by John Nourse, 1736.

[New73] Allen Newell. "You can't play 20 questions with nature and win: Projective comments on the papers of this symposium." In William G Chase, editor, *Visual Information Processing: Proceedings of the Eighth Annual Carnegie Symposium on Cognition*. Academic Press, 1973.

[NH10] Vinod Nair and Geoffrey E Hinton. "Rectified linear units improve restricted boltzmann machines." In *Proceedings of International Conference on Machine Learning (ICML)*, 2010.

[PDP13] Alexandros Paraschos, Christian Daniel, Jan R Peters, and Gerhard Neumann. "Probabilistic Movement Primitives." *Advances in Neural Information Processing Systems (NeurIPS)*, **26**, 2013.

[Pea09] Judea Pearl et al. "Causal inference in statistics: An overview." *Statistics Surveys*, **3**:96–146, 2009.

[PGC17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in PyTorch." In *NIPS-W*, 2017.

[PGH16] Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. "The curious robot: Learning visual representations via physical interactions." In *European Conference on Computer Vision (ECCV*, pp. 3–18. Springer, 2016.

[PJK16] Chris Paxton, Felix Jonathan, Marin Kobilarov, and Gregory D Hager. "Do what I want, not what I did: Imitation of skills by planning sequences of actions." In *IROS*. IEEE, 2016.

[PK15] Karl Pauwels and Danica Kragic. "Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking." In *IROS*. IEEE, 2015.

[PKQ15] Tu-Hoa Pham, Abderrahmane Kheddar, Ammar Qammaz, and Antonis A Argyros. "Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces." In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2810–2819, 2015.

[PMT16] Ganna Pugach, Artem Melnyk, Olga Tolochko, Alexandre Pitti, and Philippe Gaussier. "Touch-based admittance control of a robotic arm using neural learning of an artificial skin." In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3374–3380. IEEE, 2016.

[PNZ15] Seyoung Park, Bruce Xiaohan Nie, and Song-Chun Zhu. "Attribute And-Or Grammar for Joint Parsing of Human Attributes, Part and Pose." *ICCV*, 2015.

[Pop10] Ronald Poppe. "A Survey on Vision-Based Human Action Recognition." *Image and vision computing*, **28**(6):976–990, 2010.

[PR98] Ronald Parr and Stuart J Russell. "Reinforcement learning with hierarchies of machines." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 1998.

[Pre00] Doina Precup. *Temporal abstraction in reinforcement learning.* University of Massachusetts Amherst, 2000.

[PSD18] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. "FiLM: Visual Reasoning with a General Conditioning Layer." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In *Proceedings of the conference on Empirical Methods in Natural Language Processing (EMNLP*, 2014.

[PSY13] Mingtao Pei, Zhangzhang Si, Benjamin Z Yao, and Song-Chun Zhu. "Learning and parsing video events with goal and intent prediction." *Computer Vision and Image Understanding*, **117**(10):1369–1383, 2013.

[PZ15] Seyoung Park and Song-Chun Zhu. "Attributed grammars for joint estimation of human attributes, part and pose." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[QCG09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jermey Leibs, Rob Wheeler, and Andrew Y. Ng. "ROS: an open-source Robot Operating System." In *ICRA Workshop on Open Source Software*, 2009.

[QHW17] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. "Predicting human activities using stochastic grammar." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1164–1172, 2017.

[QJZ18] Siyuan Qi, Baoxiong Jia, and Song-Chun Zhu. "Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction." In *International Conference on Machine Learning*, pp. 4171–4179. PMLR, 2018.

[QZH18] Siyuan Qi, Yixin Zhu, Siyuan Huang, Chenfanfu Jiang, and Song-Chun Zhu. "Human-centric Indoor Scene Synthesis Using Stochastic Grammar." In *CVPR*, 2018.

[QZY21] Chen Qu, Hamed Zamani, Liu Yang, W Bruce Croft, and Erik Learned-Miller. "Passage Retrieval for Outside-Knowledge Visual Question Answering." *arXiv preprint arXiv:2105.03938*, 2021.

