Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Active Vision for Embodied Agents Using Reinforcement Learning

Dissertation
submitted to the Universität Hamburg
with the aim of achieving a doctoral degree at the Faculty of Mathematics, Informatics and Natural Science, Department of Informatics.

Mengdi Li
Hamburg, 2024

Submission of the dissertation:
18.04.2024

Day of oral defense:
04.06.2024

Dissertation Committee:

Prof. Dr. Stefan Wermter (advisor)
Dept. of Computer Science
Universität Hamburg, Germany

Prof. Dr. Jianwei Zhang (reviewer)
Dept. of Computer Science
Universität Hamburg, Germany

Prof. Dr. Janick Edinger (chair)
Dept. of Computer Science
Universität Hamburg, Germany

Abstract

Embodied agents can only partially observe their surroundings from their egocentric perspectives at any moment. This leads to the prevalent situations of insufficient observations, where the agents respond according to insufficient task-relevant information. To resolve this, they must actively explore the environment according to task requirements to collect sufficient task-related information. The development of the capability of active vision plays an essential role in embodied artificial intelligence agents operating in real-world application scenarios.

Different from the majority of existing work on embodied agents that is focused on learning active vision control, as seen in tasks such as object-goal navigation, this thesis concentrates on action-response embodied agents that have two distinct output channels: one for active vision control for goal-oriented visual information collection, and the other for task-relevant responses. Embodied agents of this setup are natural, aligning closely with human capabilities, and are especially needed in scenarios where rich interaction between agents and human users plays an essential role. However, research on this kind of embodied agent has not yet received much attention.

This thesis first studies disembodied models in situations of insufficient observations to investigate to what extent the issue of insufficient observations can be addressed without the application of active vision. Specifically, we study the issue in a special setup of the task of visual question answering (VQA), where the visual information of an image is possibly insufficient to answer a given question. Our experiments demonstrate that it is non-trivial to develop disembodied models capable of detecting the sufficiency of perceived information while giving accurate responses when the information is sufficient. In addition, our approach reveals an inherent limitation of disembodied AI models, i.e., the lack of the capability of active perception hinders the development of a progressive model that can produce helpful responses in situations of insufficient observations. This motivates our work on embodied agents with the capability of active vision.

Then, driven by the inherent limitation of disembodied AI models, we focus on the development of active vision control for embodied AI agents. Motivated by the neuroscientific learning theory that the components of sensory perception, attention mechanisms, and value evaluation are all involved in the rewarding process in the brains of humans and animals, we model the action-response agent utilizing a modular network and train the active vision control policy through reinforcement learning (RL). To effectively evaluate the performance of the model and the training method, we introduce the task of robotic object existence prediction (ROEP), where the situation of insufficient observations arises from potential occlusions between objects. The experimental results demonstrate the effectiveness of the proposed model and the training method in developing action-response agents.

Next, motivated by the observation that the efficient training of the proposed model is challenging, this thesis analyzes the training process and generalizes the learning paradigm of the proposed model into a novel reinforcement learning framework, namely, internally rewarded reinforcement learning (IRRL). Theoretical and empirical analyses demonstrate that the inherent issues of noisy rewards and insufficient observations in the training process of IRRL lead to an unstable training loop where neither the policy nor the discriminator can learn effectively. It is proven that the shape of the reward function has an impact on the stability of the training process, based on which the clipped linear reward function is proposed to mitigate the unstable training issue.

In summary, the task setups, simulation environments, methodologies, and findings presented in this thesis contribute to the development of active vision for embodied agents and associated areas within the realm of reinforcement learning. The reinforcement learning framework proposed in this thesis, which incorporates diverse components such as visual perception, active vision control, and task-relevant discrimination, provides a unified approach to the development of active vision for action-response embodied agents, serving as a fundamental contribution.

Zusammenfassung

Verkörperte Agenten können zu jedem Zeitpunkt ihre Umgebung nur partiell aus ihrer egozentrischen Perspektive erfassen. Dies führt häufig zu Situationen, in denen Beobachtungen unvollständig sind und die Agenten auf Basis der unvollständigen, aufgabenbezogenen Informationen reagieren müssen. Um diese Herausforderung zu bewältigen, ist es erforderlich, dass sie ihre Umwelt entsprechend den Erfordernissen der Aufgabe aktiv erkunden, um genügend relevante Informationen zu sammeln. Die Entwicklung des aktiven Sehens ist von wesentlicher Bedeutung für verkörperte KI-Agenten, die in realistischen Szenarien eingesetzt werden.

Im Unterschied zu einem Großteil der bisherigen Forschungen über verkörperte Agenten, die vornehmlich das aktive Sehen in Kontexten wie Objekt-Ziel-Navigation thematisieren, richtet diese Arbeit ihr Augenmerk auf verkörperte Agenten, die mit zwei verschiedenen Ausgabekanälen ausgestattet sind: einem für die Steuerung des aktiven Sehens zur gezielten Sammlung visueller Informationen und einem zweiten für aufgabenbezogene Handlungen. Solche verkörperten Agenten sind natürlich und entsprechen den menschlichen Fähigkeiten. Sie werden vor allem in Szenarien benötigt, in denen eine reiche Interaktion zwischen Agenten und menschlichen Nutzern eine wesentliche Rolle spielt. Allerdings hat die Forschung zu dieser Art von verkörperten Agenten noch nicht viel Aufmerksamkeit erhalten.

In der vorliegenden Arbeit werden zunächst Modelle ohne physische Präsenz in Kontexten mit unvollständigen Beobachtungen untersucht, um zu ergründen, inwiefern die Herausforderung unvollständiger Information auch ohne den Einsatz aktiven Sehens bewältigt werden kann. Konkret wird das Problem in einem speziellen Szenario betrachtet, bei dem die visuellen Informationen eines Bildes möglicherweise nicht genügen, um eine gegebene Fragestellung zu beantworten. Die Untersuchung zeigt, dass die Entwicklung nicht-physischer Modelle, die sowohl eine Situation unvollständiger Information erkennen, als auch in Situationen ausreichender Datenlage präzise Antworten liefern können, eine keineswegs triviale Aufgabe darstellt. Zudem wird eine grundlegende Einschränkung derartiger nicht-physischer KI-Mod-elle aufgezeigt: Es lässt sich lediglich ein konservatives Modell realisieren, das in Szenarien mit unvollständiger Information die Ausgabe potenziell schädlicher Antworten vermeidet. Diese Erkenntnis legt den Grundstein für weiterführende Arbeiten an verkörperten Agenten, die durch aktives Sehen in der Lage sind, proaktiv die benötigten Informationen zu erwerben und so Reaktionen zu erzeugen, die nicht nur unschädlich, sondern auch von Nutzen sind.

Angesichts der grundlegenden Einschränkungen nicht-physischer KI-Modelle fokussieren wir uns auf die Entwicklung einer fortschrittlichen Steuerung des aktiven Sehens für verkörperte KI-Agenten. Inspiriert von neurowissenschaftlichen Lerntheorien, welche die Integration von Sinneswahrnehmung, Aufmerksamkeitsmechanismen und Bewertungsprozessen in den Belohnungsmechanismus des menschlichen und tierischen Gehirns betonen, entwerfen wir den Action-Response-Agenten mittels eines modularen Netzwerks und trainieren die Strategie für die Steuerung des aktiven Sehens durch Reinforcement Learning (RL). Zur effektiven Leistungbewertung des Modells und der Trainingsmethode implementieren wir die

Aufgabe für den Roboter, die Existenz von Objekten vorherzusagen (ROEP), die durch mögliche Verdeckungen durch andere Objekte bei unvollständigen Beobachtungen geprägt ist. Experimentelle Ergebnisse belegen die Machbarkeit, ein Modell für einen Action-Response-Agenten zu entwickeln, der sowohl in der Lage ist, die Steuerung des aktiven Sehens auszuüben als auch aufgabenbezogene Aktionen durchzuführen.

Anschließend, inspiriert von der Erkenntnis, dass das effiziente Training des vorgeschlagenen Modells eine signifikante Herausforderung darstellt, widmet sich diese Arbeit einer detaillierten Analyse des Trainingsprozesses und erweitert das Lernkonzept des vorgeschlagenen Modells zu einem innovativen Reinforcement-Learning Konzept, dem intern belohnten Reinforcement Learning (IRRL). Theoretische Überlegungen und empirische Untersuchungen zeigen, dass die grundlegenden Problematiken von verrauschten Belohnungssignalen und unvollständigen Beobachtungen innerhalb des IRRL-Trainingsprozesses zu einer instabilen Lernschleife führen, in welcher weder die Strategiefindung noch das Diskriminatorlernen effektiv stattfinden können. In dieser Arbeit konnte nachgewiesen werden, dass die Gestaltung der Belohnungsfunktion einen entscheidenden Einfluss auf die Stabilität des Trainingsverlaufs nimmt. Vor diesem Hintergrund wird eine gedeckelte lineare Belohnungsfunktion vorgeschlagen, um das Problem der Trainingsinstabilität zu entschärfen.

Zusammenfassend trägt diese Arbeit durch die Präsentation spezifischer Aufgabenstellungen, Simulationsumgebungen, Methoden und Ergebnisse wesentlich zur Fortentwicklung des aktiven Sehens für verkörperte Agenten und zu angrenzenden Themenfeldern im Kontext des RL bei. Der in dieser Arbeit vorgeschlagene Rahmen für das RL, der verschiedene Komponenten wie die visuelle Wahrnehmung, die Steuerung des aktiven Sehens und die aufgabenbezogene Differenzierung integriert, bietet einen einheitlichen Ansatz für die Entwicklung des aktiven Sehens für verkörperte Agenten und stellt einen grundlegenden Beitrag dar.

Contents

Chapter 1

Introduction

1.1 Motivation

Due to the inherent constraints of sensors and the presence of occlusions, humans and animals are always only able to partially perceive the environment they inhabit. The visual perception of humans is limited to the view directly in front of us. Thus, when searching for something, even in a small room, we often need to turn our bodies around or interact with objects to remove potential occlusion to fully explore the environment. Similarly, the field of view of robots is constrained by the state of their physical bodies. We expect that the visual perception of robots operates in a similar pattern to that of humans in the 3D world, with the ability to actively adjust the camera's position, orientation, and focus.

Embodied agents possess physical or virtual bodies in real-world or simulated environments, with the capability of interacting with their environments [Duan et al., 2022]. Due to their maneuverability, they can actively adjust the state of their vision system to perceive freely in the environment like humans. An ideal embodied agent, when presented with a specific objective, is expected to proactively collect the necessary information to accomplish the goal. Nevertheless, in reality, the issue of *insufficient observations* could arise, posing challenges in the development of the active visual system. The unexpected situation happens when the observed information is still not enough to provide a reliable response, yet the agent chooses to cease further exploration. This issue of insufficient observations potentially leads to harmful and helpless responses from an intelligent assistant.

The expected functionality of an embodied agent is determined by its target tasks. Depending on the setup of target tasks, the active-vision agent could have *a single-output channel* or *dual-output channels*. The majority of existing work in embodied agents with active vision is focused on only learning active vision control actions, as seen in tasks such as object-goal navigation [Anderson et al., 2018a], embodied language grounding [Hermann et al., 2017], and robotic instruction following [Anderson et al., 2018c]. In these scenarios, the only output of the agent is the active vision control action (see Fig. 1.1). Another class of scenarios that is realistic but has not yet gained much attention involves agents with two distinct

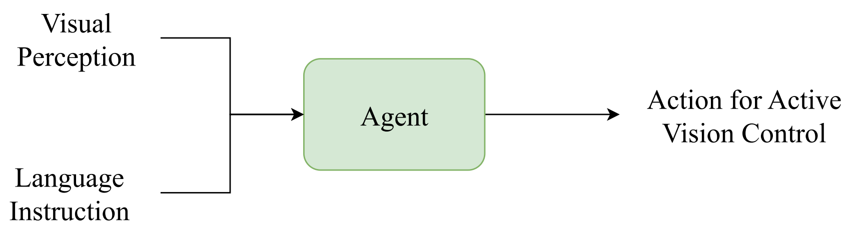


Figure 1.1: Diagram of the *action-only embodied agents* with the single-output channel of active vision control. This framework is applicable in tasks where only the control of the active vision system is enough to solve the task, e.g., in the task of object-goal navigation [Anderson et al., 2018a].

output channels: one for low-level active action control aimed at goal-driven visual information gathering, and another for high-level, task-relevant responses for accomplishing a specific task goal (see Fig. 1.2). This setup closely resembles how humans operate in their daily routines. For example, to find out the number of chairs in a room, we first search for chairs in the room and then give the number. In order to figure out the amount of money in a wallet, we need to examine the wallet's content and count the notes and coins to arrive at the total sum. Throughout the active vision exploration process, we gradually reduce our uncertainty regarding possible answers and eventually come up with a final response with a high level of certainty. This dual-output setup is more complex and presents greater challenges compared to tasks where the agent only produces active vision control actions. Typical tasks involving this type of agent include embodied question answering [Das et al., 2018], embodied amodal recognition [Yang et al., 2019a], and robotic occlusion reasoning [Li et al., 2021]. We introduce the term *action-response embodied agents* to refer to embodied agents with dual-output channels for action and response, and *action-only embodied agents* to those possessing a single-output channel for action. This thesis concentrates on action-response embodied agents due to their wider application potential and the relatively lesser extent of exploration they have received in existing research.

Consider the following real-life scenario involving an action-response human assistant, to better understand the issue of insufficient observations and the impact of active vision behaviors on the resultant responses. How would an assistant respond to a request "What's the weather like today?" from an elderly person with limited mobility or a visually impaired person? A responsible and diligent assistant may head over to the windows of the room to visually assess the weather conditions. In case the windows of the room are obscured by curtains, the assistant may open the curtains and then perform the assessment. However, an irresponsible assistant may simply give a random guess about what the weather is like, e.g., "The weather is good!", instead of actively exploring the environment to collect further necessary observations to assess the weather conditions. This response could be wrong and misleading, thus potentially harmful for the questioner if the weather is actually cloudy and is going to rain soon. It could also happen that the assistant

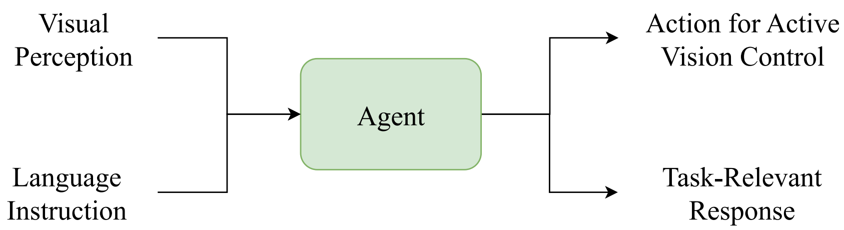


Figure 1.2: Diagram of the *action-response embodied agents* with dual-output channels: the output of the active vision control channel is aimed at collecting sufficient information for accomplishing the given task, and the output of the task-relevant response channel is to produce the final response utilizing the information gathered through active vision. This framework is applicable in cases where *the active vision system works as the support component for a high-level cognition task*, e.g., in the task of embodied question answering [Das et al., 2018].

is lazy and prefers to avoid mistakes by simply responding “I don't know”. This response is better than the random guess from the perspective of the questioner since it is at least not misleading. However, this response is unarguably not helpful. The first type of response is what we expect from an assistant since it tackles the issue of insufficient observations well through reasonable environmental interaction and exploration. To address the challenges posed by insufficient observations, an ideal human assistant, as well as an action-response embodied agent, should possess the following capabilities to produce rapid and reliable responses:

1. Evaluating the sufficiency of observed information for achieving the given task;

2. Producing immediate responses in case the observed information is sufficient;

3. Interacting with the environment to actively acquire additional task-relevant information in case the observed information is insufficient.

However, endowing action-response embodied agents with such capabilities to tackle the issue of insufficient observations is nontrivial. As the active vision policy is goal-oriented, the agent is expected to have a good understanding of both the goal of the task and the environment to evaluate information sufficiency and perform both high-level action planning and low-level interaction control. A wide range of techniques must be employed, encompassing natural language understanding, image processing, common-sense reasoning, action policy learning, world model acquiring, and so on.

In this thesis, we attempt to model action-response agents using *unified modular neural networks*, and develop the active vision control policy through *reinforcement learning*. This is biologically plausible, as the components of sensory perception, attention, and value evaluation are all involved in processing rewards in the brains of humans and animals [Schultz, 2015] (cf. Chapter 3.1). Different from existing works focusing on large-scale realistic environments and agents that are expected to integrate a wide range of capabilities, this thesis is not aimed at solving the challenging problem in scale. Instead, we specifically focus on the active vision control policy and its collaboration with the task-relevant response. By working on small-scale environments, we intentionally set aside distractions from other capabilities, such as commonsense reasoning and world modeling. Although the experimental environment used in this thesis is constrained, this research can provide valuable insights regarding the development of action-response agents in more realistic and large-scale environments.

1.2 Research Questions

To achieve the aforeintroduced goal, this thesis is dedicated to addressing the following research questions.

- How to endow a disembodied model with the capability of information sufficiency evaluation?
- How to model action-response embodied agents using neural networks and optimize the active vision control policy using reinforcement learning?
- How to stabilize the reinforcement learning process of the active vision control policy to make the training more efficient?

1.3 Research Methodology

To answer the first research question, we study the task of irrelevant visual question detection, where a model is tasked to detect visual questions that are unanswerable. This task is closely related to the task of visual question answering, which operates under a precondition that all the visual questions are answerable. To perform the task, a model should be able to detect the sufficiency of the visual information given a visual question. We employ a model architecture that was originally designed for the task of visual question answering and investigate whether the abilities for information sufficiency detection are aligned with the abilities for visual question answering. Furthermore, we endeavor to obtain an integrated model that is capable of both visual question answering and irrelevant visual question detection, which can prevent the generation of potentially harmful answers when the visual information is insufficient while generating answers when the visual information is sufficient. To accomplish this objective, we adopt a strategy of training a model on an assembled dataset containing both relevant and irrelevant image-question pairs, along with an additional answer option "irrelevant" that indicates unanswerable questions.

To answer the second research question, we adopt the methodology of recurrent visual attention [Mnih et al., 2014], which is a seminal work in the attention mechanism in the realm of computer vision. An architecture named Recurrent Attention Model (RAM) is designed for the image classification task. Align with the

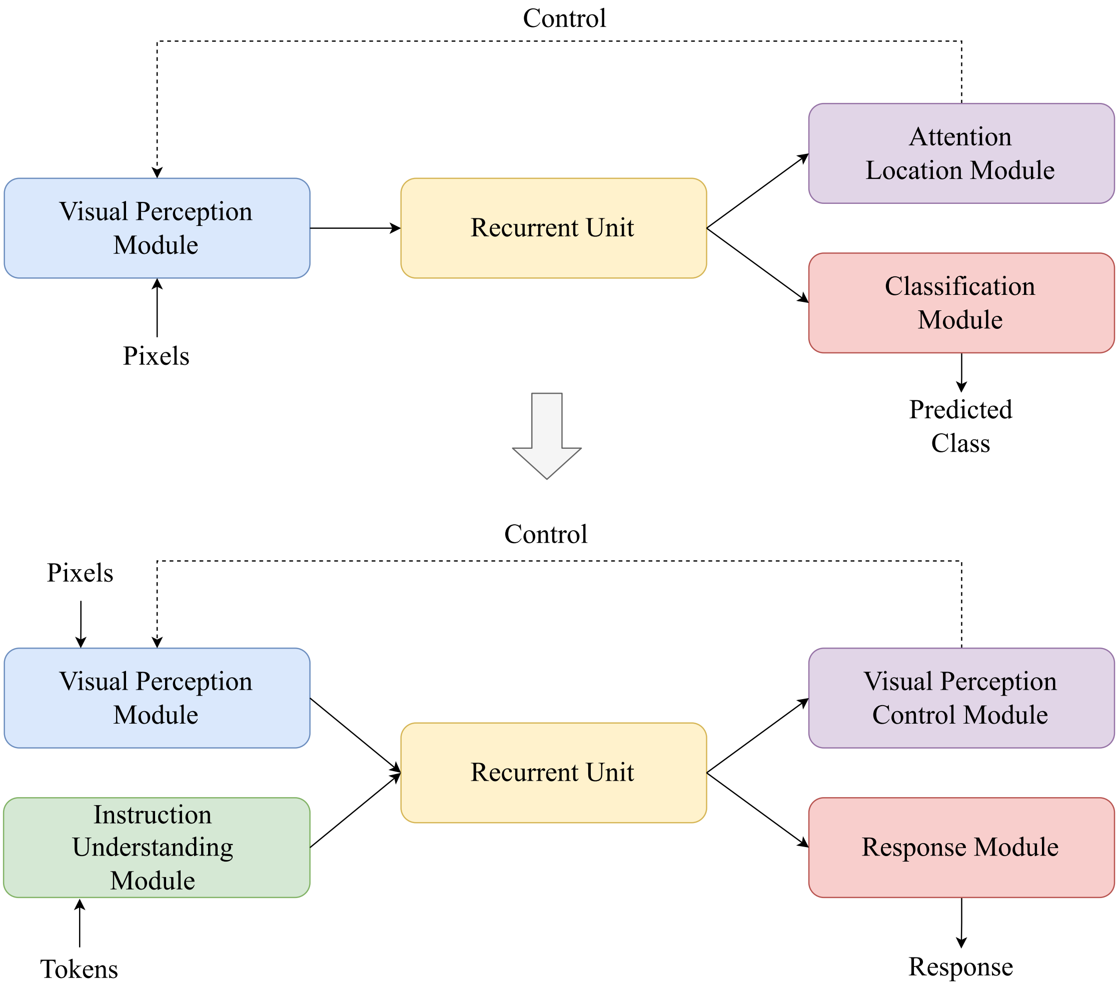


Figure 1.3: From the framework of the Recurrent Attention Model [Mnih et al., 2014] to the framework of the action-response agent. **Top**: high-level framework of the recurrent attention model; **Bottom**: high-level framework of the action-response agent with active vision.

framework of action-response agents (cf. Fig. 1.2), RAM has two distinct output modules: one for attention action generation and the other for image classification. RAM can only perceive a small portion of pixels of the whole image with a limited field of perception at each time step. Nevertheless, the location of the perception field, controlled by a location network, can be changed at each time step to collect visual information sequentially. The location network is trained in reinforcement guided by feedback from the classification network, and the classification network is trained in supervised learning using the human-annotated digit classes. By recurrently processing a sequence of partial observations, RAM classifies digits in images. We aim to adapt the model architecture and training methods of RAM in the development of action-response embodied agents, see Fig. 1.3.

In addition, to evaluate the effectiveness of the proposed method, we create the task of robotic object existence prediction, where a robot with an egocentric view can move around a table to predict the existence of a target object. This is a small-scale and controllable task with a focus on the active vision capability of the embodied agent. Compared with existing tasks in large-scale environments, this task is more suitable for evaluating the capability of active vision in action-response agents.

To answer the third research question, we generalize the reinforcement learning of active vision control policy learning into a framework named internally rewarded reinforcement learning, where the reward for reinforcement learning is from a discriminator that is jointly optimized with the policy. We study the issue of the unstable training loop in the training process of internally rewarded reinforcement learning and investigate the impact of reward functions on the instability of the training process.

1.4 Contributions of the Thesis

In this thesis, we highlight the importance of addressing the issue of insufficient observations for embodied agents. We attempt to develop embodied agents with an active vision that are able to handle these situations. Specifically, we focus on action-response embodied agents, which have two output channels: one for the active vision control and the other for the task-relevant response that depends on the visual information collected through active vision.

As a preliminary study, we first work on disembodied models in scenarios of insufficient observations. Specifically, we study the capacity to detect the sufficiency of visual information in the visual question answering task. We demonstrate the feasibility of endowing existing visual question answering models with the capacity to detect information sufficiency and prevent producing potentially harmful responses in situations with insufficient visual information through training the model on datasets containing samples with both sufficient and insufficient visual information. We also demonstrate the limitations of this approach by showing that the performance of the resulting model degrades on samples with sufficient information, in contrast to a model trained exclusively for scenarios with sufficient information. This study underscores the challenges of employing supervised learning to endow a model with the capability of detecting information sufficiency. It also suggests the necessity of deploying embodied agents with active vision in realistic application scenarios.