[RA07] Deepak Ramachandran and Eyal Amir. "Bayesian inverse reinforcement learning." In *IJCAI*, volume 51, pp. 1–4, 2007.

[RA15] Siddharth S Rautaray and Anupam Agrawal. "Vision based hand gesture recognition for human computer interaction: a survey." *Artificial intelligence review*, **43**(1):1–54, 2015.

[Rav36] James C Raven. *"Mental tests used in genetic studies: The performance of related individuals on tests mainly educative and mainly reproductive."*. Master's thesis, University of London, 1936.

[Rav38] J. C. et al. Raven. "Raven's progressive matrices." *Western Psychological Services*, 1938.

[RC98] John C Raven and John Hugh Court. *Raven's progressive matrices and vocabulary scales.* Oxford pyschologists Press, 1998.

[RCC16] Leonel Rozo, Sylvain Calinon, Darwin G Caldwell, Pablo Jimenez, and Carme Torras. "Learning Physical Collaborative Robot Behaviors from Human Demonstrations." *IEEE Transactions on Robotics*, **32**(3):513–527, 2016.

[RDG16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[RGB11] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. "A reduction of imitation learning and structured prediction to no-regret online learning." In *Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTAT)*, pp. 627–635, 2011.

[RHG15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[RJL18] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know what you don't know: Unanswerable questions for SQuAD." *arXiv preprint arXiv:1806.03822*, 2018.

[RKZ15] Mengye Ren, Ryan Kiros, and Richard Zemel. "Exploring models and data for image question answering." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[RSR19] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer." *arXiv preprint arXiv:1910.10683*, 2019.

[RSR20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." *Journal of Machine Learning Research*, **21**(140):1–67, 2020.

[RZ09] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond.* Now Publishers Inc, 2009.

[RZB09] Nathan Ratliff, Matt Zucker, J Andrew Bagnell, and Siddhartha Srinivasa. "CHOMP: gradient optimization techniques for efficient motion planning." In *International Conference on Robotics and Automation (ICRA)*, 2009.

[RZL16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text." In *EMNLP*, 2016.

[SAL21] Ander Salaberria, Gorka Azkune, Oier Lopez de Lacalle, Aitor Soroa, and Eneko Agirre. "Image Captioning for Effective Use of Language Models in Knowledge-Based Visual Question Answering." *arXiv preprint arXiv:2109.08029*, 2021.

[SB98] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 1998.

[SB08] Richard Souvenir and Justin Babbs. "Learning the viewpoint manifold for action recognition." In *CVPR.* IEEE, 2008.

[SBP04] Jennie Si, Andrew G. Barto, Warren Buckler Powell, and Don Wunsch. *Handbook of Learning and Approximate Dynamic Programming.* Wiley-IEEE Press, 2004.

[SC13] Ioan A Sucan and Sachin Chitta. "Moveit!" *Online at http://moveit. ros. org*, 2013.

[SCH17] Robyn Speer, Joshua Chin, and Catherine Havasi. "Conceptnet 5.5: An open multilingual graph of general knowledge." In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[SCO11] Daniel L Schwartz, Catherine C Chase, Marily A Oppezzo, and Doris B Chin. "Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer." *Journal of Educational Psychology*, **103**(4):759, 2011.

[SCS13] Claes Strannegård, Simone Cirillo, and Victor Ström. "An anthropomorphic method for progressive matrix problems." *Cognitive Systems Research*, **22**:35–46, 2013.

[SDL18] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. "Airsim: High-fidelity visual and physical simulation for autonomous vehicles." In *Field and service robotics*, pp. 621–635. Springer, 2018.

[SDZ18] Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W Cohen. "Open domain question answering using early fusion of knowledge bases and text." *arXiv preprint arXiv:1809.00782*, 2018.

[SE05] Noah A Smith and Jason Eisner. "Contrastive estimation: Training log-linear models on unlabeled data." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2005.

[SG14] Linsey Smith and Dedre Gentner. "The role of difference-detection in learning contrastive categories." In *Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci)*, 2014.

[SG18a] Snejana Shegheva and Ashok Goel. "The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[SG18b] Snejana Shegheva and Ashok K. Goel. "The Structural Affinity Method for Solving the Raven's Progressive Matrices Test for Intelligence." In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[SGG08] Andreas J Schmid, Nicolas Gorges, Dirk Goger, and Heinz Worn. "Opening a door with a humanoid robot using multi-sensory tactile feedback." In *ICRA*. IEEE, 2008.

[SGR17] Tianmin Shu, Xiaofeng Gao, Michael S Ryoo, and Song-Chun Zhu. "Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions." In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1669–1676. IEEE, 2017.

[SH56] Allan M Schrier and Harry F Harlow. "Effect of amount of incentive on discrimination learning by monkeys." *Journal of comparative and physiological psychology*, **49**(2):117, 1956.

[SHK14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research*, 2014.

[SHL13] John Schulman, Jonathan Ho, Alex X Lee, Ibrahim Awwal, Henry Bradlow, and Pieter Abbeel. "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization." In *Robotics: Science and Systems (RSS)*, 2013.

[SHM16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search." *nature*, **529**(7587):484, 2016.

[Sim07] Jeffry A Simpson. "Psychological foundations of trust." *Current directions in psychological science*, **16**(5):264–268, 2007.

[SKM19] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. "Habitat: A platform for embodied ai research." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9339–9347, 2019.

[SLB10] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. "Intrinsically motivated reinforcement learning: An evolutionary perspective." *IEEE Transactions on Autonomous Mental Development*, **2**(2):70–82, 2010.

[SLS15] Gaspare Santaera, Emanuele Luberto, Alessandro Serio, Marco Gabiccini, and Antonio Bicchi. "Low-cost, fast and accurate reconstruction of robotic and human postures via IMU measurements." In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2728–2735. IEEE, 2015.

[SLV18] Xander Steenbrugge, Sam Leroux, Tim Verbelen, and Bart Dhoedt. "Improving generalization for abstract reasoning tasks using disentangled feature representations." *arXiv preprint arXiv:1811.04784*, 2018.

[SM04] Daniel L Schwartz and Taylor Martin. "Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction." *Cognition and Instruction*, **22**(2):129–184, 2004.

[SMS00] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. "Policy gradient methods for reinforcement learning with function approximation." In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2000.

[SMY19] Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. "Kvqa: Knowledge-aware visual question answering." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8876–8884, 2019.

[SNS19] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. “Towards vqa models that can read.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

[SP12] Susanne Still and Doina Precup. “An information-theoretic approach to curiosity-driven reinforcement learning.” *Theory in Biosciences*, **131**(3):139–148, 2012.

[Spe23] Charles Spearman. *The nature of ”intelligence” and the principles of cognition.* Macmillan, 1923.

[Spe27] Charles Spearman. *The abilities of man.* Macmillan, 1927.

[SPN05] Stefan Schaal, Jan Peters, Jun Nakanishi, and Auke Ijspeert. “Learning Movement Primitives.” In *Robotics research. the eleventh international symposium*, pp. 561–572. Springer, 2005.

[SPS99] Richard S Sutton, Doina Precup, and Satinder Singh. “Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning.” *Artificial Intelligence*, **112**(1-2):181–211, 1999.

[SRB17] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap. “A simple neural network module for relational reasoning.” In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[SRZ16] Tianmin Shu, Michael S Ryoo, and Song-Chun Zhu. “Learning social affordance for human-robot interaction.” *arXiv preprint arXiv:1604.03692*, 2016.

[SSK13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. “Real-time human pose recognition in parts from single depth images.” *Communications of the ACM*, **56**(1):116–124, 2013.

[SSS17a] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. “Mastering the game of go without human knowledge.” *nature*, **550**(7676):354–359, 2017.

[SSS17b] Jaeyong Sung, J Kenneth Salisbury, and Ashutosh Saxena. “Learning to Represent Haptic Feedback for Partially-Observable Tasks.” pp. 2802–2809, 2017.

[ST00] Brian J Scholl and Patrice D Tremoulet. “Perceptual causality and animacy.” *Trends in cognitive sciences*, **4**(8):299–309, 2000.