To facilitate the evaluation of the performance of action-response embodied agents focusing on the active vision ability, we design a task with a reasonable difficulty, namely robotic object existence prediction (ROEP), and develop its corresponding simulation environment and dataset. We propose an end-to-end modular network to model the active-vision agent and successfully train the agent using supervised learning and reinforcement learning using the curriculum training strategy. Experimental results on the ROEP task suggest the effectiveness of the proposed agent.

Based on the finding that the successful convergence of the proposed agent is difficult to achieve, we analyze the training procedure in depth and generalize the learning paradigm to a class of reinforcement learning problems. We prove empirically and theoretically that the shape of the reward function has an impact on the stability of the training process, based on which we proposed the clipped linear reward function to mitigate the unstable training issue.

To the best of our knowledge, this is the first research that systematically studies the learning of the active vision control policy for action-response embodied agents from the perspective of unified modeling and reinforcement learning.

1.5 Structure of the Thesis

In Chapter 2, we briefly introduce the background in embodied agents with active vision, especially related applications and approaches to learning the active vision policy. Next, in Chapter 3, we introduce fundamental concepts of reinforcement learning, with a particular focus on the generation of rewards for reinforcement learning, including the neuroscientific theory of reward processing for humans and animals, and the techniques of reward function design from a computational intelligence background. Then, in Chapter 4, we work on the issue of insufficient observations on the VQA task, where the issue comes from irrelevant visual questions. We attempt to mitigate the issue in a supervised learning approach by augmenting the dataset with additional samples of irrelevant questions and the corresponding answer "I don't know". In this work, we aim to answer the questions "How to enhance disembodied models that inherently lack active vision to address the issue of insufficient observations?" and "Is active vision an essential component to tackle the issue of insufficient observations?". Next, in Chapter 5, we turn to embodied agents that have the capability of actively collecting information to respond correctly. We propose an end-to-end recurrent neural network that learns the active action policy and the task-relevant responding strategy simultaneously. To evaluate the performance of the proposed method for learning active vision policy, we design an evaluation task, named "robotic object existence prediction". Then, in Chapter 6, with the goal of improving the stability of the training process, we delve into the method introduced in Chapter 5 and generalize it as a class of reinforcement learning problem named "internally reward reinforcement learning", which is applicable in a variety of problems. In the end, in Chapter 7, we discuss the contributions and future work of this thesis.

Chapter 2

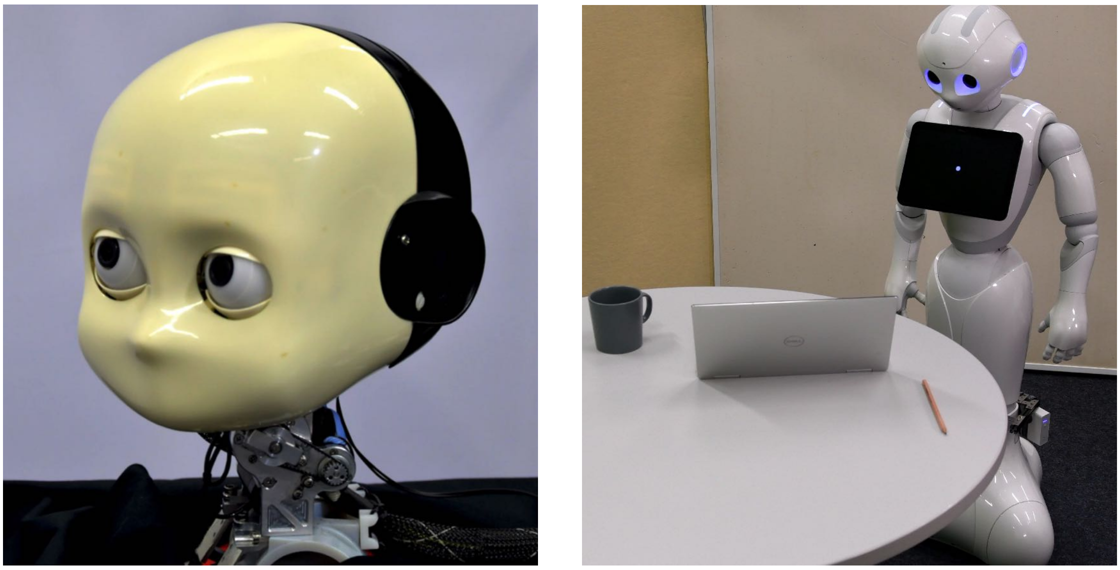
Embodied Agents with Active Vision

2.1 Introduction

Given the achievements and success of disembodied AI methods on static datasets, the enthusiasm for extending the application area of these methods into a wider range of scenarios spurs researchers to focus on embodied AI agents, which are intended to integrate AI models more closely with humans in the physical world and enable them to autonomously interact with the environment.

Embodied agents are intelligent entities that have a physical or virtual presence, which is always robotic platforms or simulated avatars, in realistic or virtual environments [Duan et al., 2022]. Key characteristics of embodied agents include the capability of *physical interaction* with the environment and *active perception* of the environment. Embodied agents are expected to have automatic control of their physical interaction and active perception behaviors. The learning of embodied agents is based on the theory of embodied cognition [Heidegger, 1988, Merleau-Ponty and Smith, 1962, Varela et al., 2017], which is a theory in psychology and cognitive science. It suggests that the interaction between the intelligent entity and its environment plays a crucial role in cognition development and shaping intelligence, challenging traditional cognition theories that the physical body is independent of the mind. Driven by the theory of embodied cognition, embodiment learning in the realm of artificial intelligence attempts to learn perception, cognition, decision-making, and other intelligent capabilities through the physical interaction of embodied agents with the environment.

Active vision [Aloimonos et al., 1988], or more generally active perception [Bajcsy, 1988], is a primary aspect of embodiment learning. We humans are able to achieve active vision through saccades (moving eyes to change the point of fixation) and whole-body movements (moving the body to reach novel viewpoints). To achieve active vision control of artificial embodied agents, they learn control policies for visual sensor states, such as position, orientation, and focus, to actively collect visual information through subtle sensor or whole-body movements. Ac-



(a) Eyes of the iCub robot head can move flexibly to achieve saccades.

(b) Pepper robot can change its viewpoint through head and whole-body movements.

Figure 2.1: Various robotic active vision mechanisms based on different robot hardware.

tive vision is closely linked with embodied agents and robotics [Chen et al., 2011], where the physical body of the robot serves as a dynamic platform for the camera, granting it the capability of active movement. Fig. 2.1a shows the iCub robot head[1], whose eyes can move flexibly, mimicking human eyes, to actively change its gaze point. Fig. 2.1b shows the Pepper robot[2], which can perform active vision through head and whole-body movements. Due to the growing trend of developing embodied AI agents, active vision has gained increasing attention in recent years.

Passive vision is a concept opposite to active vision, where the vision system passively processes the visual input fed by the users. In contrast to passive vision, active vision is more analogous to the vision system of humans and animals, where the field of view is adjusted to collect information driven by the goal of achieving specific tasks. Research in computer vision has been dominantly focusing on passive vision using static image datasets over the past few decades, e.g., ImageNet [Deng et al., 2009], MS COCO [Lin et al., 2014], and Visual Genome [Krishna et al., 2017]. Researchers were dedicated to developing models and approaches to improve the performance of computer vision systems on such datasets on tasks including image classification [Krizhevsky et al., 2012], object detection [Girshick et al., 2014], semantic segmentation [Ronneberger et al., 2015], etc.

Embodied agents that are highly generalizable to the complex real world have a prosperous impact on human society and civilization. They can make humans' lives easier and better, for example, autonomous driving vehicles can relieve humans

[1] https://icub.iit.it/

[2] https://www.softbankrobotics.com/emea/en/pepper

from the time-consuming and energy-draining task of driving while also enhancing the safety of everyday transportation, versatile household support robots can free humans from repetitive daily chores, and elderly assistance robots can improve the living standards of the seniors in societies facing a labor shortage. On the other hand, embodied AI agents can also fulfill tasks that exceed humans' capabilities, e.g., completing tasks in environments with extreme heat or potential collisions, where the carbon-based human body is vulnerable to injury. The capability of active vision and active information acquisition plays an essential role in such embodied agents. Though building embodied agents with active vision is very challenging, it is one of the long-standing goals of both the artificial intelligence and robotics research communities.

2.2 Tasks of Active Vision

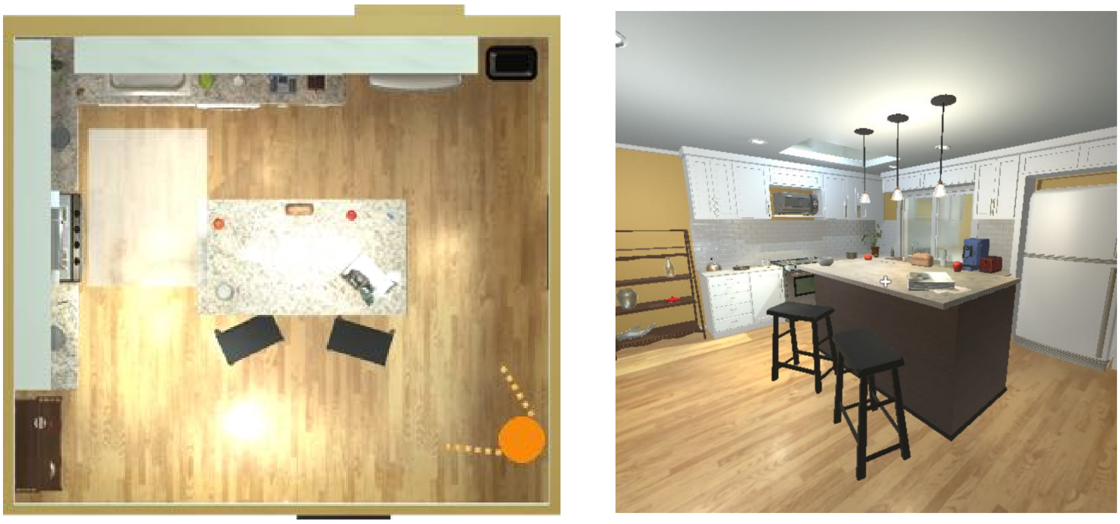
Active vision systems, such as autonomous vehicles, indoor-assistant robots, industrial robots, and unmanned aerial vehicles, are already widespread, and it is anticipated that they will become even more prevalent with the rapid progress in artificial intelligence. Powered by the capability of active vision, these hardware platforms can perform a range of challenging tasks, such as object and site modeling, manipulation, autonomous navigation, tracking, surveillance, and recognition [Chen et al., 2011]. In this section, some representative applications will be introduced.

2.2.1 Visual Navigation

Visual navigation is a straightforward application of active vision [Zhu et al., 2017, Huang et al., 2023, Ramakrishnan et al., 2022]. In visual navigation, the agent navigates the environment primarily based on its visual perception, mimicking the navigation strategy of humans and animals. In contrast to laser-based navigation, visual navigation is more versatile due to the incorporation of high-level semantic information obtained from visual understanding. Visual navigation remains a very challenging task, particularly in extensive and realistic environments. Accomplishing generalizable visual navigation demands a superior level of visual understanding, memory, world modeling, common-sense reasoning, high-level action planning, and low-level interaction with the environment (cf. Fig.2.2).

Several simulation environments have been developed to facilitate research in this area. AI2Thor [Kolve et al., 2017] is a simulation environment of indoor scenes. It is built utilizing the Unity game engine[3], which enables physical simulations and provides near photo-realistic rendering. AI2Thor provides 120 simulated indoor scenes, covering four types of rooms: bathrooms, bedrooms, kitchens, and living rooms. These indoor scenes are designed by expert 3D designers to mimic real-world room layouts and conditions. It offers extensive flexibility in controlling environmental variables, such as light conditions, object materials, colors,

[3] https://unity.com



(a) Top-down view

(b) First-person view

Figure 2.2: An example kitchen scene in the AI2Thor simulator [Kolve et al., 2017]. The scene is designed by mimicking realistic human living environments. Visual navigation in this kind of realistic environment is challenging. For example, in the task of searching for a kitchen sponge, given the above first-person view, the agent needs to first navigate to the stove area, and then to the sink area to perform a more detailed search, driven by the common-sense knowledge about the layout of a common kitchen and relative locations between objects, i.e., a sink is always close to the stove and a kitchen sponge is always placed around a sink. During the navigation process, the agent also needs to avoid collision with the furniture for safety and avoid retracing explored areas for efficiency.

positions, and states. Besides visual navigation, this simulation environment also supports research in a large spectrum of other fields, such as embodied question answering [Gordon et al., 2018], and language grounding [Zellers et al., 2021]. Fig.2.2 shows views of an example kitchen room of the AI2Thor simulator. iGibson [Shen et al., 2021, Li et al., 2022] is a simulation environment based on the physics engine Bullet[4]. Similar to AI2Thor, iGibson provides visual rendering and physical simulation. It contains 15 fully interactive home-size scenes with 108 rooms. In contrast to the human-designed indoor room environments in AI2Thor, scenes in iGibson are directly reconstructed from homes and offices in the real world.

Anderson et al. [2018a] defined three types of visual navigation tasks: point-goal navigation, object-goal navigation, and area-goal navigation. In the point-goal navigation task, an agent is instructed to navigate to a location specified by the relative coordination to the agent in unseen environments, e.g., "Go 7 meters south, 5 meters west of you". This task is nontrivial when the environment is empty, however, it is challenging for the agent to navigate efficiently and safely in unseen and realistic environments populated with furniture and objects. This task

[4] https://pybullet.org

has been recently successfully solved with an almost perfect (99.9%) success rate in various simulation environments by an approach using RGB-D, GPS, and compass data [Wijmans et al., 2020]. The success of this method is largely due to a variant of the Proximal Policy Optimization (PPO) algorithm [Schulman et al., 2017], which is named Decentralized Distributed Proximal Policy Optimization (DD-PPO). This method exhibits a notable property in scaling, where the performance of the agent continuously improves with more training steps.

In the object-goal navigation task, an agent is instructed to navigate to an object of a category specified by its name, e.g., “laptop”. To achieve this task efficiently, the agent needs to learn the common appearance and locations of the target object, and also the exploration strategy based on its observations. This task is more challenging compared to the point-goal navigation task since there is more semantic understanding and common-sense knowledge needed to effectively perform this task. The straightforward idea of applying end-to-end RL-based methods has been proven ineffective in this task, despite its success in point-goal navigation. Researchers are attempting to address this task from the perspective of incorporating *explicit memory* and *extra prior knowledge* into the agent.

Given that the agent can only partially observe the environment, memory plays an essential role in efficient navigation. It helps the agent to explore the environment efficiently, for instance, avoiding retracing areas that are already explored. While certain model architectures, like LSTM, have built-in memory components, the addition of external and task-specific memory can further enhance navigation performance. There are mainly two types of task-specific memory structures used for navigation tasks: spatial memory and topological memory [Wu et al., 2019]. Spatial memory represents the spatial structure of the environment. In addition to geometric data, semantic information can also be encoded into spatial memory to build semantic maps [Chaplot et al., 2020, Huang et al., 2023]. As spatial memory saves comprehensive information about the environment, it facilitates long-term planning of the navigation trajectory. However, the size of spatial memory and the required computation demands increase with the size of the environment. In contrast to spatial memory, topological memory encodes the environment in topological graphs, where only abstract information about the environment is encoded, such as landmarks and relations between objects [Wu et al., 2019]. The use of topological memory is biologically inspired – rather than memorizing precise spatial maps for navigation, humans primarily depend on the relative relation between landmarks [Foo et al., 2005, Wang and Spelke, 2002]. Fig. 2.3 demonstrates an example of topological memory represented by a graph.

Commonsense knowledge, e.g., common relative positions between objects, plays an important role in semantic navigation, especially in unseen environments. For example, a garbage bin is always located in a room corner, a laptop is commonly on a table, and a pillow is always placed on a bed. Numerous studies have focused on incorporating such commonsense knowledge into models for object-goal navigation. Some of them explore knowledge graphs to encode prior knowledge about relative spatial relations between objects and incorporate this knowledge into the agent through the usage of graph neural networks Yang et al. [2019b], Wu et al.

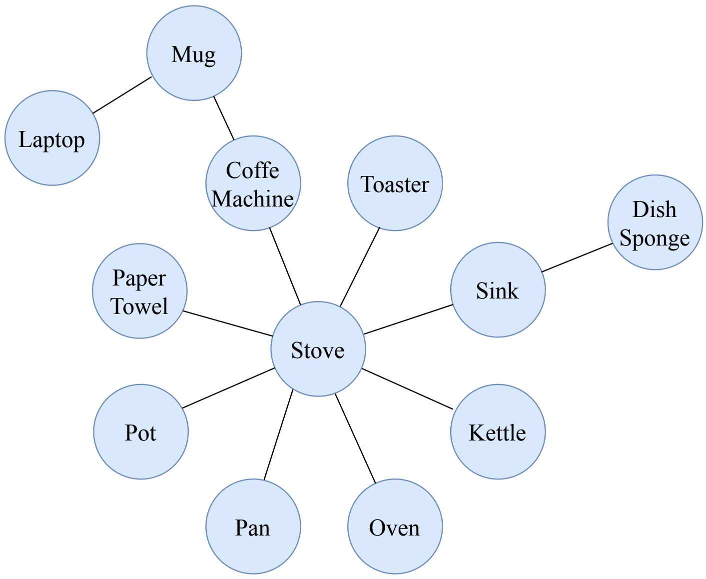


Figure 2.3: An example graph illustrating the topological memory of relative spatial relations between objects with edges representing the semantic concept "close to".

[2019]. Motivated by the assumption that rich commonsense knowledge and world models have been already encoded in pretrained large language models (LLMs) during the large-scale pretraining, LLMs are recently studied to facilitate commonsense reasoning and high-level action planning in the visual navigation task [Ramakrishnan et al., 2022, Huang et al., 2023, Zhou et al., 2023]. In Appendix A, we explore the use of commonsense knowledge encoded in LLMs to help robotic action planning in occlusion reasoning scenarios.

2.2.2 3D Reconstruction

Another interesting and growing application of active vision is 3D reconstruction, a field that lies in the intersection between computer vision, computer graphics, and robotics. 3D reconstruction is a process of capturing the appearance of objects or scenes in the real world and converting them into digital 3D models. It is useful in a variety of domains. In the entertainment industry, it is employed to generate digital assets for virtual reality and video games. In the area of architecture, it helps in the creation of building models [Zhou et al., 2020, Kompis et al., 2021]. In the field of archaeology, 3D reconstruction technology helps digitize and preserve cultural heritage [Gomes et al., 2014]. 3D reconstruction also aids in creating 3D spatial maps for robotic and autonomous vehicle navigation, supporting the application of visual navigation that has been introduced before [Adamkiewicz et al., 2022, Kwon et al., 2023].

Conventional 3D reconstruction relies on dense 2D views. However, the consumption of time and energy required to sample dense views may not be allowed in resource-constrained scenarios, for example, in the case of reconstructing a large

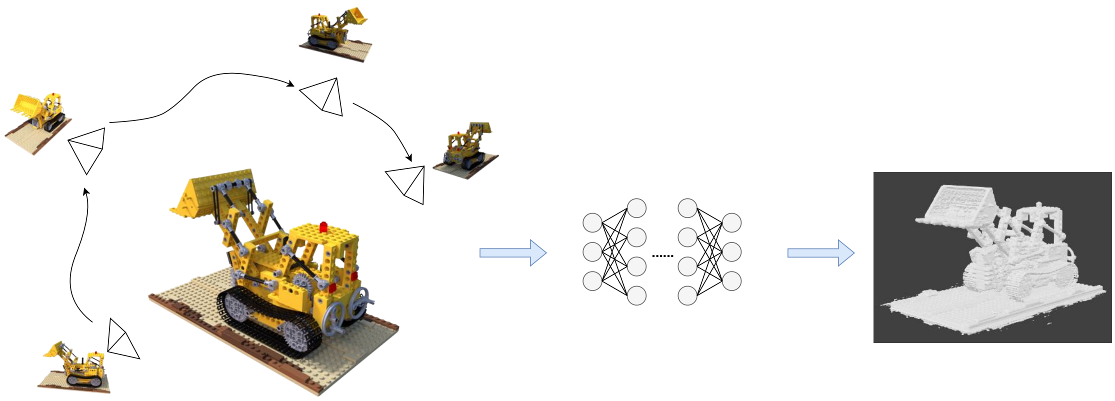


Figure 2.4: An illustration of active vision for 3D object reconstruction. An active vision control policy selects the next-best view sequentially, instead of sampling views densely and uniformly, to capture views of the object. Based on the selected views, a reconstructed digital model of the object is generated by deep neural networks. The exampel object is from the NeRF dataset [Mildenhall et al., 2020].

building in minutes using a drone [Zhou et al., 2020, Kompis et al., 2021]. With active vision, the viewpoint and focus of the camera can be dynamically adjusted, through the control of a mobile robot, a robotic manipulator, or a drone, to optimize the trade-off between efficiency and quality of data collection for 3D reconstruction. The goal of active vision control is always described as the selection of the Next-Best View (NBV).

Object reconstruction is a typical application in the field of 3D reconstruction. In the task of active object reconstruction, the camera is controlled by a mobile robot or a robotic arm to actively select views for the target object. Existing work in object reconstruction can be categorized into two general classes: *explicit 3D geometric modeling* and *implicit neural representation learning*. Explicit 3D geometric modeling directly creates 3D models of objects using geometric primitives like points, polygons, or voxels, while implicit neural representation learning leverages deep neural networks to learn neural representations of 3D objects and explicit 3D models can be obtained using the obtained representations through a separate conversion procedure [Rakotosaona et al., 2023]. Fig. 2.4 demonstrates the task of active object 3D reconstruction.

Explicit 3D geometric modeling with active vision has been studied for a long time. Isler et al. [2016] considered the problem of selecting the next-best view for efficient volumetric reconstruction of a target object as a task of information gain maximization. They proposed a set of formulations for evaluating the amount of information gain from a set of candidate views based on an online constructed probabilistic volumetric map. The next-best-view selection policy can be then directly obtained according to the information gain formulations. Deep learning approaches have been studied for this task in recent years. For instance, Yang et al. [2018] proposed a unified deep learning model for view planning and object reconstruction. This model is a modular neural network, inspired by the recurrent attention neural network [Mnih et al., 2014]. Specifically, a 3D decoder module, which generates 3D volume, is trained in supervised learning to minimize the difference between the predicted and the ground-truth volume. A view planner, which predicts the locations of a sequence of informative and discriminative views, is trained in reinforcement learning using a reward function that considers the reconstruction accuracy, 3D to 2D projection accuracy, and a penalty on the movement distance.

Neural Radiance Fields (NeRF) [Mildenhall et al., 2020] is a recent technology in neural rendering. It learns *implicit neural representations* of 3D objects and scenes, which can be used to render highly realistic views from arbitrary viewpoints and generate 3D models. This technique showcases impressive capability in realistic 3D modeling. Compared to explicit representations like points and voxels, implicit neural representations offer advantages including a lower memory footprint and higher precision in reconstruction.

Conventional NeRF relies on a large amount of images and corresponding camera states. With a limited number of sampled images, NeRF struggles to model the target object in a satisfactory quality [Yu et al., 2021]. Active vision for NeRF seeks to reduce the number of images needed for learning high-quality neural representations by actively sampling informative images. This idea has attracted a lot of attention recently [Pan et al., 2022, Ran et al., 2023, Lee et al., 2022, Yan et al., 2023]. However, it is challenging to select the next-best view using implicit neural representations because it is not straightforward to evaluate the importance of a given view image in 3D modeling. To tackle this problem, Pan et al. [2022] borrow ideas from active learning and propose to incorporate uncertainty estimation into NeRF for informative view selection. Specifically, NeRF predicts the value of uncertainty estimate along with occupancy and color. Align with the method of Pan et al. [2022], Lee et al. [2022] estimate the uncertainty of 3D modeling given a novel view. Specifically, they infer the uncertainty of the 3D modeling using the entropy of the weight distribution of the color samples along each ray of the object's implicit neural representations. A next-best-view selection policy is then obtained under the guidance of the uncertainty estimate. Similarly, to evaluate the uncertainty of candidate views, Yan et al. [2023] proposed to directly calculate entropy from the reconstructed occupancy probability field. Ran et al. [2023] also attempt to measure the importance of candidate views by evaluating the view quality using a proxy criterion of Peak Signal-to-Noise Ratio (PSNR), a commonly used metric in the field of image and video processing to qualify the quality of lossy compression.

2.2.3 Visual Attention

Visual attention has a significant influence on the research on the active vision of embodied agents, especially the mechanism of saccadic active vision. Hard attention and soft attention are two primary attention mechanisms used in computer vision. In contrast to soft attention, where the model takes the whole image as the input and assigns different attention weights on different regions, hard attention only processes a subset of the whole input data at each time step and observes other parts by actively changing the location of the attention. Though soft attention is more commonly used mostly due to the simplicity in integrating it with various neural network architectures, hard attention yields advantages in scenarios with limited computation resources, for example, in the task of image classification for high-resolution geometrical satellite images [Wang et al., 2019, Rangrej et al., 2022]. It also benefits situations where focusing on specific regions of the input is particularly important, for example, in tasks of fine-grained image classification, such as the classification of bird species, where detailed information is crucial for accurate classification [Ba et al., 2015, Li et al., 2017, Liu et al., 2016, Fu et al., 2017]. Through the hard attention mechanism, the model can attend to areas containing subtle task-relevant features to perform high-accurate fine-grained classification.

Hard attention has a close relationship with cognition science, as it is analogous to human attention mechanisms [Rong et al., 2021, Das et al., 2017]. Both hard attention and human attention enjoy a selective characteristic, namely, the ability to focus on specific parts of the perception field while ignoring others, thereby enabling efficient allocation of computation and cognition resources. Due to the close connection between these two research areas, insights and findings from one field can inform and improve the other.

2.3 Active Vision Control

There are three primary approaches to developing task-driven active vision control (see Fig. 2.5). The most long-studied and well-established approach is *information gain-driven active vision control* [Isler et al., 2016, Kompis et al., 2021, Chen et al., 2022]. This approach is based on the concept of information gain maximization, which is often used in decision-making processes. The idea is to make action decisions that provide the most informative view, to most effectively reduce the uncertainty of performing the given task. However, this approach heavily relies on a well-defined and task-specific evaluation metric for information gain and thus suffers from low generality. In addition, semantic information is hard to be involved in this class of methods where low-level representations are processed in information gain evaluation, which limits the application of these methods.

Reinforcement learning-based active vision control directly learns the policy by optimizing a surrogate objective that is represented by a reward function [Mnih et al., 2014, Yang et al., 2018, 2019a, Wijmans et al., 2020, Li et al., 2021]. This approach is particularly effective when the surrogate objective represented by the reward function is close to the true objective [Schulman et al., 2015a], for example in the task of visual navigation. This method has demonstrated effectiveness in learning active vision control in various tasks, including point-goal navigation [Wijmans et al., 2020], 3D reconstruction [Yang et al., 2018]. However, this approach is not always applicable because it is not feasible to obtain reliable reward

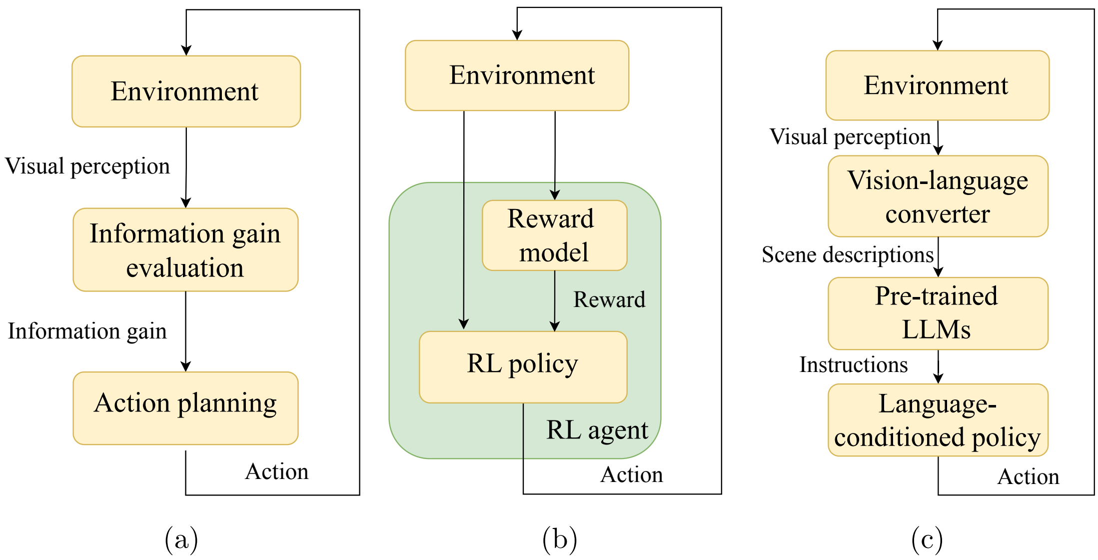


Figure 2.5: Three primary approaches to active vision control. From left to right are diagrams of (a) *information gain-driven*, (b) *reinforcement learning-based*, and (c) *large language model-based* methods, respectively.

signals in certain tasks, such as in tasks of embodied question answering, and embodied amodal recognition [Das et al., 2018, Yang et al., 2019a]. What's more, the issue of unstable training in reinforcement learning is still an open problem [Greensmith et al., 2004, Schulman et al., 2015b, van Hasselt et al., 2016, Li et al., 2023], increasing the barrier to applying this approach in reality. Compared with information gain-driven approaches, reinforcement learning-based approaches can involve more high-level semantic information in the system through the functionality of the reward model, which can guide the RL policy learning with high-level semantic objectives.

Another approach that is attracting growing attention is *large language models-based active vision control* [Ahn et al., 2022, Driess et al., 2023, Zhao et al., 2023, Yu et al., 2023]. These approaches take advantage of commonsense knowledge in pretrained LLMs or pretrained multimodal language models for active vision control. These approaches demonstrate great generalization in complex realistic scenarios. However, existing work mainly concentrates on utilizing commonsense knowledge in pretrained LLMs for high-level planning, rather than for low-level action control. It is because low-level interaction action is highly dependent on the physical properties of the hardware, which is not likely learned in pretrained LLMs. A promising approach is integrating reinforcement learning-based methods and LLM-based methods by taking advantage of both of them: reinforcement learning learns low-level interaction policies that are highly dependent on the physical environment and the pretrained LLM guides high-level planning and helps to accelerate the process of reinforcement learning [Chu et al., 2023].

2.4 Action-only and Action-response Agents

As previously introduced (cf. Fig. 1.1 and Fig. 1.2), we categorize embodied agents with active vision into two classes: action-only agents and action-response agents, based on the role of their active vision systems. Action-only agents are those operating in tasks where learning the active vision control policy of the agent itself is the primary objective, e.g., in the task of point-goal navigation [Anderson et al., 2018a, Wijmans et al., 2020], where the active vision control policy is essentially a navigation policy. Agents for various visual navigation tasks are action-only. Besides that, agents designed for tasks such as target tracking, goal searching, robotic grasping and manipulation with interactive exploration, and environmental monitoring are also action-only.

Action-response agents are those that operate in scenarios where the active vision system serves as an information collector for gathering relevant data, with the ultimate objective of the task being to produce responses based on the processing of this data, e.g., in tasks of embodied question answering [Das et al., 2018], and embodied amodal recognition [Yang et al., 2019a]. Besides, the agent for tasks such as scene classification [Wang et al., 2019], scene description [Tan et al., 2020], 3D reconstruction [Yan et al., 2023], and interactive question answering [Gordon et al., 2018] also belong to this category, where the agent needs to learn both an active vision control policy and a task-specific response strategy.

Compared to action-only agents, action-response agents appear to have wider applications in human-centered real-life scenarios but are more challenging. The accomplishment of these tasks depends on a combination of skills, for example, in the embodied question answering task, the agent needs to understand complicated questions in natural language, perform both long-horizon action planning and low-level interaction in indoor environments to collect necessary question-relevant visual information, and generate answers according to collected visual information in the end. Existing action-response agents struggle to achieve these tasks, for example, it has been demonstrated that existing embodied question answering models learn biases to accomplish the task [Anand et al., 2018, Ilinykh et al., 2022]. Thus, these tasks are not suitable for research primarily focusing on the methodology of developing active vision control policies.

2.5 Discussion

In this chapter, we introduced some background on the research area of active vision for embodied agents and relevant topics. It can be seen that this direction is rooted in existing research on robotics, computer vision, and artificial intelligence, and is developing rapidly as the advances in technologies such as deep learning, reinforcement learning, physical simulation, and computer graphics. Within the broad research field of active vision for embodied agents, this thesis particularly focuses on the study of methodologies for developing the active vision control policy for action-response agents using reinforcement learning-based approaches.

Chapter 3

Reinforcement Learning for Behavior Control

3.1 Introduction

How do humans or other animals learn their behaviors? Learning from demonstrations of a supervisor seems a straightforward way to behavior acquisition, however, learning from trial and error through interacting with the environment seems a more natural paradigm, which is termed *reinforcement learning* (RL). Humans and animals adopt this strategy commonly to learn behaviors. They receive *rewards* or *penalties* from the environment, depending on their interaction behaviors with the environment and their subjective preference and goals in mind. These feedback signals are captured by their sensors and are processed by their brains, and further used to optimize their behaviors for more reward acquisition.

From the perspective of neuroscience, rewards or penalties are captured by sensations (e.g., visual observations, tactual sensations, etc.), then operate in neural processing, leading to the generation of chemical messengers, such as dopamine, and ultimately lead to the emergence of behaviors learning, decision-making, etc. Animals gain the ability to improve their behavior strategies from rewards, and those who are better at reward acquisition survive as a result of natural selection [Darwin, 1859].

In a formal formulation by Schultz [2015], the rewarding process of reinforcement learning consists of three components: sensory components, attentional components, and value components, see Fig. 3.1. *Sensory components* of rewards involve visual, auditory, and gustatory sensations, among others, reflecting the impact of the external environment on the organism. They are generated through sensory perception by identifying stimuli and objects, lying as the foundation of rewards. *Attentional components* of rewards arise because rewards come from only a part of the entire environment. An attention mechanism is triggered to make the reward processing efficient. *Value components* of rewards evaluate the positive effects of rewards. The evaluation value is not determined by objective or physical facts but reveals the brain's subjective evaluation of the effects of goal achieve-

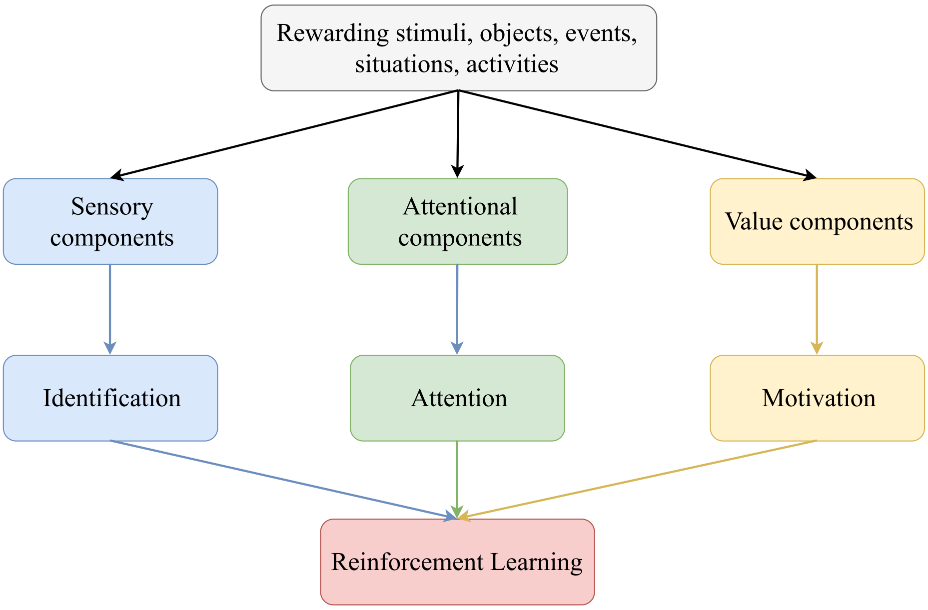


Figure 3.1: Three components of the rewarding process in reinforcement learning. All reward components operating together lead to the occurrence of the reinforcement learning process for reward maximization. This figure is based on Schultz [2015].

ments, such as survival, reproduction, or obtainment of essential substances. According to whether the aforementioned reward components are determined by the environments or brains, they are categorized into external or internal components. Sensory components are external as sensory sensations are directly stimulated by environmental feedback. Attention components can be either external or internal, depending on what causes attention. For example, attention triggered by the physical properties of objects is external (e.g., sparkling objects or sharp sounds tend to raise attention), while attention triggered by surprise is internal. Value components are considered internal as value evaluation depends on the subjective goals of the individual.

The rewarding process for reinforcement learning is commonly simplified in the literature of computational intelligence, compared to its mechanism revealed in the literature of neuroscience. For example, most of the existing research revolves around scenarios where the reward signals are values directly provided by the environment outside the agent as an evaluation of the superiority of an action step (cf. Fig. 3.3), e.g., scores in a video game, 0 or 1 indicating the success of performing a grasping task in a robotic manipulation task, etc. These algorithms take such reward values for policy learning, without the involvement and updating of other reward components as studied in the literature of neuroscience.

However, in the application of embodied agents with active vision, the system is more comprehensive, and more components are involved in the reinforcement learning framework. Correspondences between these components and the three aforementioned reward components in the reward theory of neuroscience can be well established (see Fig. 3.1 and Fig. 3.2). Specifically, visual perception of the agent works as the sensory component; active vision control works as the attention component, controlling the viewpoint of the visual perception component; information sufficiency evaluation that assesses the the sufficiency of perceived information for achieving the given task (i.e., the quality of the active vision control policy and the visual perception component), and eventually generates rewards for RL algorithms works as the value component. The information sufficiency evaluation component varies depending on the type of agents. For action-only agents, the information sufficiency evaluation component is normally defined using a fixed surrogate metric, such as information gain. In contrast, the information sufficiency evaluation component is more complicated for action-response agents, as the component should be formulated and optimized as a reward model. Rewards are determined by all three components, instead of simply coming from the external environment as an evaluation of the superiority of the action policy according to predefined rules.

The alignment between rewarding components of active-vision agents and those of humans and animals motivates us to model such agents using *unified modular neural networks*, and train the active vision control policy using *reinforcement learning*, to mimic the natural learning mechanism. Reinforcement learning is a promising technique to endow action-response embodied agents (cf. Fig. 1.2) with the capability of active vision, with reasons as below:

1. RL has demonstrated versatility in learning behaviors across diverse scenarios, ranging from game environments to realistic robotic control tasks.

2. The effectiveness of deep neural networks in function approximation endows RL with great potential in incorporating other components, for example, multimodal perception components for state approximation. [Hermann et al., 2017, Hill et al., 2021].

3. The characteristics of *trial-and-error search* and *delayed reward* endow RL algorithms with the ability to optimize action policies based on high-level long-term objectives. This aligns well with the task setup of active vision control in action-response embodied agents.

Nevertheless, it is known that RL algorithms are unstable to train. This issue is even more fierce in the training process of the active vision control policy under the high-level framework of the action-response agent with active vision (cf. Fig. 1.3) because it is difficult to define a reliable reward function for the learning of the active vision control policy using feedback from the response module. The final task-relevant response highly depends on the performance of the active vision control policy, and in turn, the reward for training the active vision control policy depends on the performance of the response module. How to stabilize the simultaneous training process of such models is an open question.

In this chapter, we begin by introducing the basics of reinforcement learning for behavior policy learning, including its mathematical formulation, the characteristics of RL, the relationship between reinforcement learning, supervised learning,

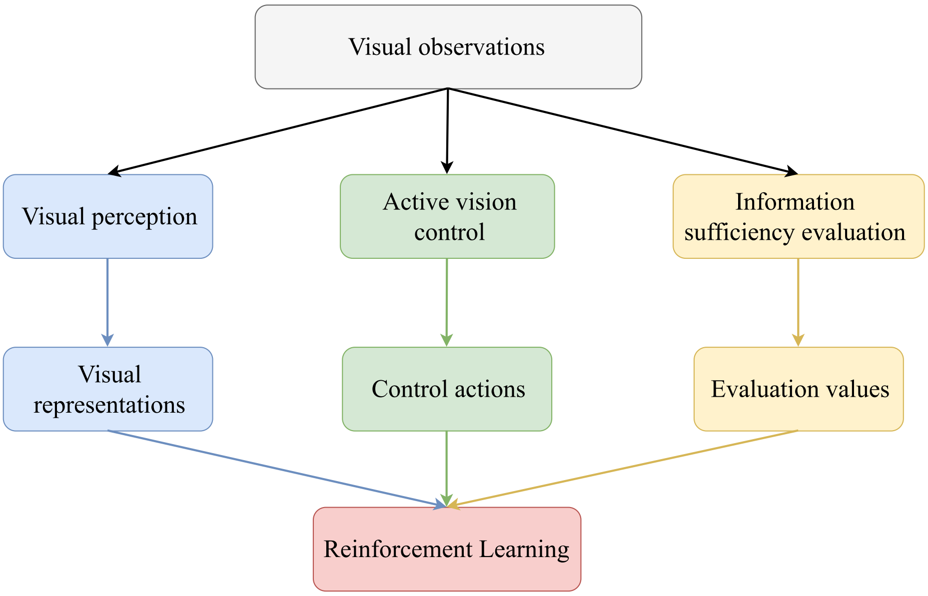


Figure 3.2: Components of the rewarding process in reinforcement learning for the active vision of embodied agents.

and unsupervised learning, and the introduction to some common RL algorithms. Following this, we discuss the background of reward functions in RL, covering aspects such as their significance in stable and successful RL, challenges, current methodology, and potential promising solutions. Following Sutton and Barto [1998], in this thesis, we use the term *reinforcement learning* to refer to the problem, the class of approaches, and the research field simultaneously.

3.2 Reinforcement Learning Basics

The learning paradigm of "learning from trial and error" seems intuitive and simple. However, how to formulate it, develop corresponding computational approaches, and use these approaches to train artificial models for tackling practical problems has been a long-standing research question. A large amount of research has been conducted to answer this question.

3.2.1 Formulation of Reinforcement Learning

In an RL scenario, an **agent** interacts with an **environment** (cf. Fig. 3.3). At each time step ($t = 0, 1, 2, ..., T$), the agent is at a state $s_t \in \mathcal{S}$, where $\mathcal{S}$ is the state space. The **policy** of the agent $\pi(a_t \mid s_t) = \mathbb{P}_\pi [A = a_t \mid S = s_t]$ selects and performs an action $a_t \in \mathcal{A}$, where $\mathcal{A}$ is the action space. Given the present state s_t and the selected action a_t, a transition probability function $P(s_{t+1} \mid s_t, a_t)$ represents the probability of a state the agent will arrive in at the next time step. A reward function $R(s_t, a_t, s_{t+1})$ specifies the reward r_{t+1} the agent receives after the execution of the action and arrival at the new state. Both the transition probability and the reward function are defined by the world model (also called the dynamic model), which is always unknown in real-world task setups. The interaction between the agent and its environment leads to a trajectory $\{S_0, A_0, R_1, S_1, A_1, R_2, ..., S_T\}$, where state S_T is the terminal state. RL aims to learn a sequential decision-making policy $\pi(a \mid s) = \mathbb{P}_\pi [A = a \mid S = s]$ to maximize the expectation of cumulative rewards over time, i.e., $\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$. The interaction between the agent and its environment can be formulated using Markov Decision Processes (MDPs) [Bellman, 1957].

MDPs are stochastic processes satisfying the Markov property, i.e., future states depend only on the present state and action. The assumption of the Markov property provides reasonable simplification for the modeling of sequential decision-making problems, endowing the framework of MDPs with a tradeoff between applicability and mathematical tractability. For example, the Markov property makes it plausible to use the data of one-step transitions for policy learning, e.g., the policy updating method in temporal-difference learning, which will be introduced in Sec. 3.2.4. An MDP is formally formulated as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p_{\mathrm{E}}, \rho, r, \gamma \rangle$, where, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p_{\mathrm{E}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the state transition probability, $\rho : \mathcal{S} \to \mathbb{R}$ is the distribution of the initial state, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward on each transition, and $\gamma \in (0, 1)$ is a discount factor.

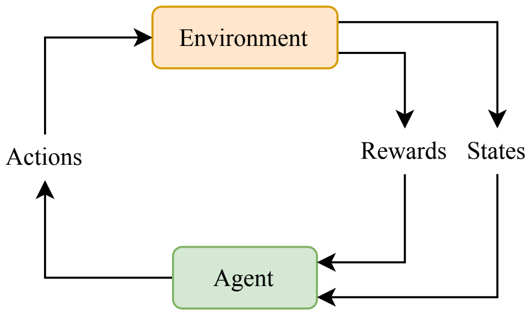


Figure 3.3: The agent-environment interaction loop of reinforcement learning. This figure is based on Sutton and Barto [1998].

3.2.2 Characteristics of Reinforcement Learning

Two key characteristics distinguish RL from other machine learning approaches. The first one is *trial-and-error search* [Sutton and Barto, 1998], i.e., an agent actively tries different trajectories in the environment and updates its policy according to the environmental feedback. This makes it possible for the agent to learn its action strategies through interaction with the environment rather than human supervision. The distribution of training data depends on the previous and current policies of the agent, as training data is collected dynamically during the training process. In contrast, in supervised learning and unsupervised learning, the training data is pre-collected, thus the data distribution is deterministic and static. This raises the challenge of *the trade-off between exploration and exploitation*. On the one hand, the agent needs to explore the environment with novel action strategies to discover trajectories with higher cumulative rewards compared with those that have been visited in history to make it possible for the agent to learn a better policy. On the other hand, the policy of the agent is updated to encourage the action strategies that have been recognized as good ones. Thus, during the learning process, the agent tends to exploit visited trajectories with high cumulative rewards, which would lead to a different distribution of future collected data compared with the strategy of only encouraging exploration. In addition, from the perspective of learning in a stochastic environment, an agent needs to exploit a trajectory many times to estimate the expected reward corresponding to the trajectory. Thus, it is critical for the RL agent to balance exploration and exploitation to achieve the acquisition of optimal decision-making strategies. Fig. 3.4 illustrates the effects of exploitation and exploration in policy learning.

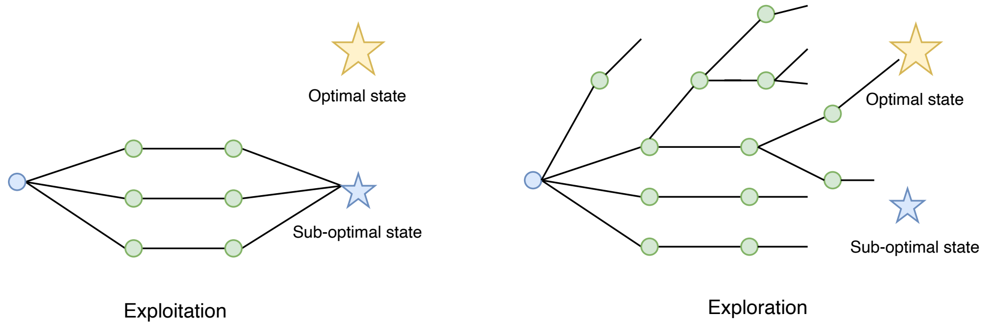


Figure 3.4: Illustration of exploitation and exploration in the discovery of the optimal trajectory. Through exploitation, the agent possibly misses the optimal state and gets stuck in a sub-optimal solution, while exploration contributes to the success of finding the optimal solution. Blue circles in the graphs indicate starting states, and green circles indicate intermediate states.