[STC16] T Shu, S Thurman, D Chen, SC Zhu, and H Lu. “Critical features of joint actions that signal human interaction.” In *Proceedings of the 38th annual meeting of the cognitive science society (CogSci)*, 2016.

[STD21] Violetta Shevchenko, Damien Teney, Anthony Dick, and Anton van den Hengel. “Reasoning over Vision and Language: Exploring the Benefits of Supplemental Knowledge.” *arXiv preprint arXiv:2101.06013*, 2021.

[Sto95] Andreas Stolcke. “An efficient probabilistic context-free parsing algorithm that computes prefix probabilities.” *Computational linguistics*, 1995.

[SV78] Robert M Shapley and Jonathan D Victor. “The effect of contrast on the transfer properties of cat retinal ganglion cells.” *The Journal of physiology*, **285**(1):275–298, 1978.

[SWB05] Özgür Şimşek, Alicia P Wolfe, and Andrew G Barto. “Identifying useful subgoals in reinforcement learning by local graph partitioning.” In *Proceedings of International Conference on Machine Learning (ICML)*, 2005.

[SXS18] Tianmin Shu, Caiming Xiong, and Richard Socher. “Hierarchical and Interpretable Skill Acquisition in Multi-task Reinforcement Learning.” In *International Conference on Learning Representations (ICLR)*, 2018.

[SYZ17] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. “Semantic Scene Completion From a Single Depth Image.” In *CVPR*, 2017.

[TB19] Hao Tan and Mohit Bansal. “Lxmert: Learning cross-modality encoder representations from transformers.” *arXiv preprint arXiv:1908.07490*, 2019.

[TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. “Mujoco: A physics engine for model-based control.” In *IROS*, 2012.

[TGZ17] Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J Mankowitz, and Shie Mannor. “A Deep Hierarchical Approach to Lifelong Learning in Minecraft.” In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[THG16] Gilles Tagne, Patrick Hénaff, and Nicolas Gregori. “Measurement and analysis of physical parameters of the handshake between two persons according to simple social contexts.” In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 674–679. IEEE, 2016.

[TKM13] Thomas Taylor, Seungoh Ko, Carlos Mastrangelo, and Stacy J Morris Bamberg. “Forward kinematics using imu on-body sensor network for mobile analysis of human kinematics.” In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1230–1233. IEEE, 2013.

[Tom10] Michael Tomasello. *Origins of human communication.* MIT press, 2010.

[TPZ13] Kewei Tu, Maria Pavlovskaia, and Song-Chun Zhu. "Unsupervised Structure Learning of Stochastic And-Or Grammars." volume 26, 2013.

[TSM15] Kai Sheng Tai, Richard Socher, and Christopher D Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks." In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[TT41] Louis Leon Thurstone and Thelma Gwinn Thurstone. "Factorial studies of intelligence." *Psychometric monographs*, 1941.

[TZS16] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. "Movieqa: Understanding stories in movies through question-answering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4631–4640, 2016.

[VDL19] Ramakrishna Vedantam, Karan Desai, Stefan Lee, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. "Probabilistic neural symbolic models for interpretable visual question answering." In *International Conference on Machine Learning*, pp. 6428–6437. PMLR, 2019.

[VGL12] Michal Valko, Mohammad Ghavamzadeh, and Alessandro Lazaric. "Semi-Supervised Apprenticeship Learning." In *EWRL*, 2012.

[VOS17] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. "FeUdal Networks for Hierarchical Reinforcement Learning." In *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[VSP17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2017.

[Wat89] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, King's College, Cambridge, 1989.

[WG15] Xiaolong Wang and Abhinav Gupta. "Unsupervised learning of visual representations using videos." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[WKM19] Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, et al. "Improving natural language inference using external knowledge in the science questions domain." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 7208–7215, 2019.

[WLS21] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. "Multi-Modal Answer Validation for Knowledge-Based VQA." *arXiv preprint arXiv:2103.12248*, 2021.

[WLW13] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. "Learning actionlet ensemble for 3D human action recognition." *IEEE transactions on pattern analysis and machine intelligence*, **36**(5):914–927, 2013.