There are already some approaches to control the trade-off between exploration and exploitation in the literature. A simple and commonly used one is the ε-greedy method, where we can adjust the value of ε in the range of 0 to 1. The agent chooses the action with the highest estimated probability with probability $1 - \varepsilon$ and other actions with probability ε. However, the value of ε needs careful tuning in real applications, depending on the task and environment. Other approaches include Softmax exploration where the chosen action is sampled from the estimated distribution, adaptive epsilon where the value of ε decreases to reduce exploration over the training process, count-based exploration where the agent is encouraged to select actions that appear rarely in the past, and parameter noise where exploration is encouraged by adding noise to parameters of policy or value functions or by injecting noise into the action selection process [Eberhard et al., 2022]. The determination of the trade-off method is still an empirical process in an individual RL problem.

The second characteristic of RL is *delayed reward* [Sutton and Barto, 1998]. Optimizing the policy on delayed rewards makes it possible to consider the task as a whole and optimize the policy to maximize the cumulative reward over time. This characteristic makes it possible to achieve a globally optimal solution in principle. However, technically, it is challenging to perform RL in long-horizon tasks with sparse reward signals, where the agent can only get rewards at the end of each episode, i.e., the reward is delayed. For example, in a goal-seeking navigation task in a large environment, where the agent should take many steps sequentially, including searching for keys, moving to doors, opening doors, and reaching the goal, the agent can only get a reward when the agent successfully achieves the goal or when the budget (e.g., maximum running time and number of movement steps) runs out. The task-specific navigation policy is challenging to learn through RL because it rarely happens that the agent achieves the goal by accident. How to assign credit (or blame) to action steps when the task is successfully achieved (or failed) is crucial for RL agents to handle delayed rewards to learn policies efficiently. Some methods were developed for better credit assignment, e.g., eligibility traces [Barto et al., 1983] and experience replay [Andrychowicz et al., 2018]. However, accurate credit assignment is still a challenge. Fig. 3.5 presents an example of policy learning from delayed rewards in a grid world environment.

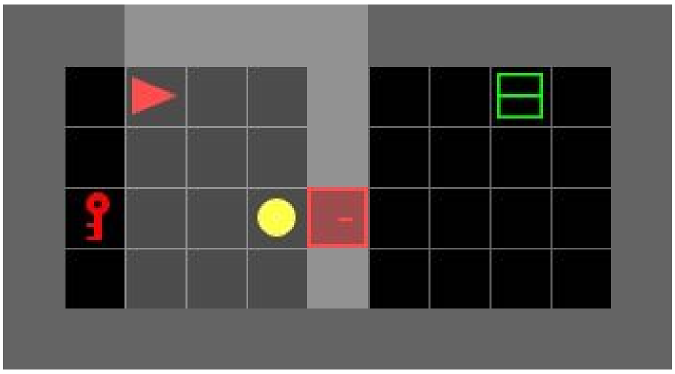


Figure 3.5: Delayed reward in an example case of the BabyAI environment [Chevalier-Boisvert et al., 2019]. In this example, the agent (red triangle), is tasked to pick up the green box, which is located in a neighbor room of the agent. Only by successfully picking up the target object can the agent receive a positive reward. Otherwise, the agent gets a reward of 0. To collect high rewards, the agent needs to remove the obstacle (yellow circle) blocking the door, pick up the red key and use it to open the red door and pick up the target object (green box) in the end. The reward is delayed and sparse in this task. On the one hand, the delayed reward provides an overall optimization objective for policy learning. On the other hand, the sparse reward makes the policy learning hard.

3.2.3 Reinforcement Learning, Supervised Learning, and Unsupervised Learning

Reinforcement learning, supervised learning, and unsupervised learning are three primary approaches in the realm of machine learning. Each of them has its unique features, strengths, and application areas.

Supervised learning has been the most dominant class of problems in machine learning research in the last decades. Human-annotated data is needed in supervised learning, where models learn mapping functions from samples and corresponding human-annotated labels and are expected to generalize to unseen samples. Reinforcement learning, different from supervised learning, does not require dense human supervision signals provided in annotations. Though it is also possible to learn action behaviors using supervised learning in principle, it is always impractical to collect and annotate agent states with desired actions in a sequential decision process, e.g., it is intractable for human annotators to annotate all the possible optimal next movement in a Go game [Silver et al., 2016, 2017]. On the contrary, the mechanisms of trial-and-error searching and delayed rewarding in reinforcement learning make it possible to learn optimal action behaviors without the need for human-annotated actions. Supervised learning and reinforcement learning are not mutually exclusive. Some components of reinforcement learning approaches are trained using supervised learning, e.g., the critic model in the actor-critic methods [Konda and Tsitsiklis, 1999, Mnih et al., 2016], and world models in model-based methods [Schrittwieser et al., 2020, Hafner et al., 2023].

Unsupervised learning is another important category in machine learning research. It is similar to reinforcement learning from the perspective that both of them do not need human-annotated data. However, these two approaches are different in their purposes: unsupervised learning is aimed at uncovering hidden representations of a bunch of data, while reinforcement learning is to maximize rewards to learn interaction behaviors.

Recent research suggested that the pathway to highly generalizable artificial intelligence systems may involve a combination of all of these three main machine learning methods [Ouyang et al., 2022, Touvron et al., 2023]. Specifically, unsupervised learning is used to learn general representations from unlabeled data, supervised learning is to incorporate rich task-relevant human knowledge into the system, and reinforcement learning is used to finetune the system to pose fine-grained control to the system by providing an extra small amount of guidance signals. Though supervised learning and unsupervised learning are both crucial and interesting to study, in this thesis, we focus on reinforcement learning approaches for the learning of action behaviors of embodied agents.

3.2.4 Reinforcement Learning Algorithms

Based on how to derive the action policy, RL methods can be divided into two major categories: value-based and policy gradient algorithms. Both of these two methods are built based on value function estimation (see Section 3.2.5). Value-based algorithms solve RL problems by estimating the state-value function or the action-value function. The action is selected using value estimates, for example, selecting the action with the highest action value in a given state or sampling actions according to the distribution of the action value estimates obtained from a Softmax function. Typical value-based algorithms include SARSA [Rummery and

Niranjan, 1994], Q-learning [Watkins, 1989], and Deep Q-Networks [Mnih et al., 2013]. Value-based algorithms are more suitable in RL problems with a discrete action space. On the contrary, policy-gradient algorithms directly optimize a policy function and thus are more suitable for solving RL problems with a continuous action space. Typical policy gradient methods include REINFORCE [Williams, 1992], TRPO [Schulman et al., 2015a], PPO [Schulman et al., 2017]. Some of these algorithms will be introduced later in Sec. 3.2.6 and Sec. 3.2.7 in more detail.

Actor-critic methods combine ideas of both value-based and policy gradient methods, where an actor, working as a policy function, suggests actions and the critic, working as an action-value function, evaluates the quality of the action selected by the actor. This class of methods can handle tasks with continuous action spaces and the training process is more stable compared to policy gradient algorithms. In the optimization of actor-critic methods, the actor is optimized to maximize estimated values by the critic to maximize the expectation of cumulative rewards, while the critic is usually updated using temporal-difference learning to optimize the accuracy of value estimates. The actor and critic functionally intertwine with each other thus careful hyperparameter tuning is required. Some variants of the original actor-critic method, such as Advantage Actor-Critic (A2C) and Asynchronous Advantage Actor-Critic (A3C) [Mnih et al., 2016] have been designed for higher training efficiency.

Next, we will introduce some representative RL algorithms, most of which have demonstrated effectiveness in applications. We note that some important and prospective methods are not introduced in this part, such as model-based RL algorithms [Schrittwieser et al., 2020, Hafner et al., 2023], offline RL algorithms [Levine et al., 2020], and transformer-based RL algorithms [Chen et al., 2021]. We recommend readers read corresponding references if they are interested in up-to-date work in these areas.

3.2.5 Value Functions

The estimation of **value functions** is essential for developing the optimal policy for most RL methods. There are two types of value functions: the state-value function $V(s)$ estimates the goodness of the agent being in a given state, and the action-value function $V(s,a)$ estimates the goodness of the agent performing a given action in a given state.

Value functions are defined using the term **return**, i.e., the cumulative discounted summation of future rewards $G_t = \sum_{i=t+1}^{T} \gamma^{i-t-1} R_i$, where $t \in [0,T)$ and $G_T = 0$. $\gamma \in (0,1)$ is the discount factor that is to penalize the contribution of rewards in farther future. The state-value function is defined as $V_\pi(s) = \mathbb{E}_\pi [G_t \mid S_t = s]$, and the action-value function is defined as $Q_\pi(s,a) = \mathbb{E}_\pi [G_t \mid S_t = s, A_t = a_t]$. The relation between V_π and Q_π can be formulated using the distribution of estimated actions $\pi\left(a \mid s\right)$, as

$$
\begin{aligned}
V_\pi\left(S_t=s\right) &= \mathbb{E}_\pi\left[Q_\pi\left(S_t=s, A_t=a\right)\right] \\
&= \sum_{a \in \mathcal{A}} \pi\left(A=a \mid S=s\right) Q_\pi\left(S_t=s, A_t=a\right), \\
&\text{or} \\
&= \int_{\mathcal{A}} \pi\left(A=a \mid S=s\right) Q_\pi\left(S_t=s, A_t=a\right) da.
\end{aligned}
$$

The subscript π of V_π and Q_π indicates that value functions depend on the policy. Given a policy, value functions can be estimated from its experience trajectories. We expect to obtain an optimal policy that can lead to optimal value functions, i.e., optimal expectation of return.

The **advantage** is defined as the difference between the action-value function and the state-value function, i.e., $A(s,a) = Q_\pi(s,a) - V_\pi(s)$. It represents the value or gain of taking action a at state s compared with taking a random action sampled from the distribution of policy $\pi(a \mid s)$. It is a better measurement of the quality of an action at a state compared with the action value because it considers the goodness of the given state and the relative improvement of taking a specific action. The advantage can be represented as a normalization of the action value over the action distribution determined by the policy:

$$
\begin{aligned}
A(s,a) &= Q_\pi(s,a) - V_\pi(s) \\
&= Q_\pi(s,a) - \sum_{a \in \mathcal{A}} \pi\left(A=a \mid S=s\right) Q_\pi\left(S_t=s, A_t=a\right).
\end{aligned}
$$

It has a lower variance compared with the action value, thus leading to a more stable and efficient learning process. The idea of using advantage for policy learning is a general technique across a wide range of RL algorithms. For example, it is incorporated into the actor-critic algorithm, where the critic estimates the state value to calculate the advantage for more stable policy updating.

Value functions are recursive. Given a policy π, a functional relation between the value of a state $S_t = s$ and its successor state $S_{t+1} = s'$ can be derived as

$$\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi\left[G_t \mid S_t = s\right] \\
&= \mathbb{E}_\pi\left[\sum_{i=t+1}^{T} \gamma^{i-t-1} R_i \mid S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \sum_{i=t+2}^{T} \gamma^{i-t-1} R_i \mid S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}} \sum_{r} p(s', r \mid s, a)\left[r + \gamma \mathbb{E}\left[G_{t+1} \mid S_{t+1} = s'\right]\right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{S}, r} p(s', r \mid s, a)\left[r + \gamma V_\pi(s')\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s\right].
\end{aligned}$$

This is the **Bellman equation** for the state value. It suggests that the value of the present state equals the summation of the discounted expectation of the value of the next state and the corresponding expectation of the reward obtained by reaching the next state. A similar form of function exists also for the action-value function as below:

$$\begin{aligned}
Q_\pi(s, a) &= \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a_t\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s, A_t = a_t\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma \mathbb{E}_\pi Q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a_t\right].
\end{aligned}$$

As we can see from the above equations, Bellman functions build the relation between state values (action state values) and their successors.

3.2.6 Value-based Algorithms

Temporal-Difference Learning Temporal-Difference (TD) learning can evaluate and update policies when the world model is unknown and is known as a core method in RL. The target of value updating for a given state is $V^*(S_t) = R_{t+1} + \gamma V(S_{t+1})$, which is the Bellman equation without expectation, which results in bias in the optimization goal of value functions. The present state value can be updated in a heuristic manner as below:

$$\begin{aligned}
V(S_t) &\leftarrow V(S_t) + \alpha(V^*(S_t) - V(S_t)) \\
&\leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)).
\end{aligned}$$

As we can see, TD learning updates value estimation using existing value estimates instead of directly using returns calculated from experience trajectories, which is called "bootstrapping".

SARSA and Q-Learning SARSA and Q-learning are two representative methods of TD learning [Rummery and Niranjan, 1994, Watkins, 1989]. SARSA is an on-policy approach that uses online transition data of a quintuple (S_t, A_t, R_t, S_{t+1}, A_{t+1}), from which the name "SARSA" comes, to update the action-value function as the following formula:

$$Q_\pi(S_t, A_t) \leftarrow Q_\pi(S_t, A_t) + \alpha(R_{t+1} + \gamma Q_\pi(S_{t+1}, A_{t+1}) - Q_\pi(S_t, A_t)).$$

In contrast, Q-learning works in an off-policy manner where the (target) policy is updated with the objective of selecting the action corresponding to the highest action value, regardless of the (behavior) policy that generates the training trajectories. The action-value function updates in the following formula:

$$Q_\pi(S_t, A_t) \leftarrow Q_\pi(S_t, A_t) + \alpha(R_{t+1} + \gamma \mathrm{max}_{a_{t+1} \in \mathcal{A}} Q_\pi(S_{t+1}, a_{t+1}) - Q_\pi(S_t, A_t)).$$

Compared to SARSA, Q-learning uses the transition data of a quadruple (S_t, A_t, R_t, S_{t+1}) to update its action value.

Deep Q-Networks Value functions can be represented using tables by saving state or action values. However, when the state and action space are large or continuous, it is infeasible and impractical to build such tables. Deep Q-Networks (DQNs) use neural network-based approximation functions as a variant of Q learning [Mnih et al., 2013]. However, the issue of instability and divergence arises when the techniques of function approximation, bootstrapping, and off-policy training are applied together, which is called the issue of Deadly Triad [Sutton and Barto, 1998]. To stabilize the training process, DQNs use the technique of experience replay, and a separate and scheduled updated target network. This method has achieved great success in the task of Atari games and has been recognized as a milestone in the development of deep RL. The action-value function is a neural network with parameters θ, which is trained with the training objective of minimizing the loss

$$\mathcal{L}_\theta = \mathbb{E}_{(s,a,r,s') \sim D}\left[\left(r + \lambda \mathrm{max}_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right],$$

where D is the data of transitions, θ^- is the parameters of the separate target network, and θ is the parameters of the network of the actual action-value function.

3.2.7 Policy-gradient Algorithms

Vanilla Policy Gradient The above-introduced value-based algorithms all learn value functions and then select actions accordingly, for example, through greedy or ε-greedy action selection. Policy gradient methods directly learn the parameterized policy function $\pi(a \mid s, \theta) = \mathbb{P}(A_t = a \mid S_t = s, \theta)$, where θ is the parameters of the policy function. These methods have several advantages compared to value-based methods. First, the continuous parameters of a policy gradient method can update and adjust the policy smoothly, which can make the training process more stable and efficient. Second, policy gradient methods could result in a better policy for a task whose policy function is easier to learn.

For discrete state and action space, the learned policy function estimates the action preference value $h(s, a, \theta)$ for each state-action pair, which is then processed by a softmax function for action selection to get the action distribution. A prior distribution, e.g., a Gaussian or Laplace distribution, is used to model the policy for continuous state and action space. Value functions are not required for action selection even though they could be useful, e.g., in actor-critic methods. For episodic tasks, the learning objective of policy gradient methods can be defined as the value of the starting state of an episode, i.e., $\mathcal{J}(\theta) := V^{\pi}(s)$, where s is the initial state of an arbitrary trajectory. The parameter of the policy is updated in the following scheme

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathcal{J}(\theta),$$

where α is the updating step size. The optimization objective can be extended as

$$\begin{aligned}\mathcal{J}(\theta) &= \sum_{s \in \mathcal{S}} d^{\pi}(s) V^{\pi}(s) \\ &= \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\theta}(a \mid s) Q^{\pi}(s, a),\end{aligned}$$

where $d^{\pi}(s)$ is the deterministic distribution of a Markov chain and $Q^{\pi}(s, a)$ is the action-state function, both of which are subject to the policy. The effect of the policy parameter on the state distribution is nontrivial to formulate. Fortunately, the **policy gradient theorem** provides an analytical solution to calculate the gradient of the optimization objective with respect to the policy parameter, which is what we need for optimizing the policy function through gradient descent. The gradient of $\mathcal{J}(\theta)$ with respect to θ is derived as

$$\begin{aligned}\nabla_{\theta} \mathcal{J}(\theta) &\propto \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} Q^{\pi}(s, a) \nabla_{\theta} \pi_{\theta}(a \mid s) \\ &= \mathbb{E}_{s \sim d^{\pi}(s), a \sim \pi_{\theta}} [Q^{\pi}(s, a) \nabla_{\theta} \ln \pi_{\theta}(a \mid s)].\end{aligned}$$

This is the derivation of the vanilla policy gradient algorithm. This gradient estimate has no bias but high variance, leading to an unstable learning process. Many follow-up works were done aiming at gradient estimates with a trade-off between the variance and bias, to achieve more stable and efficient training [Schulman et al., 2015b].

Monte-Carlo Policy Gradient Monte-Carlo policy gradient, i.e., REINFORCE, is a simple and representative policy-gradient algorithm [Williams, 1992]. As the return is an unbiased estimation of the action value, i.e., $Q^{\pi}(S, A) = \mathbb{E}_{\pi}[G \mid S, A]$, the gradient of $\mathcal{J}(\theta)$ with respect to θ can be formulated as

$$\begin{aligned}\nabla_{\theta} \mathcal{J}(\theta) &= \mathbb{E}_{s \sim d^{\pi}(s), a \sim \pi_{\theta}} [Q^{\pi}(s, a) \nabla_{\theta} \ln \pi_{\theta}(a \mid s)] \\ &= \mathbb{E}_{s \sim d^{\pi}(s), a \sim \pi_{\theta}} [G \nabla_{\theta} \ln \pi_{\theta}(a \mid s)],\end{aligned}$$

The gradient estimates of REINFORCE are unbiased but have a large variance, which makes the training unstable. To mitigate this issue, a variant of REINFORCE, named actor-critic REINFORCE, was designed. In actor-critic REINFORCE, a value function is estimated as in value-based methods. Then an advantage value $A(S, A) = Q(S, A) - V(S) = \mathbb{E}_\pi[G \mid S, A] - V(S)$ is calculated for the gradient estimation. The use of the advantage in place of the action value or the return reduces the variance of gradient estimates while keeping the gradient estimation unbiased. Compared to the basic REINFORCE method, the actor-critic variant can learn the policy faster and more stable.

Proximal Policy Optimization Proximal Policy Optimization (PPO) is the current most popular RL algorithm even though it has been proposed for many years [Schulman et al., 2017]. Different from the REINFORCE and actor-critic methods, which are both on-policy algorithms, PPO is an off-policy algorithm: the policy used for generating data for training (*the behavior policy*) is different from the policy that the RL method is going to improve (*the target policy*). The main advantage of off-policy algorithms is that off-policy algorithms have higher sample efficiency because they can use the samples from collected transitions multiple times to update the policy. Besides, off-policy methods potentially have better exploration capability because the data is collected by an old policy that is different from the current policy. The training objective of PPO is defined as

$$\begin{aligned}\mathcal{J}(\theta) &= \mathbb{E}_{s\sim\rho^{\pi_{\theta'}}(s)}\left[V(s)\right] \\ &= \mathbb{E}_{s\sim\rho^{\pi_{\theta'}}(s)}\left[\sum_{a\in\mathcal{A}}\pi_\theta(a \mid s)\hat{A}_{\theta'}(s,a)\right],\end{aligned}$$

where $\pi_{\theta'}$ is the behavior policy, $\rho^{\pi_{\theta'}}(s)$ is the state distribution corresponding to the policy $\pi_{\theta'}$, $\hat{A}(\cdot)$s is the estimated advantage, which is aimed at reducing the variance of gradient estimates, π_θ is the target policy. By using the *importance sampling* trick, which is a technique of estimating probabilities of a distribution using samples from another different distribution, the resulting objective function can be further extended as

$$\begin{aligned}\mathcal{J}(\theta) &= \sum_{s\in\mathcal{S}}\rho^{\pi_{\theta'}}(s)\sum_{a\in\mathcal{A}}(\pi_\theta(a \mid s)\hat{A}_{\theta'}(s,a)) \\ &= \sum_{s\in\mathcal{S}}\rho^{\pi_{\theta'}}(s)\sum_{a\in\mathcal{A}}(\pi_{\theta'}(a \mid s)\frac{\pi_\theta(a \mid s)}{\pi_{\theta'}(a \mid s)}\hat{A}_{\theta'}(s,a)) \\ &= \mathbb{E}_{s\sim\rho^{\pi_{\theta'}}(s),a\sim\pi_{\theta'}}\left[\frac{\pi_\theta(a \mid s)}{\pi_{\theta'}(a \mid s)}\hat{A}_{\theta'}(s,a)\right],\end{aligned}$$

where the ratio $\frac{\pi_\theta(a|s)}{\pi_{\theta'}(a|s)}$ is called *importance weight*, representing the relative difference between the target policy and the behavior policy. The ratio between the target policy and the old policy can be defined as

$$r(\theta) = \frac{\pi_\theta(a \mid s)}{\pi_{\theta'}(a \mid s)},$$

which is a function of θ.

During training, the difference between the behavior policy and the target policy needs to be constrained because of the issue of the surrogate optimization objective: the surrogate optimization objective should be as close as possible to the real optimization objective [Schulman et al., 2015a]. PPO provides a simple yet effective method to impose such a constraint on the training procedure. Specifically, the ratio $r(\theta)$ is forced to be in a small range $[1-\varepsilon, 1+\varepsilon]$, where ε is a small number as a hyperparameter, to constrain the difference between the two policies. The training objective then becomes

$$\mathcal{J}(\theta) = \mathbb{E}_{s\sim\rho^{\pi_{\theta'}}(s), a\sim\pi_{\theta'}}\left[\min(r(\theta)\hat{A}_{\theta'}(s,a), \mathrm{clip}(r_\theta, 1-\varepsilon, 1+\varepsilon)\hat{A}_{\theta'}(s,a))\right],$$

where the function $\mathrm{clip}(r_\theta, 1-\varepsilon, 1+\varepsilon)$ clips the value of r_θ between the range of $[1-\varepsilon, 1+\varepsilon]$. The function $\min(\cdot)$ returns the minimum value between the original one and the clipped one as the training objective.

In the practical PPO implementation, the policy network shares parameters with the value network to predict state values. Two additional terms are added to the training objective below

$$\begin{aligned}\mathcal{J}(\theta) = \mathbb{E}_{s\sim\rho^{\pi_{\theta'}}(s), a\sim\pi_{\theta'}}[&\min(r(\theta)\hat{A}_{\theta'}(s,a), \mathrm{clip}(r_\theta, 1-\varepsilon, 1+\varepsilon)\hat{A}_{\theta'}(s,a)) \\ &- c_1(V_\theta(s) - V_{\mathrm{target}})^2 + c_2 S[\pi_\theta](s)],\end{aligned}$$

where the square error loss $(V_\theta(s) - V_{\mathrm{target}})^2$ indicates the value estimation error, and $S[\pi_\theta](s)$ denotes an entropy bonus to encourage exploration, c_1 and c_2 are two hyperparameters to control the weights of three factors in the training objective.