[WMZ13] Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. "Video-based hand manipulation capture through composite motion control." *ACM Transactions on Graphics (TOG)*, **32**(4):1–14, 2013.

[WNM19] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. "Multi-passage bert: A globally normalized bert model for open-domain question answering." *arXiv preprint arXiv:1908.08167*, 2019.

[WNX14] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. "Cross-view action modeling, learning and recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2649–2656, 2014.

[WRB11] Daniel Weinland, Remi Ronfard, and Edmond Boyer. "A Survey of Vision-Based Methods For Action Representation, Segmentation and Recognition." *Computer Vision and Image Understanding*, **115**(2):224–241, 2011.

[WS09] Kilian Q Weinberger and Lawrence K Saul. "Distance metric learning for large margin nearest neighbor classification." *Journal of Machine Learning Research*, **10**(Feb):207–244, 2009.

[WS15] Ke Wang and Zhendong Su. "Automatic Generation of Raven's Progressive Matrices." In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[WSG10] Ying Nian Wu, Zhangzhang Si, Haifeng Gong, and Song-Chun Zhu. "Learning active basis model for object detection and recognition." *International Journal of Computer Vision (IJCV)*, **90**(2):198–235, 2010.

[WSH16] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. "Dueling Network Architectures for Deep Reinforcement Learning." In *ICML*, 2016.

[WWS15] Peng Wang, Qi Wu, Chunhua Shen, Anton van den Hengel, and Anthony Dick. "Explicit knowledge-based reasoning for visual question answering." *arXiv preprint arXiv:1511.02570*, 2015.

[WWS17] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. "Fvqa: Fact-based visual question answering." *IEEE transactions on pattern analysis and machine intelligence*, **40**(10):2413–2427, 2017.

[WXL18] Ying Nian Wu, Jianwen Xie, Yang Lu, and Song-Chun Zhu. "Sparse and deep generalizations of the FRAME model." *Annals of Mathematical Sciences and Applications*, **3**(1):211–254, 2018.

[WXZ07] Tian-Fu Wu, Gui-Song Xia, and Song-Chun Zhu. "Compositional boosting for computing hierarchical image structures." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[WZ11] Tianfu Wu and Song-Chun Zhu. "A numerical study of the bottom-up and top-down inference processes in and-or graphs." *International journal of computer vision*, **93**(2):226–252, 2011.

[WZZ13a] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. "Modeling 4d human-object interactions for event and object recognition." In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pp. 3272–3279, 2013.

[WZZ13b] Ping Wei, Nanning Zheng, Yibiao Zhao, and Song-Chun Zhu. "Concurrent action detection with structural prediction." In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3136–3143, 2013.

[WZZ16] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. "Modeling 4d human-object interactions for joint event segmentation, recognition, and object localization." *IEEE transactions on pattern analysis and machine intelligence*, **39**(6):1165–1179, 2016.

[XCW15] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[XLE18] Xu Xie, Hangxin Liu, Mark Edmonds, Feng Gaol, Siyuan Qi, Yixin Zhu, Brandon Rothrock, and Song-Chun Zhu. "Unsupervised learning of hierarchical models for hand-object interactions." In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4097–4102. IEEE, 2018.

[XLZ16] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. "A theory of generative convnet." In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.

[XLZ19] Xu Xie, Hangxin Liu, Zhenliang Zhang, Yuxing Qiu, Feng Gao, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. "Vrgym: A virtual testbed for physical and interactive ai." In *Proceedings of the ACM Turing Celebration Conference-China*, pp. 1–6, 2019.

[XZH18] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. "Gibson Env: Real-World Perception for Embodied Agents." In *CVPR*, 2018.

[YGW21] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. "An Empirical Study of GPT-3 for Few-Shot Knowledge-Based VQA." *arXiv preprint arXiv:2109.05014*, 2021.

[YKY18] Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. "Representer Point Selection for Explaining Deep Neural Networks." In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[YLF15] Yezhou Yang, Yi Li, Cornelia Fermuller, and Yiannis Aloimonos. "Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web." In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 29, 2015.