3.3 Reward Functions in Reinforcement Learning

The above-introduced RL algorithms all depend on well-designed and accurate reward signals for policy updating. *Reward function* is one of the key elements in RL to evaluate and produce reward signals given states and actions. The design of reward functions is critical for the success of learning the expected behavior policies [Ma et al., 2023]. Nevertheless, in practice, most RL practitioners design their reward functions in an ad hoc process of trial and error. They always search for a relatively good reward function manually and can not guarantee that the selected one is the optimal one [Booth et al., 2023]. It has been reported that inappropriate reward functions could lead to the issue of the learning of sub-optimal policies, reward hacking [Sutton and Barto, 1998, Everitt et al., 2017], unstable training, and the failure of behavior learning. In this section, we introduce common techniques in the designing of good reward functions for RL methods to use.

The difficulty of designing a proper reward function varies depending on the task. It is easier to define appropriate reward functions in tasks like board games and video games, where the rule of achieving the task and the evaluation of success are clear, compared to tasks in complicated real-life scenarios, such as robotic tasks, where the environment is complex and the evaluation of success is challenging. Thus, apart from the developing of RL algorithms that have good inherent probabilities in stable and efficient learning, the designing of proper reward functions is essential as well to successfully applying RL approaches in more broad and challenging realistic scenarios.

3.3.1 Human-designed Reward Functions

Reward functions are normally defined by human experts using their domain knowledge about the target task [Booth et al., 2023]. Though it is always an empirical and iterative process to determine a good reward function with trial and error, some general rules have been discovered.

Reward Shaping using Expert Knowledge Reward shaping is an effective technique to accelerate RL or a key to successful RL [Dorigo and Colombetti, 1994, Randløv and Alstrøm, 1998]. When using reward shaping, the ultimate goal of the target task is not changed, while the reward function is adjusted to make the RL agent learn the expected behavior more efficiently. A common use case of reward shaping is converting sparse reward into dense reward, thus the reward function can provide more guidance signals for the agent. This technique makes the convergence faster by explicitly guiding the agent to explore desired behaviors. For example, in a robotic manipulation task, a shaped reward typically contains the distance between the robot grasper and the target object as an additional reward term along with the ultimate task reward, i.e., the success of grasping the target object. This distance-based reward element encourages the robot grasper to reach the object faster in exploration and thus accelerates the learning process [Nagpal et al., 2020].

Another common idea of reward shaping is separating a task manually into a sequence of sub-tasks, each of which has a sub-goal [Chane-Sane et al., 2021]. For example, in a robotic pick-and-place task where the robot is asked to pick up a target object and place it into a specific location, the task can be separated into a sequence of sub-tasks, including reaching the goal object, grasping the object, picking up the object, and placing the object into the target place. Only rewarding the success of the whole task makes the reward signal very sparse and rarely happens, which makes the RL very slow. By separating the task into a sequence of sub-tasks and rewarding each of them, the robot can easily achieve the long-horizon task and thus learn the policy more efficiently.

However, reward shaping can potentially incur unexpected biases in behavior learning, which leads to suboptimal or wrong policy learning [Sutton and Barto, 1998, Norvig, 2019]. So shaped reward functions should be designed carefully. For example, the issue of reward hacking [Everitt et al., 2017] can happen when the shaped reward function incurs unintended incentives to the agent. The agent may exploit the shaped objective to find unintended shortcuts, which leads to bad performance on the original task while gathering high reward values. For example, if we use the mass of garbage collected as an intermediate reward for a cleaning robot, the agent may learn to put useful stuff into the garbage bin to gain a higher reward, which is unexpected and harmful.

Reward Scaling Deep learning methods have been proven to be sensitive to the scale of input data. A large amount of research has been done such as batch normalization [Ioffe and Szegedy, 2015], layer normalization [Ba et al., 2016], and etc. Large input values can lead to instability and slow convergence in deep learning, for example, leading to the issue of gradient exploding. Reward signals, as a kind of input data of neural networks, also follow this principle. Reward scaling is a technique to enhance the learning process of RL to make the training more stable and efficient [Engstrom et al., 2019]. By scaling rewards, the magnitude of the value of the reward is changed for the sake of numerical computation, while the pattern of the reward remains unchanged. Reward scaling can be seen as a part of the design of the reward function, or a kind of postprocessing of reward signals as well.

As introduced before, value functions are trained with the objective of regressing on the expectation of the return (see Sec.3.2.4). The scale of the reward value should align with the value function for effective learning. Reward scaling can be achieved simply by multiplying all rewards by a fixed factor to change the range of reward values. This is like the linear transformation of the raw reward values. The more generalized form of this transformation is post-processing raw rewards with a specific function, such as the exponential function or the logarithmic function, which not only changes the range of reward values but also the shape of the reward curves [Li et al., 2023]. Reward normalization is suitable for RL algorithms that process a batch of transition data for policy updating, such as PPO [Schulman et al., 2017]. Raw rewards are subtracted by the mean and divided by the standard derivation to be mapped into the range of 0 to 1 [Touvron et al., 2023]. Reward clipping constraints reward value in a specific range to avoid extremely large or low reward values [Strouse et al., 2022].

These reward scaling methods are used empirically depending on specific environments and tasks without sufficient theoretical guidance. Additional hyperparameters are introduced in these methods, such as the scaling factor or the clipping range. Optimal hyperparameters should be tuned according to the observation of the training curve. Thus, these methods are always treated as practical tricks for achieving stable and efficient RL training.

Intrinsic Rewards As it could be nontrivial to design appropriate reward functions for specific tasks, learning basic task-agnostic skills beforehand and then finetuning the agent with these skills on specific tasks is a promising solution [Singh et al., 2004]. The assumption is that the skills learned utilizing intrinsic rewards are reusable for various downstream tasks, thus they can make the learning of task-specific skills easier. These basic skills are learned utilizing intrinsic rewards that

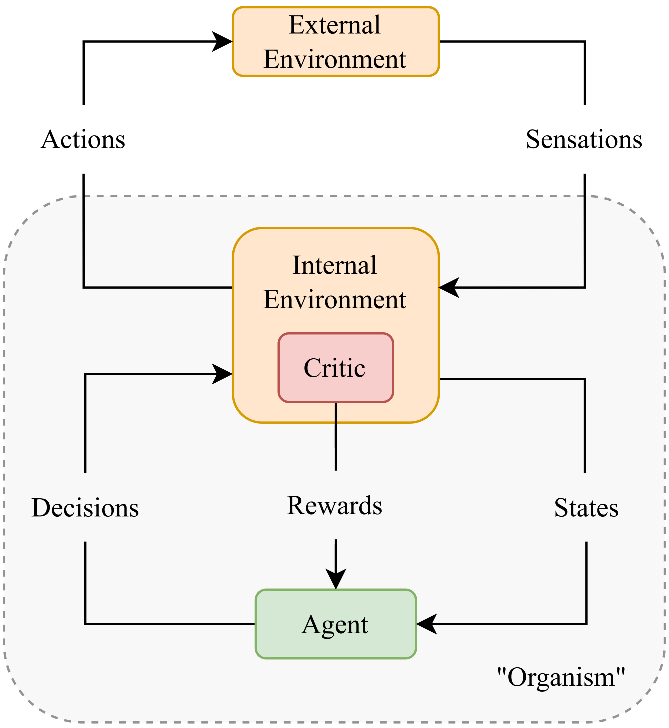


Figure 3.6: An elaborated view of the agent-environment interaction loop of reinforcement learning. This figure is based on Singh et al. [2004].

are task agnostic, for example, they are designed to encourage the agent to explore unseen views of the environment or to discover novel skills to interact with the environment. The theory of intrinsic rewards is derived from an elaborated view of the agent-environment interaction loop (cf. Fig.3.6), where the environment is separated into the external environment, and the internal environment, The external environment is outside of the agent, while the internal environment is in the same organism, e.g., the brain, with the RL agent. The intrinsic motivation is produced by the critic of the internal environment, e.g., identifying the novelty of an arrived state. This is in line with the theory of Schultz [2015], where the reward components can be divided into external and internal components, according to whether the reward is determined by the environment or the brain. How to define intrinsic rewards and how to reuse learned skills obtained from intrinsic rewards are attracting a lot of attention in the RL community in recent years [Singh et al., 2004, Strouse et al., 2022, Li et al., 2023].

3.3.2 Learning-based Approximation of Reward Functions

Reward functions can also be approximated using learning-based methods [Christiano et al., 2017, Ouyang et al., 2022, Li et al., 2023, Ho and Ermon, 2016]. Compared with the classic RL setup where the reward comes from a manually designed reward function by processing feedback from the environment, this approach is useful in situations where the reward function is too complex to design by human experts. The reward function can be approximated by a neural network using data like demonstrations, or other supervision signals like classes in a classification task.

Inverse Reinforcement Learning (IRL) [Ng and Russell, 2000] is a seminal work in recovering reward functions and learning behaviors from observed trajectories. In IRL, a reward function is first recovered and then an RL policy is learned utilizing the recovered reward function as in a normal RL setup. IRL is useful in situations where teacher demonstrations are available while the knowledge of defining an appropriate reward function is absent or not enough. It is challenging in terms of determining the optimal reward function that can lead to the optimal learned policy.

Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon, 2016] is another work in learning a policy from expert demonstrations. Different from IRL, GAIL does not recover a reward function before the learning of the policy. Instead, GAIL formulates the task as a generative adversarial learning problem and simultaneously learns a policy and a discriminator which works as the reward function to provide reward signals. The discriminator is trained to distinguish the ground truth and the generated trajectories, and the policy is trained in RL using the output of the discriminator as reward signals. Compared to the classic generative adversarial networks (GAN) [Goodfellow et al., 2014], the distribution of the collected trajectories is analogous to the ground-truth data distribution, and the policy is analogous to the generator.

Different from GAIL where the discriminator and the policy mutually improve in a continuous adversarial improvement process, in some task setups, the relationship between the policy and the discriminator is collaboration rather than adversarial competition [Mnih et al., 2014, Christiano et al., 2017, Ouyang et al., 2022, Li et al., 2023]. For example, in the task of hard attention for image classification [Mnih et al., 2014], the policy controls the attention position of the perception field, and a discriminator takes collected visual information for image classification. The policy and the discriminator functionally collaborate in a way that the performance of the discriminator gives feedback regarding the performance of the policy, while the performance of the policy influences the performance of the discriminator in turn. In the finetuning of large language models (LLMs) using RL from human feedback (RLHF) [Ouyang et al., 2022], a reward model is firstly trained in a supervised learning manner on preference data to obtain the ability to evaluate human preference of generated sentences to provide reward signals. The RL algorithm PPO is then applied to learn a chat policy utilizing the rewards produced by the reward model. As the chat policy improves, the distribution of the generated data shifts, thus the reward model needs to be retrained to handle the novel data distribution for preference evaluation. The reward model and the policy model functionally depend on each other and are trained iteratively. Similar to IRL, where a cost function should be trained for policy learning, the training process of this RLHF paradigm is unstable and hard to converge.

The same training paradigm, where a discriminator and a policy collaborate and improve simultaneously, exists in a broad range of tasks, e.g., learning game and robotic control from human preference [Christiano et al., 2017], unsupervised skill discovery [Gregor et al., 2017, Strouse et al., 2022], active perception in robotics [Lakomkin et al., 2018, Li et al., 2021].

3.3.3 AI-designed Reward Functions

Designing an appropriate reward function is always very challenging and crucial in the application of RL algorithms in real-world application scenarios. The empirical and iterative selection process of the reward function is time-consuming when the task is complicated. Taking advantage of rich common-sense knowledge encoded in the pretrained large models, especially pretrained large language models (LLMs), designing reward functions with the help of large models has been proven an effective method for designing reward functions. This kind of work is generally termed reinforcement learning from AI feedback (RLAIF) [Bai et al., 2022b, Lee et al., 2023]

Though this approach seems promising, how to ground the model into the target task and environment is challenging. RLHF, as has been introduced before, is known to be relying on dense human labels, which are expensive to collect. As an improvement of RLHF, Bai et al. [2022b] proposed the concept of Constitutional AI (CAI), where an LLM is finetuned using a small set of basic principles defined by humans, i.e., a constitution, instead of using prompt-response pairs that are constructed using human feedback. CAI uses a separated pretrained LLM to provide feedback, according to the predefined principles as responses, to align the LLM with human values and intentions. Specifically, CAI includes two training stages. The first one is the supervised stage, where a helpful-only model generates responses to harmful prompts, critiques the responses according to a randomly selected principle from the constitution, and then revises the responses. This pipeline runs multiple times with different principles to get a final revised response. After that, the helpful-only model is finetuned on the revised responses with supervised finetuning. The second stage is the RLAIF stage, where the model from the first stage is asked to generate pairs of responses for harmful prompts and is then instructed to choose the best one out of a pair of responses according to the predefined principle. This procedure constructs a preference dataset, which is then used to train a preference model and finetune the policy.

Lee et al. [2023] proposed a straightforward RLAIF method where the reward model is trained with data labeled by a basic LLM after supervised fine-tuning. Similar to CAI, this method also defines some principles but there is no critique and revision process involved. They demonstrated that with well-designed prompts, AI feedback can achieve comparable alignment improvement for LLMs compared with human feedback. Pang et al. [2023] proposed an approach named Self-Improvement by Reinforcement Learning Contemplation (SIRLC), which is built on top of the observation that LLMs are good at self-evaluation. In line with CAI, SIRLC works in a self-evaluation manner. In contrast, there are no predefined principles in self-evaluation but only an evaluation score computed by the LLM itself. A target LLM is then optimized with RL to maximize the expectation of the scores to achieve the goal of LLM alignment.

The idea of using pretrained large models to design reward functions has also been studied in the realm of agents and robotics [Klissarov et al., 2023, Yu et al., 2023, Hu and Sadigh, 2023, Du et al., 2023]. Ma et al. [2023] proposed EUREKA (Evolution-driven Universal REward Kit for Agent) to use pretrained LLMs to automatically design reward functions. Specifically, this method uses the code-generation capability of LLMs, taking the environment code and the task description as the input to generate the code of reward functions. Two crucial techniques, reward candidate sampling, and reward reflection, are invented to tackle the problem of unexecutable reward functions and improve reward functions, respectively. Experimental results suggest that reward functions designed by EUREKA outperform those designed by human experts, showcasing the potential of utilizing feedback from pretraining models for more efficient RL training in more complicated tasks.

3.4 Discussion

In this chapter, we introduced reinforcement learning from both neuroscientific and computational perspectives, with a particular interest in the generation and functions of rewards. It can be seen that there is a gap between the learning theory in neuroscience and existing methods in computational reinforcement learning. Existing research in computational reinforcement learning revolves around a relatively simplified setup, primarily focusing on reinforcement learning algorithms utilizing highly artificial and human-designed reward values. On the contrary, the reinforcement learning theory in neuroscience covers a relatively comprehensive range of procedures, spanning from sensory perception to attention mechanisms, and goal-oriented and subjective value evaluation. These procedures are always not considered and integrated into the computational reinforcement learning framework in existing work on computational reinforcement learning. This thesis seeks to explore possible approaches to incorporate these components into the reinforcement learning process, particularly with a focus on the application of embodied agents with active vision, as the development of such agents necessitates these components in the reinforcement learning process.

Chapter 4

Disembodied Models in Addressing Insufficient Observations

4.1 Introduction

The prevalent application scenarios of insufficient observations motivate our research on embodied agents equipped with active vision. Before our work on embodied agents, it is worth studying to what extent the issue of insufficient observations can be tackled without the application of active vision, i.e., with disembodied models. The answer to this research question motivates our interest in active vision and embodied agents.

In the last decades, deep neural network (DNN) models have achieved significant success in a variety of tasks. A large portion of existing work is on disembodied models that passively process static data fed by users. This is often referred to as "disembodied AI", as the models are not embodied in the physical world, instead, they interact with users through a virtual interface. Thanks to the efficiency of deep learning, certain DNN models have reached human-level performance on many large-scale datasets. Most of these datasets are composed of data that contains *sufficient information* to deduce the corresponding human-annotated label. This simplification makes the datasets appropriate benchmarks for early-stage study on primary principles and methods of deep learning, without too much distraction from annoying, complex, and uncontrollable cases from realistic use cases. For example, the ImageNet dataset [Deng et al., 2009] includes images that provide ample visual details of the object (e.g., pixels of a dog) corresponding to the labeled category (e.g., "a dog") for the image classification task[1], questions are guaranteed answerable given the corresponding images in widely used datasets of the task of visual question answering (VQA) [Antol et al., 2015, Goyal et al., 2017].

[1]It should be noted that in the ImageNet dataset, there are images unintentionally labeled incorrectly, which is called the issue of noisy labeling [Yun et al., 2021].

Although these static datasets were not initially created with the consideration of the issue of insufficient observations, the growing necessity to deploy AI models in diverse real-world situations is driving researchers to look into this issue. In realistic application settings, the challenge of insufficient observations is not only present in embodied agents operating in 3D environments but also occurs in situations dealing with static data. The issue of insufficient observations manifests differently on different tasks. On the VQA task, this issue corresponds to situations where the question is *irrelevant* or *unanswerable* to the corresponding image, i.e., the visual information provided by the image, which serves as the information base for the QA task, is insufficient for answering the given question [Ray et al., 2016, Mahendru et al., 2017, Whitehead et al., 2022, Wu et al., 2023]. For example, in realistic applications, users may mistakenly ask the question "What is the color of the cat" to an image containing only a dog [Mahendru et al., 2017], visually impaired people may ask visually irrelevant questions to an assistant application which is designed to answer visual questions [Gurari et al., 2018]. Similarly, in the image classification task, the issue corresponds to situations where the entities in the given images are unknown to the model. For instance, in daily use, users may feed a blurry image with nothing recognizable to an image classification model to classify or an image with a novel class of objects that are not included in the training data. These situations with insufficient information pose novel requirements for such models to distinguish the sufficiency of present visual information and produce reliable responses.

Though these situations are common in realistic application scenarios and dealing with these situations is essential for improving user experience, it has been demonstrated that existing models across various domains, including visual question answering [Gurari et al., 2018, Whitehead et al., 2022], image captioning [Rohrbach et al., 2018], and pretrained vision-language models (VLMs) [Wu et al., 2023, Zhou et al., 2024] struggle in scenarios with insufficient observations. When the provided visual information is insufficient for responding to the query and the model does not have the capability of actively acquiring further information, these models tend to produce arbitrary responses, which could be potentially harmful to the users. It is unarguable that harmlessness should be the most prioritized property of the responses from an AI model or an embodied assistant agent. An idea to eliminate harmfulness in the responses is converting a model, which produces arbitrary answers in scenarios with insufficient observations, into a conservative one, which abstains from answering.

In the application of models without the capability of active vision, the response of abstaining from classification or answering is better or probably an optimal solution compared with producing an arbitrary classification or answer, as the response is at least not misleading even though the response is helpless [Li et al., 2020, Bai et al., 2022a, Whitehead et al., 2022]. We use the VQA task and specifically study the straightforward idea of training a VQA model with extra "I don't know" labels to convert a potentially harmful model into a harmless one when the question is irrelevant to the visual information provided by the image. As the capability of detecting insufficient observation scenarios is the prerequisite for proceeding with further actions to interact with the environment, this work serves as a preliminary study under the umbrella of the thesis topic on embodied agents with active vision.

4.2 VQA with Irrelevant Questions

Visual question answering [Antol et al., 2015] is an important multimodal task in the field of artificial intelligence in recent years. In this task, a model is expected to produce answers to natural language questions regarding the visual content of corresponding images. This task has received significant interest from researchers because it not only can be utilized to examine the development of multimodal and crossmodal technologies [Fu et al., 2020], but also has great potential in real-world application scenarios [Gurari et al., 2018].

Despite significant progress in recent years, the majority of conducted research focuses on improving accuracy on current hand-curated VQA datasets [Antol et al., 2015, Goyal et al., 2017, Kafle and Kanan, 2017], in most of which questions are relevant to corresponding images by default, i.e., visual information regarding the question is sufficient for generating answers. When given an irrelevant question to an image, current state-of-the-art models would still produce an answer with high certainty, represented by a high probability score, rather than predict that the question is irrelevant and cannot be answered correctly. Obviously, it is not what we expect for an intelligent VQA system. On the one hand, this situation suggests that current VQA models do not truly understand the visual information of images and what questions are asking about. On the other hand, producing answers to irrelevant questions would be *harmful* to user experience and mislead users by conveying misinformation that the premises in question are all correct.

More formally, irrelevant questions in the context of VQA can be defined by premises [Mahendru et al., 2017], which are facts implied by questions. For instance, the question "What's the black cat on the table doing?" implies the presence of a black cat, a table, and that the cat is on the table. Mahendru et al. [Mahendru et al., 2017] categorize premises into three classes of order. The first-order premises mean *the presence of objects* (e.g. a cat). The second-order premises reflect *attributes of objects* (e.g. a black cat) and the third-order premises are about *relations and interactions between objects* (e.g. a cat on a table). Once there is at least one false premise in a question, the question should be classified as an irrelevant question to the paired image. In the previous example, if there is a dog instead of a cat, or the cat is under the table in the image, the question is irrelevant to the image. In this case, if a VQA model still gives an answer like "sleeping", misinformation that there is "a black cat on the table" in the image would be conveyed to the asker.

Current approaches treat the VQA task as a multiclass classification problem. Given a question $q \in Q$ and an image $v \in V$, a VQA model is expected to give the ground truth answer $a^* \in A$ with the highest classification score

$$\hat{a} = \underset{a \in A}{\operatorname{argmax}}\, p_\theta(a|v, q), \tag{4.1}$$

where $\hat{a}$ is the predicted answer, and θ are the parameters of the trained model. The task of irrelevant question detection can be defined as a binary classification task. For a question-image pair (q, v), the task is to classify whether the question q is relevant to the image v.

In our approach, we have the hypothesis that *the abilities required for detecting irrelevant questions align with those required for answering visual questions*, and aim to answer the research question: *if we can endow a state-of-the-art neural network on existing VQA datasets with the ability to detect irrelevant questions by simply training on an extended VQA dataset with irrelevant questions?* Similar to the process of answering visual questions, judging whether a question is relevant to an image also requires a model to have a thorough and comprehensive understanding of both images and questions. To achieve this task, a model has to acquire information about classes of objects, colors, relative locations, counts, etc. Based on this hypothesis, using an end-to-end network architecture designed for the VQA task for detecting irrelevant questions is a more natural approach, in contrast to existing best-performing methods [Mahendru et al., 2017, Ray et al., 2016] which utilize separated image captioning models and MLP networks. In this work, we investigate the possibility of *solving the task of irrelevant question detection with a neural network designed for the VQA task*.

To integrate the ability to detect irrelevant questions into a VQA model, a straightforward idea is to train a VQA model jointly on a dataset containing both relevant cases and irrelevant cases by treating answers to irrelevant cases as "irrelevant". However, *interference between these two tasks is still unclear when jointly training them together*. Therefore we conducted several experiments to investigate this issue. We expect the performance of the joint model on both two tasks could be boosted based on our hypothesis. Our main findings of the research in this chapter are as follows:

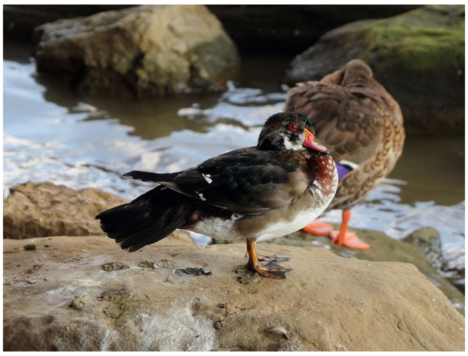
1. We demonstrate that the task of irrelevant question detection could be solved well by a neural network designed for the VQA task, and we set a new baseline performance on the QRPE dataset [Mahendru et al., 2017].

2. We empirically prove that the task of irrelevant question detection benefits from the techniques of iterative reasoning and relational modeling.

3. Our experimental results suggest that jointly training a VQA model on datasets extended with irrelevant cases sacrifices the accuracy of VQA on relevant cases, which implies that jointly training a VQA model on a mixed dataset with both relevant and irrelevant questions is not an ideal solution to address the issue of insufficient observations.

4.3 Related Work

Works of Ray et al.[Ray et al., 2016] and Mahendru et al. [Mahendru et al., 2017] are most related to ours. Ray et al. [Ray et al., 2016] firstly introduce the problem of irrelevant question detection in the context of VQA. They construct a dataset named Visual True and False Question (VTFQ) by showing annotators images paired with randomly selected questions and asking them to annotate whether the question is relevant to the corresponding image or not. Mahendru et al. [Mahendru et al., 2017] propose a premise extraction pipeline to automatically extract premise information from questions. In their paper, they give a formal definition of question premises and classify premises into the aforementioned three orders according to their complexity. A new dataset named Question Relevance Prediction and Explanation (QRPE) is constructed by them for the task of irrelevant question detection based on the premises of questions. The QRPE dataset contains irrelevant question-image pairs against the first-order or second-order premises. Fig. 4.1 and Fig. 4.2 display examples of question-image pairs where the content of the images conform to or contradict the premises of the visual questions. This dataset encompasses more ambiguous examples in comparison to the VTFQ dataset, which makes it more challenging. Several different methods have been proposed for detecting question relevance by Mahendru et al. [2017]. Their experimental results indicate that image captioning-based models have the best performance on this task. Though both of these papers briefly mention the benefits of integrating relevance detection into existing VQA systems, less attention has been devoted to relations between the relevance detection task and the VQA task.



(a) Relevant image

(b) Irrelevant image

Figure 4.1: An example case of the *first-order premise* failure in the QRPE dataset [Mahendru et al., 2017]. The corresponding visual question of the above two images is "Where are the birds standing?", where there is a first-order premise suggested by the visual question, i.e., the presence of objects "birds". The question is irrelevant and thus unanswerable given the image on the right due to the failure of the premise, i.e., there is no bird in the image.

Dealing with situations with irrelevant questions in the VQA task has potential application values. A natural application of VQA models is to help visually impaired people to perceive their environment. In contrast to normal users with good vision, users with impaired vision may take images that are of poor quality and may ask questions that are irrelevant to the image unintentionally. The VizWiz dataset [Gurari et al., 2018] is constructed with data from such an applica-

(a) Relevant image

(b) Irrelevant image

Figure 4.2: An example case of the *second-order premise* failure in the QRPE dataset [Mahendru et al., 2017]. The corresponding visual question of the above two images is "Where are the white buckets?", where there is a second-order premise suggested by the visual question, i.e., the presence of objects "buckets" with certain attributes "white". The visual question is irrelevant and thus unanswerable given the image on the right due to the failure of the premise, i.e., there is no white bucket in the image but only a black bucket.

tion scenario, where visually impaired users take images and ask visual questions regarding the image. Fig. 4.3 shows an example case of this dataset. It has been demonstrated that this dataset is very challenging for existing VQA models.

4.4 Irrelevant Visual Question Detection

4.4.1 Methodology

We choose the Multimodal Relational reasoning (MuRel) network [Cadene et al., 2019], one of the current state-of-the-art models on the VQA task, as our basic model. The use of the technologies of explicit iterative reasoning and relational modeling distinguishes MuRel from other networks. The two components associated with iterative reasoning and relational modeling in the network are the MuRel cell and the pairwise module. It has been shown that visual features play an important role in VQA performance [Anderson et al., 2018b, Jiang et al., 2018]. MuRel uses the bottom-up visual features [Anderson et al., 2018b] to represent images. specifically, an object detector Faster R-CNN [Ren et al., 2015] is used to extract region feature vectors to generate the bottom-up features of images. A pretrained skip-thought encoder [Kiros et al., 2015] is used for the question features extraction.

MuRel cell takes visual features and question features as inputs and produces updated visual features. The MuRel cell could be invoked several times to update visual features interactively. A pairwise module is an element of the MuRel cell. It obtains region features and coordinates of regions to model relations between

Figure 4.3: An example case sampled from the VizWiz dataset [Gurari et al., 2018]. The visual question corresponding to the above image is "Could you possibly tell me what the content of this jar is?". This question is unanswerable because a jar is not fully visible in the given image, making it impossible to determine its contents. In this case, an ideal VQA model is expected to abstain from answering instead of giving an answer like "food" or "water", which could cause harmful consequences for the user.

them. An efficient bilinear fusion module [Ben-Younes et al., 2019] works as the multimodal fusion strategy to combine visual and language information. A running process of the MuRel cell with the pairwise module is formalized as

$$\{s_i^t\} = MuRelCell(\{s_i^{t-1}\}, \{b_i\}, q), \tag{4.2}$$

where $t \in \{1, ..., T\}$ is the step number of the current process, s_i^t represents the updated representation of region i, b_i is the coordinate of region i and q is the representation of the input question. In the first step of the process (when $t = 1$), $s_i^0 = v_i$ exists, where v_i is the feature of region i of the visual features provided by the bottom-up features. After the last step of this process, when $t = T$, all s_i^T are aggregated together to provide a single vector s, which is then fused with question features q to produce a probability distribution $\hat{y}$ over all possible answers. This process can be formalized as

$$\hat{y} = B(s, q, \Theta_c), \tag{4.3}$$

where Θ_c are trainable parameters of the classifier.

We term MuRel-bin the MuRel relational reasoning network trained for relevance detection with a binary classifier. The network architecture of MuRel-bin is illustrated in Fig. 4.4. The network is applied to the task of irrelevant question detection. Inputs of MuRel-bin are bottom-up features of images and question fea-

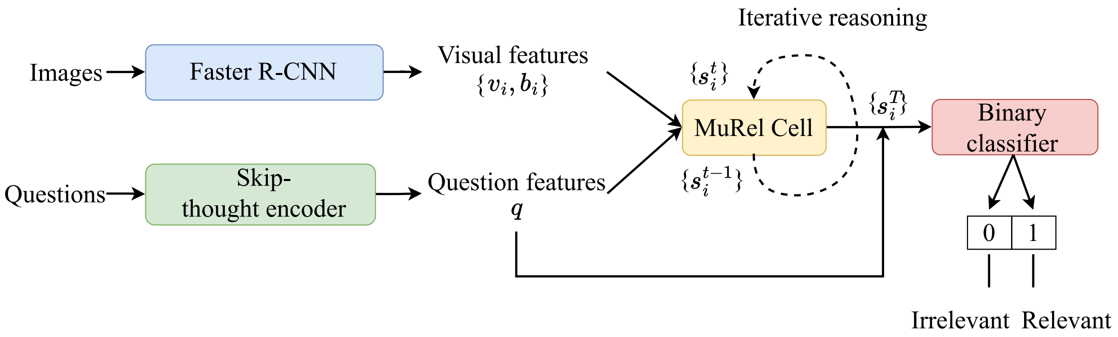


Figure 4.4: Illustration of the network architecture of MuRel-bin.

tures extracted by a skip-thought encoder. Labels of "relevant" and "irrelevant" in irrelevant question detection datasets are treated as two answers and correspond to two output neurons. Cross entropy loss is calculated to supervise the learning process.

4.4.2 Dataset

We use the QRPE dataset [2] [Mahendru et al., 2017] to evaluate the MuRel-bin model and compare it against other approaches on the task of irrelevant question detection. The QRPE dataset is curated automatically based on the MS COCO dataset [Lin et al., 2014], the Visual Genome dataset [Krishna et al., 2017], and the VQA v2 dataset [Goyal et al., 2017]. To build this dataset, first-order and second-order premises are firstly extracted from questions through a semantic tuple extraction pipeline used in the SPICE metric [Anderson et al., 2016] for evaluating the quality of image captions. For first-order premises, irrelevant images for a question are selected by checking the absence of the appropriate class label in the MS COCO annotations. For second-order premises, images that contain a matching object but a different attribute to the question premise according to annotations of Visual Genome are determined as irrelevant images. To ensure that the irrelevant image is similar enough to the relevant image, the one with the closest visual distance to the relevant image has been selected from irrelevant candidate images. In the end, every question in the QRPE dataset is paired with a relevant image and an irrelevant image. Compared to the VTFQ dataset, which is the first dataset for the task of irrelevant question detection, the QRPE dataset is balanced in the label space, larger, and constructed in finer granularity.

The training set of the QRPE dataset contains 35,486 irrelevant question-image pairs which are generated from the training set of the VQA v2 dataset. The test set of the QRPE dataset contains 18,425 irrelevant question-image pairs which are generated from the validation set of the VQA v2 dataset. Based on the order of the false premise, irrelevant cases can be divided into a first-order part and a second-order part. The number of irrelevant question-image pairs in the QRPE

[2] https://virajprabhu.github.io/qpremise/dataset/

Table 4.1: Number of irrelevant question-image pairs in the QRPE dataset.

Split	Overall	First-order	Second-order
Training set	35,486	32,939	2,547
Test set	18,425	17,096	1,329

dataset is shown in Table 4.1.

4.4.3 Experimental setup

Matching the experimental setup of existing methods we compare, we randomly select 90% of the training set of the QRPE dataset for training and the rest for validation. To avoid bias resulting from random division, we train 5 models independently and report the average accuracy on the test set as the final results. All MuRel-bin models are trained from scratch on the QRPE dataset. We performed some preliminary studies for training strategy and critical hyperparameters. We observed that overfitting problems can easily arise when inappropriate learning rates are applied. Finally, a similar learning scheduler as Cadene et al. [2019] with different settings is used in our training. We begin with a learning rate of $5e-6$, linearly increasing it at each epoch till it reaches $2e-5$ at epoch 6. Then we decrease the learning rate by a factor of 0.25 every 2 epochs from epoch 8 to epoch 14, at which we stop training. In our experiments, the batch size is set to 80, and experiments are conducted on 2 $\times$ NVIDIA Geforce 1080 TI.

4.4.4 Baseline Comparison

We compare MuRel-bin against state-of-the-art approaches on the QRPE dataset. The goal of this experiment is to evaluate whether a well-performing network designed for the VQA task can solve the task of irrelevant question detection well.

QC-Sim, PC-Sim, and QPC-Sim [Mahendru et al., 2017] are existing best-performing approaches on the QRPE dataset. QC-Sim uses an image captioning model NeuralTalk2 [Karpathy and Fei-Fei, 2015] pretrained on the MS COCO dataset to automatically generate natural language descriptions for images. An LSTM network is used to encode both the generated image captions and corresponding questions into vector representations. Then, question and caption representations are concatenated and fed into an MLP network to predict the relevance between questions and images. PC-Sim and QPC-Sim are variants of QC-Sim. PC-Sim uses automatically generated image captions and premises extracted from questions for relevance prediction. QPC-Sim considers all three sources, including questions, premises, and captions, for relevance prediction, and achieves the highest overall accuracy.

Results of MuRel-bin in Table 4.2 are achieved when the number of reasoning steps is set to 3 and the pairwise module is not used. Fig. 4.5 shows the training curves of MuRel-bin under this setting. The figure indicates that the model

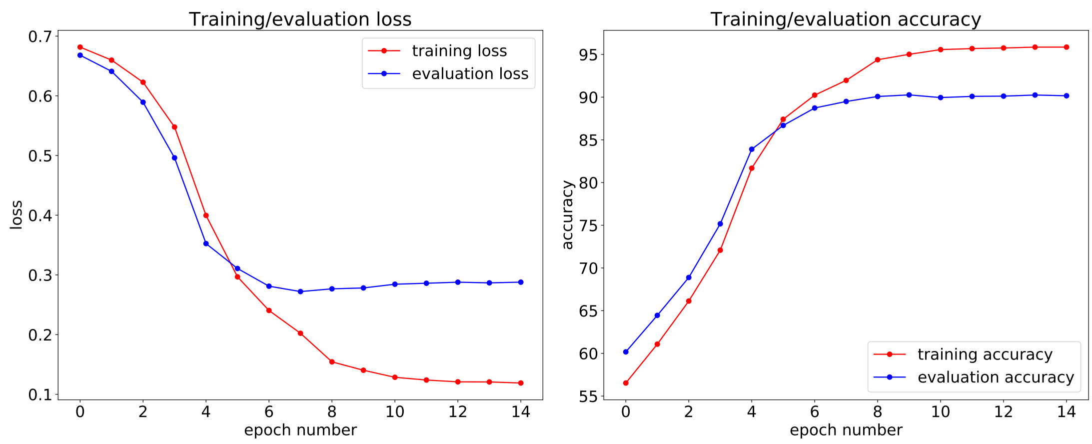


Figure 4.5: Training curves of MuRel-bin on the QRPE dataset.

Table 4.2: Comparison of accuracies on the QRPE dataset.

Models	Overall	First-order	Second-order
QC-Sim	74.35	75.82	55.12
PC-Sim	75.05	76.47	56.04
QPC-Sim	75.35	76.67	55.95
MuRel-bin	**86.62**	**88.13**	**67.02**

converges soon. After 8 epochs, an evaluation accuracy of around 90% is reached. We can notice that from epoch 8 on, the evaluation loss starts to increase slightly, which indicates that the model tends to overfit. A comparison of accuracy between MuRel-bin and other approaches on the overall and two splits of the test set of the QRPE dataset is shown in Table 4.2.

Results of QC-Sim, PC-Sim, and QPC-Sim are reported in their original paper [Mahendru et al., 2017]. The accuracy on the test set would be 50% if chosen at random since every question in the test set of QRPE is paired with a relevant and an irrelevant image. From Table 4.2, we can see that MuRel-bin outperforms existing best-performing approaches by a big margin (over 10%) both on the overall test set and each split divided according to the order of false premises. Therefore, we conclude that a well-performing network architecture designed for the VQA task can solve the task of irrelevant question detection well.

4.4.5 Ablation study

In this part, we investigate the effects of the techniques of multi-step reasoning and relational modeling on irrelevant question detection. Their contributions to the VQA task have been well proven [Cadene et al., 2019]. In Table 4.3, we compare four MuRel-bin models with different settings. To ensure comparability, we train them following the same experimental setup. The setting "Pairwise"

Table 4.3: Accuracies in the ablation study of MuRel-bin.

Pairwise	Iter.	Overall	First-order	Second-order
✗	✗	85.64	87.20	65.69
✓	✗	86.15	87.84	64.35
✗	✓	**86.62**	**88.13**	**67.02**
✓	✓	86.16	87.72	66.27

means whether the pairwise module is used and the setting "Iter." means whether iterative reasoning is used. In our experiments, the number of reasoning steps is set to 3 when iterative reasoning is used.

The results in Table 4.3 show that a MuRel-bin model with iterative reasoning but without the pairwise relational module achieves the best overall performance and the highest accuracies on both the first-order and second-order parts. The first three rows of Table 4.3 show that both iterative reasoning and relational modeling contribute to the MuRel-bin network's performance on the QRPE dataset, which is consistent with their benefits on the VQA task. However, comparing row 3 and row 4, we find adding the pairwise module to a model with iterative reasoning results in a loss of accuracy. We observed that the distance between training and evaluation loss curves increases when the pairwise module is used in this case, thus a possible explanation for this situation is using the iterative reasoning process and the pairwise module together leads to overfitting.

4.5 Integrating Irrelevant Question Detection into VQA

4.5.1 Methodology

In this part, we investigate the idea of integrating the ability to detect irrelevant questions into a VQA model by joint training a VQA model on a training set containing also irrelevant cases. For handling irrelevant cases, the model treats answers of irrelevant cases as a special answer "irrelevant". Based on our hypothesis that the abilities required for detecting irrelevant questions align with those required for answering visual questions, we expect that training data for these two tasks could benefit each other through joint training. The approach to joint training the MuRel network on an extended VQA dataset containing irrelevant cases is illustrated in Fig. 4.6.

4.5.2 Dataset

In extended training sets, irrelevant cases are annotated with the answer "irrelevant" for fitting VQA networks. In our experiments, we construct extended training sets based on the VQA v2 dataset, which is the most widely used VQA dataset.

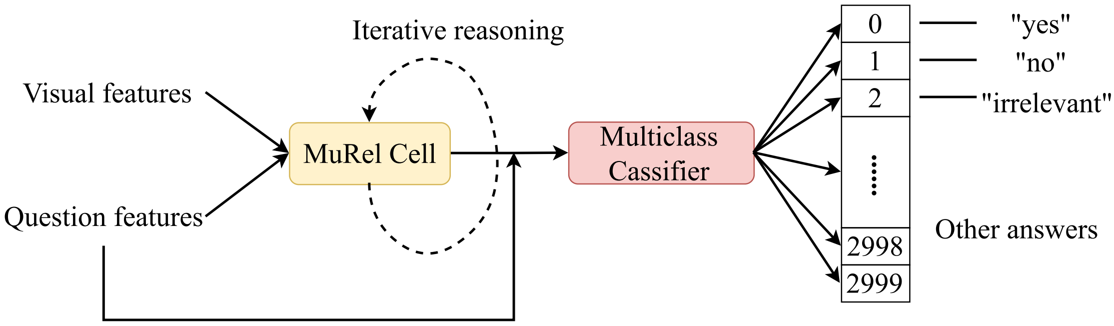


Figure 4.6: Illustration of the approach to jointly training the MuRel network on an extended VQA dataset containing irrelevant cases.

The VQA v2 dataset contains 443K, 214K, and 453K question-image pairs for training, evaluation, and testing, respectively. We denote the training set of the VQA v2 dataset as $VQAv2$ in the report of our experiments. We assume that all questions in the VQA v2 dataset are relevant to their corresponding images since human annotators are instructed to ask questions about the image that can be answered. First, we add 90% of irrelevant question-image pairs in the training set of the QRPE dataset to $VQAv2$ to build an extended training set $VQAv2+QRPE$. The reason why we only select 90% of irrelevant cases is to match the training setting in Section 4.4 for fair comparisons on the test set of the QRPE dataset. In $VQAv2+QRPE$, irrelevant cases account for 6.7% of all cases. To investigate the impact of different proportions of irrelevant cases, we construct another training set by adding all irrelevant cases in the training set of both the QRPE dataset and the VTFQ dataset. We denote this training set as $VQAv2+QRPE+VTFQ$, of which irrelevant cases account for 9.0%.

For $VQAv2$, the 3000 most frequent answers are selected as candidate answers. The top two most frequent answers are "yes" and "no", both of which occur over 80K times in the training set. Following them are answers "1" and "2", both of which occur over 10K times.

For $VQAv2+QRPE$ and $VQAv2+QRPE+VTFQ$, the special answer "irrelevant" is included in the 3000 candidate answers. In these two training sets, counts of the answer "irrelevant" are 31938 and 44024 respectively, which matches the numbers of irrelevant cases in them. Thus, in both of these two training sets, the answer "irrelevant" ranks between "no" and "1". The count of answer "irrelevant" is about half the count of answer "yes" and in the same order of magnitude as some other frequent answers.

4.5.3 Experimental setup

For experiments on joint training, we use a MuRel network with a pairwise module and a 3-step iterative reasoning process, because this setting achieves the best performance on the VQA v2 dataset. We adopt the same learning schedulers with the original MuRel model [Cadene et al., 2019] trained on the VQA v2 dataset. The

Table 4.4: Comparison of accuracies on the *test-dev* split of the VQA v2 dataset after joint training on different training sets.

Training set	Yes/No	Num.	Other	All
$VQAv2$	82.70	48.32	**56.13**	**66.19**
$VQAv2 + QRPE$	**83.03**	47.95	54.79	65.64
$VQAv2 + QRPE + VTFQ$	82.91	**48.35**	54.69	65.59

Table 4.5: Comparison of accuracies on the QRPE dataset after training on different datasets.

Training set	Accuracy
$QRPE$	86.16
$VQAv2 + QRPE$	**86.24**

starting learning rate is set to $1.5e-4$ with a batch size of 160. Models are trained for 25 epochs. Our experiments are conducted on 4 $\times$ NVIDIA Geforce 1080 TI. We train all models on different training sets following the same experimental setup to ensure comparability.

4.5.4 Results

Three MuRel models are trained on $VQAv2$, $VQAv2 + QRPE$, and $VQAv2 + QRPE + VTFQ$ respectively, and evaluated on the validation set of the VQA v2 dataset at every epoch. Checkpoints with the highest top 1 accuracy on the validation set are selected and tested on the *test-dev* split of the VQA v2 dataset for comparison. Scores of accuracy in Table 4.4 are calculated by the evaluation metric of the VQA Challenge [3] for all questions, "yes/no" questions, "number" questions, and other questions that are neither answered "yes/no" nor number.

From the accuracies reported in Table 4.4, we derive that jointly training a VQA model on training sets containing also irrelevant cases impairs its overall performance on the normal VQA data. As the proportion of irrelevant cases increases, the overall accuracy gradually decreases. We notice that the accuracy of "yes/no" questions can be improved when training on extended training sets.

We also test the MuRel model trained on $VQAv2 + QRPE$ on the test set of the QRPE dataset to see the impacts of joint training on the task of irrelevant question detection. To get the accuracy of this model on an irrelevant question detection dataset, we treat the answer of "irrelevant" as a prediction of irrelevance and other answers as a prediction of relevance. For a fair comparison, we take the same checkpoint that produces scores in Table 4.4 for testing on the QRPE test set. The overall accuracy achieved by this MuRel model on the test set of QRPE is 86.24%. This accuracy is a bit higher than the accuracy of 86.16% achieved by

[3] https://visualqa.org/evaluation.html

the MuRel-bin model with the same setting (see Table 4.5). It suggests that joint training can maintain accuracy on the task of irrelevant question detection well.

To avoid degradation on the VQA task when jointly training a model on a training set containing data for both VQA and relevance detection, we would like to suggest an alternative architecture. In this architecture, network layers for processing features of images and questions are shared for two tasks, while the output layers are separated. When the network is trained on irrelevant cases, parameters in output layers for the VQA task are not updated. This separation procedure might avoid unexpected interference with those tasks and reduce the overfitting problem.

4.6 Discussion

In this study, we investigated networks designed for VQA on the task of irrelevant question detection. A multimodal relational network for VQA was used for experiments. We demonstrated that the network adapted as a binary classifier outperforms the existing state-of-the-art methods by a large margin on the task of irrelevant question detection. It suggested that existing end-to-end neural networks-based models, which were originally designed for situations where task information is sufficient, are able to perform information sufficiency detection well when the model is specifically trained in a supervised learning manner for the task of detecting information sufficiency. The ablation study regarding the effectiveness of the components of iterative reasoning and relational modeling suggested that the relevance prediction task has the requirement for the reasoning ability and relational modeling ability as the VQA task has. This indicated that the abilities required for task-relevant information detection are not distinct compared to performing corresponding tasks when information is sufficient.

Following the aforementioned findings, we further investigated the idea of integrating the ability to detect irrelevant questions into a VQA model by training a VQA model on a training set containing irrelevant cases. The experimental results demonstrated that through supervised learning on mixed cases with both sufficient and insufficient observations, the model is able to gain the ability to detect information sufficiency, however, we noted that its previous ability on sufficient information settings is sacrificed. The performance degradation happens possibly because the change in data distribution makes it more challenging for the model to handle through supervised learning.

Future work may include building a larger and more difficult dataset for the task of irrelevant question detection. Though compared with the VTFQ dataset, the QRPE dataset is collected in a finer granularity by concerning different orders of false premises, it only contains irrelevant questions with false first-order and second-order premises and ignores irrelevant cases with false third-order premises concerning relations and interactions between objects. That makes current datasets unsuitable for the true challenges of the relevance detection task. In addition to the necessity of building new datasets, it is also promising to study methods of improving models' performance on the VQA task by taking advantage of the task of irrelevant question detection and vice versa. While we observed that jointly training a model on extended datasets containing also irrelevant cases leads to degradation of accuracy on the VQA task, we hypothesize that it may be possible to improve the performance by using other training methods, such as the method mentioned that shared layers are trained jointly while output layers are trained separately.