[YLL20] Xiaofeng Yang, Guosheng Lin, Fengmao Lv, and Fayao Liu. "TRRNet: Tiered Relation Reasoning for Compositional Visual Question Answering." In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 414–430. Springer, 2020.

[YRB21] Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. "QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering." *arXiv preprint arXiv:2104.06378*, 2021.

[YWG18] Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. "Neural-Symbolic VQA: Disentangling Reasoning from Vision and Language Understanding." *arXiv preprint arXiv:1810.02338*, 2018.

[YXL19] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. "End-to-End Open-Domain Question Answering with BERTserini." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 72–77, 2019.

[YYC19] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. “Deep modular co-attention networks for visual question answering.” In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6281–6290, 2019.

[YYT11] Lap-Fai Yu, Sai-Kit Yeung, Chi-Keung Tang, Demetri Terzopoulos, Tony F Chan, and Stanley J Osher. “Make it home: automatic optimization of furniture arrangement.” *TOG*, **30**(4):86, 2011.

[YYX18] Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. “Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering.” *IEEE transactions on neural networks and learning systems*, **29**(12):5947–5959, 2018.

[ZGB16] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. “Visual7w: Grounded question answering in images.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[ZGJ19] Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. “RAVEN: A Dataset for Relational and Analogical Visual rEasoNing.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[ZJG19] Chi Zhang, Baoxiong Jia, Feng Gao, Yixin Zhu, Hongjing Lu, and Song-Chun Zhu. “Learning perceptual inference by contrasting.” *Advances in Neural Information Processing Systems*, **32**, 2019.

[ZJZ16] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. “Inferring forces and learning human utilities from videos.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3823–3833, 2016.

[ZLH21] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. “VinVL: Making Visual Representations Matter in Vision-Language Models.” *CVPR 2021*, 2021.

[ZM07] Song-Chun Zhu, David Mumford, et al. “A Stochastic Grammar of Images.” *Foundations and Trends in Computer Graphics and Vision*, **2**(4):259–362, 2007.

[ZMB08] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. “Maximum Entropy Inverse Reinforcement Learning.” In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

[ZNS19] Yundong Zhang, Juan Carlos Niebles, and Alvaro Soto. “Interpretable visual question answering by visual grounding from attention supervision mining.” In

2019 ieee winter conference on applications of computer vision (wacv), pp. 349–357. IEEE, 2019.

[ZNZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable convolutional neural networks." In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[ZSW18] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning." In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4238–4245. IEEE, 2018.

[ZTH08] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. "Aligned cluster analysis for temporal segmentation of human motion." In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, 2008.

[ZTH12] Feng Zhou, Fernando De la Torre, and Jessica K Hodgins. "Hierarchical aligned cluster analysis for temporal clustering of human motion." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(3):582–596, 2012.

[ZW11] Ting Zhang and Wei Biao Wu. "Testing parametric assumptions of trends of a nonstationary time series." *Biometrika*, **98**(3):599–614, 2011.

[ZWM98] Song Chun Zhu, Yingnian Wu, and David Mumford. "Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling." *International Journal of Computer Vision*, **27**(2):107–126, 1998.

[ZWZ16] Jun Zhu, Tianfu Wu, Song-Chun Zhu, Xiaokang Yang, and Wenjun Zhang. "A reconfigurable tangram model for scene representation and categorization." *IEEE Transactions on Image Processing*, **25**(1):150–166, 2016.

[ZZH17] Chen Zhu, Yanpeng Zhao, Shuaiyi Huang, Kewei Tu, and Yi Ma. "Structured attentions for visual question answering." In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[ZZM13] Wenping Zhao, Jianjie Zhang, Jianyuan Min, and Jinxiang Chai. "Robust real-time physics-based motion control for human grasping." *ACM Transactions on Graphics (TOG)*, **32**(6):1–12, 2013.

[ZZZ15] Yixin Zhu, Yibiao Zhao, and Song-Chun Zhu. "Understanding tools: Task-oriented object modeling, learning and recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2855–2864, 2015.