4.7 Summary

In this chapter, we attempt to answer the first research question "*How to endow a disembodied model with the capability of information sufficiency evaluation?*" Our experiments demonstrated that we can train a disembodied model explicitly on a dataset including additional "Irrelevant" labels through supervised learning to endow the model with the capacity of information sufficiency evaluation. The resulting model responds with the special answer "Irrelevant" instead of producing normal answers, such as "black" to questions regarding the colors of objects when the information is deemed insufficient. However, this response is not very helpful to the users. We revealed the inherent limitations of disembodied AI models in situations of insufficient observations, i.e., only a conservative model is possibly achieved to prevent producing wrong and potentially harmful responses. To make responses more helpful to the users, the issue of insufficient observations should be tackled fundamentally by actively collecting more useful information. This motivated our follow-up work on embodied agents with the capability of active vision, with which a progressive model can be obtained to actively collect necessary information for producing responses that are more helpful to the users.

Chapter 5

Learning Active Vision Control Using Reinforcement Learning

5.1 Introduction

In this chapter, we will be focusing on embodied AI agents with active vision. This is driven by the inherent limitations of disembodied AI models, namely, their inability to collect necessary information in situations with insufficient observations. Specifically, we will be working on action-response embodied agents that have dual-output channels: the output of the active vision action channel collects necessary information, and the output of the task-relevant response channel produces the response based on information collected through active vision (cf. Fig. 1.2).

Existing tasks designed for action-response agents are overly complex when serving as evaluation tasks for research purposes. For instance, tasks like embodied question answering [Das et al., 2018] and interactive question answering [Gordon et al., 2018] involve large-scale indoor environments, demanding extensive abilities from the agents, such as commonsense reasoning, long-term action planning, and memory. However, such comprehensive requirements and high-level complexity hinder our understanding of the fundamental mechanism and development of active vision control and corresponding response for embodied AI agents. A task with a reasonable level of complexity is needed for effectively evaluating the performance of such models and their training methods. In line with this requirement, we designed a robotic task named robotic object existence prediction (ROEP), where the situation of insufficient observations arises from potential occlusions between objects. This task can be seen as a simplified version of embodied question answering, characterized by limited state and action spaces.

Inspired by the recurrent attention model [Mnih et al., 2014], we are interested in exploring the feasibility of developing end-to-end modular networks, where the active vision policy module and the task-relevant response module are optimized simultaneously. The idea of simultaneous optimization of the two modules is motivated by our insight that the outputs of active vision action and task-relevant responses functionally collaborate and interfere with each other. Aligning with the training strategy of the recurrent attention model, we attempt to train the active vision control policy using reinforcement learning, while the task-relevant response module is going to be trained through supervised learning.

5.2 Robotic Occlusion Reasoning

Indoor assistant robots that are able to perform tasks, such as searching for objects and answering questions about the environment, according to verbal commands from users have promising application prospects. We expect robots to not only complete these tasks correctly but also complete them efficiently, which benefits improving user experience and reduces energy requirements.

The ability to reason about potential occlusions of objects is essential for achieving the aforementioned goal. When asked to search for an object, a robot needs to reason whether the target object is possibly occluded by visible objects, and then determine whether to check the occluded space by executing movement actions. However, occlusion reasoning is non-trivial: a robot needs to know the size of the target object from the verbal instruction and compare it with the size of the visible objects to perform occlusion reasoning. Though existing work has shown that robots with active perception can achieve various tasks [Zhu et al., 2017, Ye et al., 2018, Wang et al., 2018, Yang et al., 2019a], in this work we further investigate if robots can efficiently explore environments by performing occlusion reasoning.

To answer this question, we propose a novel robotic object existence prediction (ROEP) task. Fig. 5.1 shows the task in real scenarios and a simulation environment that is built using the robot simulator CoppeliaSim [Rohmer et al., 2013]. The robot is the humanoid Pepper[1] from SoftBank Robotics, which has three omnidirectional wheels for flexible locomotion. The movement of the robot is implemented as a circular motion around the table by 30 degrees clockwise or anticlockwise. The robot receives a word instruction (e.g. "marble") and is rewarded for correctly predicting whether the target object exists on the table while executing as few movement steps as possible. There are three main challenges behind achieving this goal: 1) the robot needs to connect linguistic concepts with visual representations; 2) the robot needs to memorize past interactions with the environment to make action selection decisions; and 3) the selected actions and the final prediction functionally interact with each other, which makes the training difficult.

We propose a novel model (see Fig. 5.2) to address the above challenges. This model is a recurrent neural network consisting of five modules: a visual perception module, a word embedding module, a memory module, an action selection module, and an existence prediction module. The model can be jointly trained with reinforcement learning and supervised learning methods using a curriculum training strategy [Bengio et al., 2009].

We evaluate our model by comparing it with three baselines: a passive model without any movement, a random model with a stochastic movement selection

[1] https://www.softbankrobotics.com/emea/en/pepper.

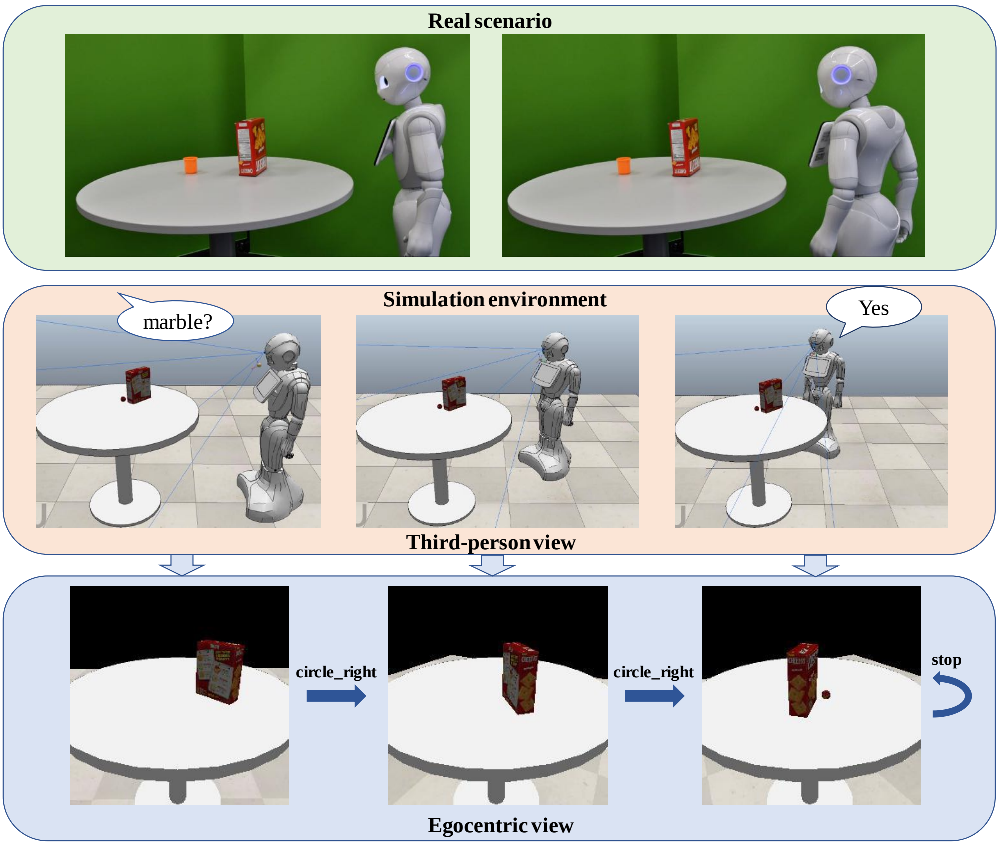


Figure 5.1: The task of robotic object existence prediction: given a word instruction (e.g. "marble"), a robot standing by a table needs to execute as few movement steps as possible to give a correct prediction (e.g. yes) whether the queried object exists on the table.

strategy, and an exhaustive exploration model that takes a maximum number of movements. Experimental results demonstrate that our model can outperform the passive and random baselines by a large margin, and achieve a similar prediction accuracy to the exhaustive exploration model while requiring only about 10% of the baseline's number of movement steps on average. This shows the necessity of active perception and occlusion reasoning to successfully achieve the task, and that a good occlusion reasoning ability is obtained by our model.

As the number of different objects increases, the number of possible combinations of two objects with occlusion increases exponentially. So a good generalization performance on novel combinations of two objects is especially important for occlusion reasoning. We evaluate the generalization performance of our model on novel object combinations held out from the training data, where we show that the generalization to novel object combinations comes with a moderate loss of accuracy while maintaining a small average number of movement steps. Moreover, the generalization performance increases when more kinds of object combinations are included in the training data.

The main contributions of the work in this chapter can be summarized as follows: 1) we formulate a novel robotic object existence prediction (ROEP) task, which poses a high requirement of active perception and occlusion reasoning ability for robots; 2) we develop a novel model that can efficiently achieve this task; and 3) we find that the proposed model generalizes to novel object combinations with a moderate loss of accuracy and that the variety of object combinations in the training data benefits generalization.[2]

5.3 Related Work

Mobile robots with active perception: Zhu et al. [2017] proposed a reinforcement learning model for the task of target-driven visual navigation. The model is expected to navigate towards a visual target in indoor scenes with a minimum number of movement steps by its egocentric visual inputs and the image of the target. Ye et al. [2018] studied the problem of mobile robots searching small target objects in arbitrary poses in indoor environments. They proposed a model integrating an object recognition module and a deep reinforcement learning-based action selection module together for the object searching task. Wang et al. [2018] focused on the efficiency of robots when searching for target objects. They proposed a scheme to encode the prior knowledge of the relationship between rooms and objects in a belief map to facilitate efficient searching. Instead of focusing on achieving tasks in large-scale indoor environments, we concentrate on the efficiency of robotic active vision when encountering a specific tabletop occlusion situation.

Object occlusion: The occlusion situation between objects is very common in robotic scenarios. However, the occlusion reasoning ability of autonomous mobile robots has not been studied well. Yang et al. [2019a] introduced the task of embodied amodal recognition focusing on the visual recognition ability of agents in scenes with occlusion. They proposed a model that can navigate in the environment to perform object classification, location, and segmentation. However, this work did not concentrate on the occlusion reasoning ability of the agent. A recent work on developing robots with occlusion reasoning ability is [Deng et al., 2021]. This work introduced the task of answering visual questions via manipulation (MQA), where a robot manipulator needs to perform a series of actions to move objects possibly occluding some small objects on a tabletop, in order to correctly answer visual questions. Similar to the ROEP task, the MQA task also requires the robot to have the ability of occlusion reasoning to perform reasonable exploration actions. However, the robot in MQA is a manipulator that explores the environment by moving objects, while we focus on autonomous mobile robots to explore the environment by active perception.

Embodied learning: Robotics research is recently benefiting from achievements in vision and language processing. On the other hand, researchers are also

[2]Code for reproducing the experimental results reported in this chapter is open-sourced at `https://github.com/mengdi-li/robotic-occlusion-reasoning`. A video showing the experimental results is available at `https://youtu.be/L4p7yo8dMmQ`.

Table 5.1: Objects used in the simulation environment

Category	Objects			
Large	cracker_box	cleanser	laptop	pitcher
	desktop_plant	wine	teddy_bear	
Medium	apple	baseball	foam_brick	mug
	rubiks_cube	meat_can	coffee_can	
Small	bolt	dice	key	marble
	card	battery	button_battery	

taking advantage of agents situated in 3D environments to conduct multimodal research. It has been proven that an active agent is able to connect linguistic concepts with visual representations of the environment through training to complete action-involved tasks [Hermann et al., 2017, Chaplot et al., 2018]. Hill et al. [2021] found that an embodied agent can achieve one-shot word learning when trained with reinforcement learning in a 3D environment. The proposed ROEP task also involves multimodalities, including vision, language, and action. Different from the abovementioned work, our model needs to specifically connect linguistic concepts with visual representations of object size through training to achieve the ROEP task.

5.4 Task Setups

5.4.1 Simulation Environment

Existing simulation environments are not suitable for the ROEP task. We create a corresponding tabletop simulation environment using the robot simulator CoppeliaSim [Rohmer et al., 2013] (see Fig. 5.1). The robot can capture egocentric RGB images by a visual sensor mounted on its head, and execute actions selected from (*circle_left*, *circle_right*, and *stop*). By taking the action *circle_left*, the robot circles around the table clockwise by 30 degrees. The action *circle_right* works in the same way but in an anticlockwise direction. When the action *stop* is selected or the maximum number of 6 movement steps is reached, the robot takes no movement action and predicts whether the queried object exists.

A total of 21 everyday objects are used in the simulation environment. Some of them are from the YCB dataset [Calli et al., 2015]. The rest of them are provided by CoppeliaSim or collected online. These objects are divided into 3 categories according to their relative size, as shown in Table 5.1. When fitting these objects into cubes, objects from the *Large* category have a minimum height of $21cm$ and an average volume of $2905cm^3$. The heights of objects from the *Medium* category are from $5cm$ to $14cm$, and their average volume is $508cm^3$. Objects from the *Small* category have a maximum height of $3cm$ and an average volume of $7cm^3$. There are potential occlusions of objects from different categories.

Table 5.2: Reasoning table

	Query		
Visible Object	*Large*	*Medium*	*Small*
One *Large*	predict	move	move
One *Medium*	predict	predict	move
One *Small*	predict	predict	predict

5.4.2 Data Generation

Our data is automatically generated based on predefined rules like the development of the CLEVR [Johnson et al., 2017] and the ShapeWorld [Kuhnle and Copestake, 2017] datasets. All the samples are generated on-the-fly during training and testing periods. Each data sample is a triplet [*Scene*, *Query*, *Prediction*]. *Scene* is an arrangement of objects on the table. *Query* is a word randomly selected with equal probability from Table 5.1 to instruct the robot to search for the referred object in *Scene*. *Prediction* is a ground-truth binary label representing whether the target object exists in *Scene*. It is randomly set as positive or negative with an equal probability of 50%. Based on a determined pair of *Query* and *Prediction*, a corresponding scene is then generated.

There are three different types of scenes: 1) scenes that contain one object; 2) scenes with two objects without occlusion from the initial field of view of the robot; and 3) scenes with two objects, one of which is occluded by the other one from the initial field of view of the robot. They account for the same proportion (1/3) in the generated data. To generate scenes with one object, the object is randomly placed on the table. To generate scenes with two objects, some geometric calculations using the position coordinates of the robot's visual sensor, and both position coordinates and heights of the two objects are applied to control whether there are occlusions in generated scenes. It should be noted that the smaller object is not necessarily fully occluded by the larger one in scenes with occlusion.

We have a reasoning table (see Table 5.2) of the ideal action strategy at the first time step in an episode. This table shows whether the robot should move to change its viewpoint or predict the existence of the target object directly when given a query for objects of a specific category (different columns), and the object seen from the initial viewpoint. Except for the situation where a *Large* object is queried, or a *Small* object is seen, the robot has to utilize both information from the word instruction and visual perception to make an ideal action decision. Because there are at most two objects on the table, whenever the robot sees two objects, the robot should give an existence prediction directly no matter which object is queried.

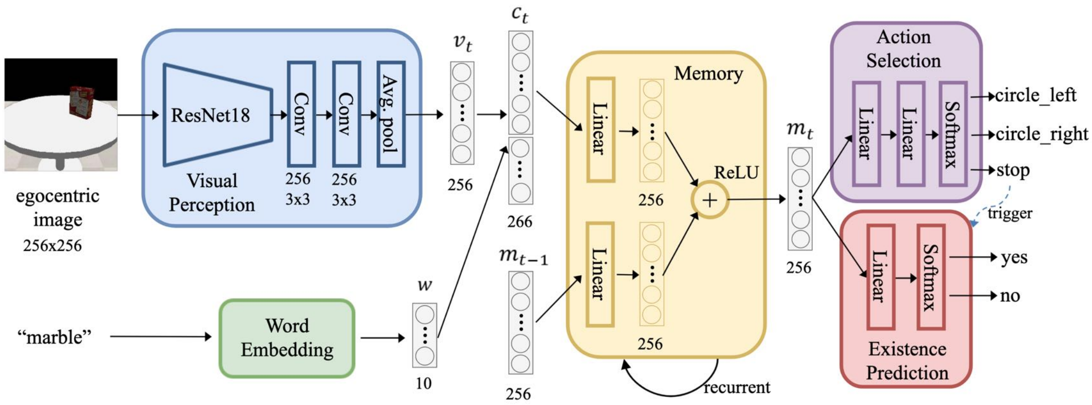


Figure 5.2: The architecture of the proposed occlusion reasoning model. The *Action Selection* module is trained through **reinforcement learning**, while *other modules* are updated using **supervised learning**.

5.5 Methodology

5.5.1 Model

Our proposed model is inspired by the recurrent attention model [Mnih et al., 2014], which was originally applied to attention-driven image classification tasks. The proposed model is a recurrent neural network overall (see Fig. 5.2), and can be divided into five parts: 1) a memory module for incrementally building up state representations, 2) a visual perception module for extracting visual representations, 3) a word embedding module for extracting distributed representations of a query word, 4) an action selection module for making action decisions, and 5) an existence prediction module for producing final predictions.

The **visual perception** module takes the egocentric RGB image (256 $\times$ 256 pixels) as input to extract visual representations. It first extracts the 128 28$\times$28 image feature maps from the *conv3* layer of a fixed ResNet18 [He et al., 2016] pretrained on ImageNet [Deng et al., 2009]. The feature is then passed through two CNN layers both with 256 3$\times$3 kernels, and an average pooling layer to obtain the visual representations v_t with a length of 256. This process is similar to the visual module of the MAC model [Hudson and Manning, 2018] designed for visual reasoning on the CLEVR dataset [Johnson et al., 2017].

The **word embedding** module maps each word instruction to a 10-dimensional word vector w. The weights of the embedding module are randomly initialized, and updated during training.

The **memory** module is a recurrent unit that takes the concatenated representations $c_t = (v_t, w)$ as the input, and combines c_t with the internal representations at the previous time step m_{t-1} to produce the new internal representations $m_t \in \mathbb{R}^{256\times 1}$. This process can be formalized as

$$m_t = f_m(m_{t-1}, c_t) = \text{ReLU}(W_m \cdot m_{t-1} + W_c \cdot c_t + b) \tag{5.1}$$

where $W_m \in \mathbb{R}^{256\times256}$ and $W_c \in \mathbb{R}^{256\times266}$ are weight matrices, $b \in \mathbb{R}^{256\times1}$ is a bias vector, $\mathrm{ReLU}(\cdot)$ is the rectified linear activation function. More sophisticated units such as LSTM or GRU are not used for the memory module because a vanishing gradient is not a problem for our task since only a small number of recurrent steps have to be taken.

The **action selection** module and **existence prediction** module are both classification networks with *softmax* outputs. The action selection module is a fully connected network with one hidden layer (128 hidden units). Its three *softmax* outputs correspond to three movement actions. The existence prediction module has a single linear layer followed by a *softmax* layer with two outputs that correspond to the positive and negative prediction respectively.

5.5.2 Model Optimization

The parameters of our model include parameters of the visual perception module, the word embedding module, the memory module, the action selection module, and the existence prediction module $\theta = \{\theta_v, \theta_w, \theta_m, \theta_a, \theta_p\}$. The model is non-differential overall. We train the model jointly with supervised learning and reinforcement learning methods, where θ_a is trained using reinforcement learning, $\{\theta_v, \theta_w, \theta_m, \theta_p\}$ are trained using supervised learning.

The task can be formalized as a partially observable Markov decision process from the perspective of reinforcement learning. The true state of the environment cannot be fully observed. The action selection module is a reinforcement learning agent, which needs to learn a stochastic policy $\pi(a_t|s_{0:t};\theta_a)$ with the parameters θ_a, where a_t is one of the three actions in the predefined action set. Executing each movement action except the *stop* action leads the model to obtain a new visual input. $s_{0:t} = w, v_0, a_0, v_1, a_1, ..., v_t$ is the history of past interactions with the environment from time step 0 to t. The internal representations m_t in the memory module is an approximation to $s_{0:t}$.

The model is expected to gain a high reward at the end of each episode. We design a cost-sensitive reward function containing two parts, an accuracy reward r_{acc} and a latency reward r_{lat}. An accuracy reward of 1 is received when a correct prediction is produced. An accuracy reward of -1 is received when an incorrect prediction is produced. The latency reward is

$$r_{lat} = \frac{1}{T+2} \tag{5.2}$$

where T is the number of movement steps the agent takes in one episode. $T = 0$ means that the *stop* action is selected at time step 0. The total reward at time step T is a summation of these two rewards: $r_T = r_{acc} + r_{lat}$. We use $T + 2$ rather than $T + 1$ as the denominator of r_{lat} to make sure that r_T is negative when the prediction is incorrect. At other time steps ($t = 0, ..., T-1$), we set $r_t = 0$.

The agent is expected to maximize the expected reward return $J(\theta_a)$ under the policy $\pi(a_t|s_{0:t};\theta_a)$.

$$J(\theta_a) = \mathbb{E}_{\pi(a_t|s_{0:t};\theta_a)}\left[\sum_{t=0}^{T} r_t\right] \tag{5.3}$$

We use Monte-Carlo policy gradient (REINFORCE) [Williams, 1992] to optimize the agent. REINFORCE uses the sample gradient to approximate the actual gradient of $J(\theta_a)$

$$\nabla_{\theta_a} J \approx \sum_{t=0}^{T} \nabla_{\theta_a} \log \pi(a_t|s_{0:t};\theta_a)(R_t - b_t) \tag{5.4}$$

where $R_t = \sum_{t'=0}^{T} r_{t'}$ is the accumulated reward following the action a_t, b_t is the estimated reward predicted by a baseline network, which has a single linear layer taking m_t as the input. The estimated reward b_t is used for reducing the variance of gradient estimation. The baseline network is trained with a mean squared error loss $\mathcal{L}_b = \frac{1}{T}\sum_{t=0}^{T}(R_t - b_t)$.

To use gradient descent algorithms for optimizing the agent, we define loss $\mathcal{L}_a = -J(\theta_a)$. It should be noted that gradients of $\mathcal{L}_a$ and $\mathcal{L}_b$ are not backpropagated to the memory, visual perception, and word embedding module.

We train these modules along with the existence prediction module using supervised learning methods to optimize the binary cross-entropy loss

$$\mathcal{L}_p = -y \log \hat{y} - (1-y)\log(1-\hat{y}) \tag{5.5}$$

where y is the labeled ground-truth prediction (1 for yes, 0 for no), $\hat{y}$ is the estimated probability of the prediction yes. Gradients of $\mathcal{L}_p$ are backpropagated to update parameters of the existence prediction θ_p, memory θ_m, visual perception θ_v, and word embedding θ_w module.

The total loss function is a weighted summation of the three losses, as

$$\mathcal{L}_{total} = \mathcal{L}_p + \alpha \cdot \mathcal{L}_a + \beta \cdot \mathcal{L}_b \tag{5.6}$$

where α and β are weight coefficients of $\mathcal{L}_a$ and $\mathcal{L}_b$ respectively.

5.5.3 Training Details

We found that it is hard to train the model from scratch on data with all three different types of scenes, which corresponds to the finding of [Yang et al., 2019a] that joint training perception and policy networks from scratch is difficult. We resort to a curriculum training strategy to train the model on data with 4 levels of increasing difficulty. We refer to data only containing scenes with one object as *L1-1-vis*, data only containing scenes with two objects without occlusion as *L2-2-vis*, data only containing scenes with two objects with occlusion as *L3-2-occ*, and data containing all types of scenes as *L4-overall*, in which three types of scenes occupy the same proportion. The model is trained on these four levels of data sequentially. The parameters obtained from one training stage are loaded as the initial parameters for the next training stage.

Table 5.3: Performance evaluation on different test data

	$Model_{L1}$		$Model_{L2}$		$Model_{L3}$		$Final\ Model$	
Test Data	Acc.	Steps	Acc.	Steps	Acc.	Steps	Acc.	Steps
L1-1-vis	**99.4%**	0.0	90.9%	0.02	88.7%	1.20	99.0%	0.74
L2-2-vis	68.3%	0.0	**97.4%**	0.01	91.2%	0.64	97.1%	0.39
L3-2-occ	74.4%	0.0	77.7%	0.07	**98.3%**	1.23	96.9%	1.02
L4-overall	80.8%	0.0	88.5%	0.03	92.6%	1.00	**97.2%**	0.71

We use the Adam optimizer with a learning rate of $1e-4$. The weight coefficients in the total loss function (cf. Eq. (5.6)) are set as $\alpha = 1e-2$, $\beta = 1$ for the training stage on the first three levels. A smaller weight coefficient $\alpha = 1e-4$ is used for the last training level to stabilize the training process.

5.6 Experiments

5.6.1 Curriculum Training

Our model is trained using a curriculum training strategy. Specifically, the model is trained sequentially on *L1-1-vis*, *L2-2-vis*, *L3-2-occ*, and *L4-overall* data with a fixed number of episodes ($900k$, $900k$, $400k$, and $400k$ respectively) in our experiments. The total training process takes about four days using one GPU (NVIDIA Titan RTX). We noticed that it is unnecessary to train the model to achieve the best performance in the first three training stages if we are only interested in the final model. We repeat the experiment three times to avoid the effect of randomness. The accuracy of correct predictions and the average number of movement steps are used as metrics to evaluate the performance.

Fig. 5.3 shows the training curves in different training stages. In the first two training stages on *L1-1-vis* and *L2-2-vis* data, the accuracy increases stably until reaching a plateau of over 97%, while the average number of movement steps stays near 0. In the third training stage on *L3-2-occ* data, the accuracy rapidly increases in the first $30k$ episodes with the rapid increase of the average movement steps. In the last two training stages on *L3-2-occ* and *L4-overall* data, the average movement steps continuously decrease after the accuracy has reached a plateau.

We refer to models obtained from the first three training stages at the $900k$, $900k$, and $400k$ episodes as $Model_{L1}$, $Model_{L2}$, and $Model_{L3}$ respectively. The final model is obtained from the last training stage at the $400k$ episodes, and denoted as *Final Model*. The performance of each model, when tested on different test data ($10k$ episodes), is presented in Table 5.3. The results show that each model scores well on the test data that corresponds to the training statistics (diagonal in bold font) and that the final model performs nearly as well as the individual models on their test data. Fig. 5.4 shows examples when there is only one object, which is larger than the target object, visible from the initial perspective of the agent. A video showing the experimental results is available at https://youtu.be/L4p7yo8dMmQ.

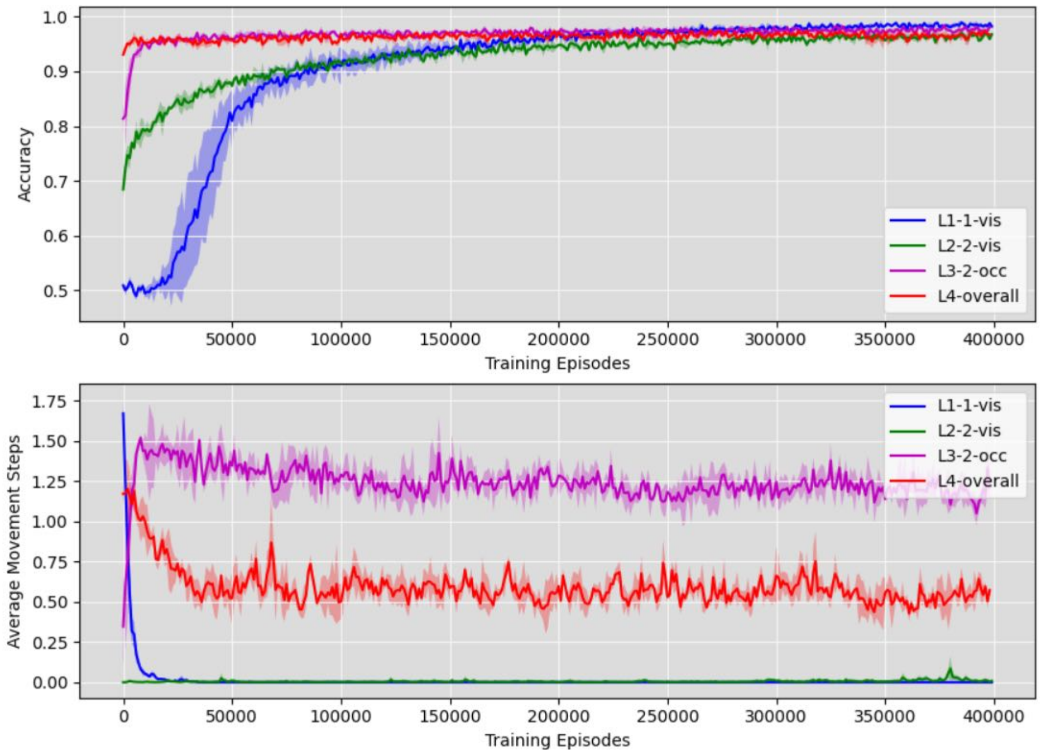


Figure 5.3: Training curves of the proposed model in different training stages. The model is sequentially trained on *L1-1-vis*, *L2-2-vis*, *L3-2-occ*, and *L4-overall* data. The parameters obtained from one training stage are loaded as the initial parameters for the next training stage.

5.6.2 Baseline Comparison

We compare the proposed model with three baselines that have the same architecture as the proposed model, but with different action selection strategies. These baselines include a passive model without any movement, a random model with a stochastic movement selection strategy, and an exhaustive exploration model that executes the *circle_left* action for a maximum number of movement steps before producing a prediction. The average movement steps of the three baselines are 0, 1.82, and 6 respectively.

The prediction accuracy of these baselines and our final model when tested on different test data is presented in Table 5.4. The passive model and the random model are able to achieve a performance close to that of the exhaustive model on *L1-1-vis*, and *L2-2-vis* data, but perform poorly on *L3-2-occ* data. This reveals that active perception is necessary to address the ROEP task. Our model can achieve a similar accuracy on all test data to the exhaustive model while requiring only 11.8% of the baseline's number of movement steps on average (0.71 steps by our model, 6 steps by the exhaustive model). This demonstrates that our model has obtained a good occlusion reasoning ability. However, there are still some challenges remaining: 1) The model learns to always choose one direction

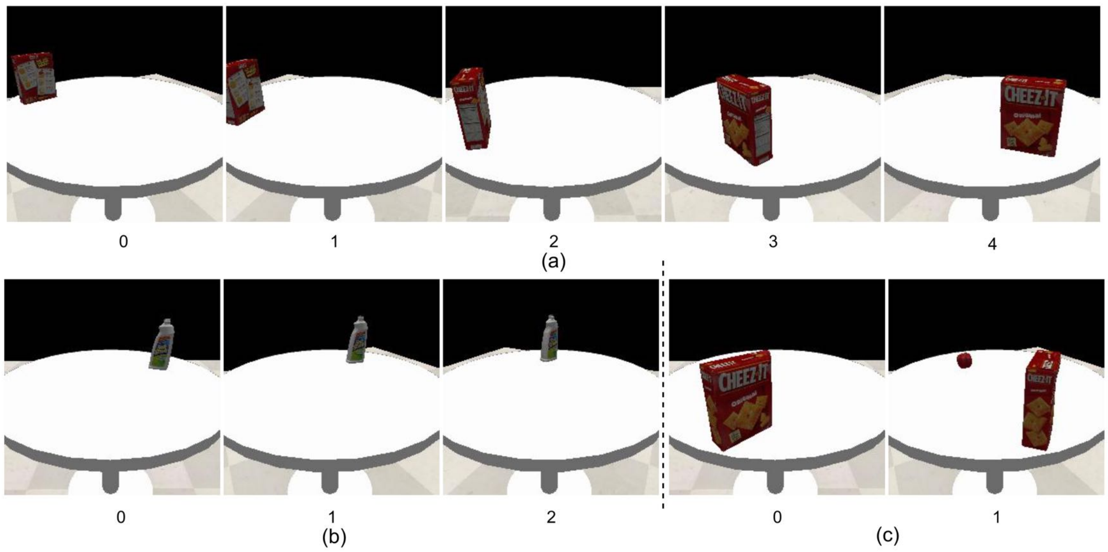


Figure 5.4: Examples of egocentric images in an episode when the query is "apple". Numbers below the images indicate the time steps in an episode. (a), (b): only one object larger than the target object exists; (c): the target object is occluded by a larger object. In all cases, the agent moves to check the occluded space and provides the correct answer after the last shown frame.

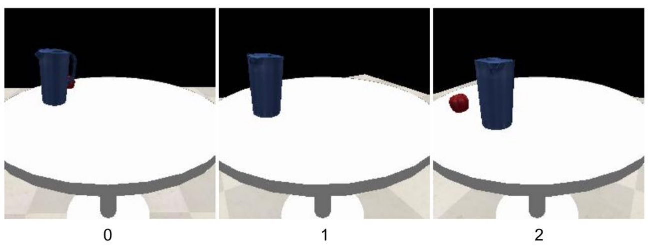


Figure 5.5: In this example, an apple is partially occluded by a pitcher. When the query is "apple", the agent does not choose the optimal action *circle_right*, instead it chooses the action *circle_left*.

to move, rather than choose the optimal direction according to the orientation of the visible object or partial occlusion to check the occluded space (see Fig. 5.5); 2) The model moves 0.39 steps on average in scenes without occlusion (*L2-2-vis*), which is unnecessary.

5.6.3 Generalization Evaluation

A good generalization performance on novel combinations of two objects is especially important for occlusion reasoning. To evaluate the generalization performance, we train our network on two different sets of training data excluding some object combinations, which are called holdout combinations. That means scenes with some specific object combinations, e.g. [mug, battery], are not included in

Table 5.4: Performance comparison with baselines

Test Data	Passive Model	Random Model	Exhaustive Model	Our Model
L1-1-vis	98.0%	97.6%	**99.2%**	99.0%
L2-2-vis	96.1%	92.6%	96.4%	**97.1%**
L3-2-occ	77.6%	83.3%	**97.4%**	96.9%
L4-overall	90.3%	91.1%	**97.4%**	97.2%

Table 5.5: Generalization Evaluation

	21 holdout		*42 holdout*	
Test Data	Acc.	Steps	Acc.	Steps
L1-1-vis	98.6%	0.651	98.8%	0.629
L2-2-vis (training)	96.8%	0.262	97.2%	0.197
L3-2-occ (training)	94.7%	0.873	96.2%	0.892
L2-2-vis (holdout)	95.8%	0.399	92.7%	0.317
L3-2-occ (holdout)	91.5%	0.895	86.7%	0.841

the training data.

There are three types of combinations of two different size categories, namely [*Large*, *Medium*], [*Large*, *Small*], [*Medium*, *Small*], and 147 ($7 \times 7 \times 3$) possible combinations of two objects from different size categories. In the first training set, 21 object combinations (7 for each category combination) are held out only for testing, which accounts for 14.3% of all possible combinations. In the second set, 42 object combinations (14 for each category combination) are held out, which accounts for 28.6% of all possible combinations. Holdout combinations are determined by randomly selecting from all possible object combinations before the start of training. Every object in Table 5.1 is shown in the training data. Experiments are repeated three times with different holdout combinations and random initialization.

Table 5.5 presents the test results of the models trained on the aforementioned two sets of training data, denoted as *21 holdout* and *42 holdout* respectively. Test data *L2-2-vis (training)* and *L3-2-occ (training)* contain scenes with object combinations used for training. Test data *L2-2-vis (holdout)* and *L3-2-occ (holdout)* only contain scenes with holdout object combinations. The results show that the two models can achieve similar high performance on scenes with object combinations used for training. When tested on *L2-2-vis (holdout)* and *L3-2-occ (holdout)*, the model trained on *21 holdout* can still work well with an accuracy of over 90% and a small average number of movement steps. The performance of the model trained on *42 holdout* drops moderately to 86.7% accuracy when tested on *L3-2-occ (holdout)*, where occlusion reasoning on novel combinations is necessary.

5.7 Discussion

Experimental setup: The experimental setup of the task of object existence prediction described in this chapter is simplified. There is a strong prior that there are at most two objects existing on the table, which limits the complexity of potential occlusion situations. An interesting extension is to generate scenes with more objects on the table and extend the task to counting objects. Moreover, the action space of the robot is small. The actions of *circle_left* and *circle_right* used in the current experimental setting limit the generalization capability to environments with tables of different sizes or shapes. More complex robot actions such as *move_ahead*, *rotate_left*, *rotate_right* can be included in the action space in future work. On the one hand, it makes it feasible to transfer a robot with these more complex actions to other environments. On the other hand, it makes the task more challenging, as the robot has greater flexibility in its movements, which places higher demands on action planning.

Training complexity: The current training process is complex since the curriculum training strategy involves four sequential training stages to obtain the final model. A possible solution to simplify training is using unsupervised learning [Ha and Schmidhuber, 2018] instead of curriculum learning to learn good visual and word representations. Another possible solution to make training easier is to strengthen the functional coordination between the reinforcement learning and supervised learning processes, which will be studied in Chapter 6. An LLM-based approach, which does not require any task-specific training, for this task is explored in Appendix A.

Sim-to-real transfer: In this work, we validate the effectiveness of the proposed model in a simulation environment. We can imagine that directly transferring the resulting model trained in a simulation environment to a real-world scenario (see Fig. 5.1) would result in a certain performance loss. Some techniques, such as fine-tuning the model in a more photo-realistic simulation environment with randomized lighting conditions of the real environment, may mitigate the performance degradation.

5.8 Summary

In this chapter, we attempt to answer the second research question "*How to model action-response embodied agents using neural networks and optimize the active vision control policy using reinforcement learning?*" To answer this question, we introduced the task of robotic object existence prediction (ROEP), which has a moderate complexity and serves as an appropriate evaluation task for the development of action-response agents. We model the action-response agent for this task with a novel recurrent neural network that was trained jointly with reinforcement learning and supervised learning methods using a curriculum training strategy.

We empirically demonstrated that the proposed model can efficiently achieve the ROEP task compared with the baselines. We also showed that generalization to novel object combinations comes with a moderate loss of accuracy while including more kinds of object combinations in the training data can increase the generalization performance. This finding, which is related to the finding in [Hill et al., 2020], can be considered as a recommendation when training a model for tasks that implicitly involve occlusion reasoning (e.g., object goal navigation [Chaplot et al., 2020]).

In contrast to the previously studied disembodied model, which might abstain from responding in situations with insufficient observations, our agent developed in this chapter has the capability to actively collect necessary visual information through active vision and respond accordingly. While we demonstrated that it is feasible to develop action-response agents that can handle both active vision control and task-relevant responses, we acknowledge the complexity of the training strategy. Specifically, we resort to a highly artificial curriculum learning strategy to facilitate the training process of the agent model proposed in this chapter. In the next chapter, we will delve deeper into the training challenge of such agents, and attempt to find methods to ease the training process.

Chapter 6

Stabilizing Reinforcement Learning for Active Vision

6.1 Introduction

In the previous chapter, we demonstrated that the active vision control policy can be developed through reinforcement learning, utilizing predictions from the response module as the reward. However, training the model proved to be challenging. The successful training of the model in Chapter 5 relied on a human-designed curriculum training strategy, which restricts the model's application in broader contexts.

We recognize that such challenges of unstable training are present in a series of models across various problems. In these problems, the RL policy is trained using reward signals that come from the discrimination performance of a discriminator module that is simultaneously trained with the RL policy, and in turn, the discriminator is trained depending on the information collected by the policy. In this chapter, we attempt to consider this training challenge from a more fundamental perspective, not only limited to the application area of active vision control policy learning but also other application areas, as long as they share the same training paradigm, such as in the realm of unsupervised skill discovery. In this class of reinforcement learning problems, reward signals for policy learning are generated by an internal reward model that is dependent on and jointly optimized with the policy. This interdependence between the policy and the reward model leads to an unstable learning process because reward signals from an immature reward model are noisy and impede policy learning, and conversely, an under-optimized policy impedes reward estimation learning.

This learning setting is referred to as *Internally Rewarded Reinforcement Learning* (IRRL) as the reward is not provided directly by the environment but *internally* by a reward model. We formally formulate IRRL and present a class of problems that belong to IRRL. We theoretically derive and empirically analyze the effect of the reward function in IRRL and based on these analyses propose the clipped linear reward function. Experimental results show that the proposed reward function can consistently stabilize the training process by reducing the impact of reward noise, which leads to faster convergence and higher performance compared with baselines in diverse tasks.

6.2 Reinforcement Learning with Reward Models

Rewards are essential for animals and artificial agents to learn by exploration in an environment. In the brain, reward signals are emitted by specific neurons as a consequence of the processing of external stimuli [Olds and Milner, 1954, Schultz, 2015]. For instance, when a child receives words of praise from the parents as feedback for exhibiting appropriate behavior, the rewards obtained are contingent upon the child's individual understanding of the words. In some cases, the child may misunderstand the praise as criticism, thus wrongly obtaining a negative reward and impeding its behavior learning. An elaborated view of the standard agent-environment interaction formulation [Sutton and Barto, 1998] of reinforcement learning (RL) demonstrates this mechanism [Singh et al., 2004]. This framework separates the environment into an *external environment*, which provides external stimuli (e.g., a word of praise from the parents), and an *internal environment*, which is in the same "organism" with the agent and contains a reward model[1] that processes external stimuli and produces reward signals (cf. Fig 6.1 left panel).

In the study of this chapter, we focus on situations where the reward is determined by both external stimuli and the state of a sophisticated and evolutionary internal environment that produces either task-relevant rewards [Mnih et al., 2014, Ba et al., 2015, Li et al., 2021, Rangrej et al., 2022] or task-agnostic rewards [Gregor et al., 2017, Strouse et al., 2022], and we use the term *Internally Rewarded Reinforcement Learning* (IRRL) to refer to the learning problem in these situations (cf. some IRRL examples in Fig. 6.3).

In IRRL, the policy of the agent is trained by RL, and the reward model of the internal environment is simultaneously trained either in self-supervised learning (SSL) manner by directly using the sensations from the external environment [Pathak et al., 2017, Gregor et al., 2017, Eysenbach et al., 2019, Strouse et al., 2022], or in a supervised learning (SL) manner by using extra human-annotated task-relevant signals [Mnih et al., 2014, Yu et al., 2017, Tan et al., 2020, Li et al., 2021, Christiano et al., 2017, Ouyang et al., 2022]. The reward model provides reward signals for training a policy that, in return, controls the collection of the trajectories for the reward model. These scenarios have become prevalent with increased interest in integrating the capability of high-level prediction and low-level control of behaviors into a single model in the realms of attention mechanisms [Mnih et al., 2014, Ba et al., 2015, Yu et al., 2017, Li et al., 2017, Rangrej et al., 2022], embodied agents [Gordon et al., 2018, Yang et al., 2019a],

[1] We adopt the term "reward model" instead of using "critic", as used by Singh et al. [2004], to prevent confusion with the term "critic" in actor-critic algorithms [Konda and Tsitsiklis, 1999].

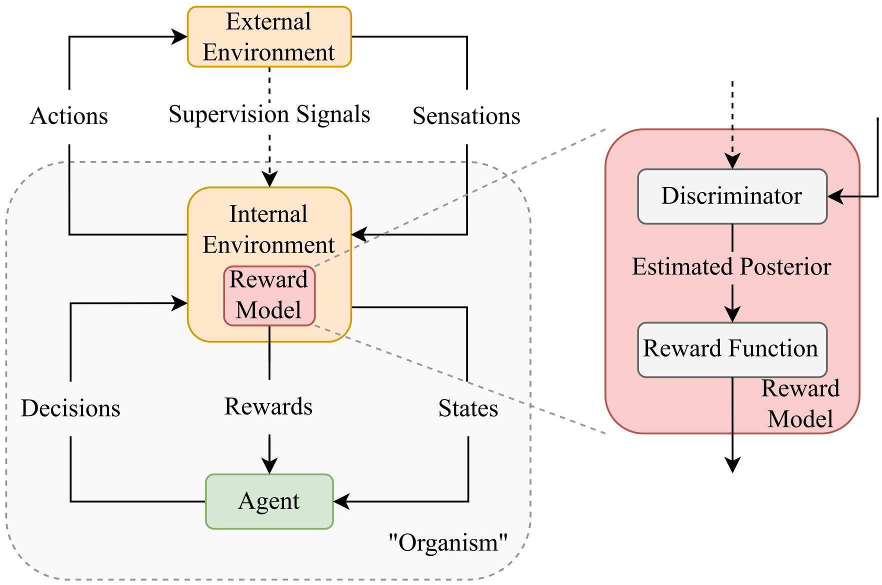


Figure 6.1: **Left:** The agent-environment interaction loop of IRRL. This diagram is based on the scheme of intrinsically motivated RL [Singh et al., 2004] with an optional path of supervision signals, which reflects an extrinsic reward. **Right:** The internal reward model consists of a *discriminator*, which estimates a posterior probability of correct discrimination given sensations and supervision signals from the external environment, and a *reward function*, which produces rewards by processing the posterior probability.

robotics [Lakomkin et al., 2018, Li et al., 2021], unsupervised RL [Gregor et al., 2017, Eysenbach et al., 2019, Strouse et al., 2022], and reinforcement learning from human feedback (RLHF) [Christiano et al., 2017, Ouyang et al., 2022].

The role of the reward model depends on the target task. In the task of digit recognition with hard attention (see Fig. 6.3a), for example, the reward model assesses the certainty of performing correct digit classification. In the unsupervised skill discovery task (see Fig. 6.3b), however, the reward model works as an intrinsic motivation system to evaluate the novelty of generated skills. The reward model consists of a discriminator and a reward function, as shown in the right panel of Fig. 6.1. The discriminator estimates the posterior probability of the target label provided by supervision signals or sensations. Given the posterior, the reward function produces rewards for the behavior learning of the agent.

Simultaneous optimization between the policy and the discriminator in IRRL is however non-trivial because of the unstable training loop where neither of them can learn efficiently (see Fig. 6.2). In this work, we seek to solve this issue by reducing the impact of reward noise, which is challenging due to the unavailability of an oracle discriminator whose posterior probability can reflect the information sufficiency for discrimination. We theoretically formulate IRRL to explicitly analyze the noisy reward issue and characterize the distribution of the noise empirically by approximating the oracle discriminator with the discriminator of a converged

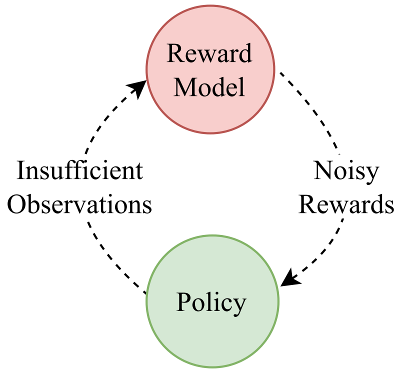


Figure 6.2: Simultaneous optimization between the policy of the agent and the reward model is challenging because an under-optimized reward model yields noisy rewards, and in turn, an immature policy yields insufficient observations, which leads to an unstable training loop.

model. Based on our formulation and empirical results, we demonstrate the effect of the reward function in reducing the bias of the estimated reward and the variance of the reward noise and propose a simple yet effective reward function that stabilizes the training process.

We present extensive experimental results on IRRL tasks with task-relevant rewards (i.e., visual hard attention, and robotic active vision), or tasks with task-agnostic rewards (i.e., unsupervised skill discovery). The results suggest that our proposed reward function consistently improves the stability and the speed of training, and achieves better performance than the baselines on all the tasks. In particular, on the skill discovery task, our approach with the simple reward function achieves the same performance as the state-of-the-art sophisticated ensemble-based Bayesian method by Strouse et al. [2022] but without using ensembles. We further demonstrate that the superiority of the proposed reward function is due to its effectiveness in noise reduction, which is in line with our theoretical analysis.[2] The contributions of the study of this chapter are summarized as follows:

1. We formulate a class of RL problems as IRRL, and formulate the inherent issue of *noisy rewards* that leads to an unstable training loop in IRRL.

2. We empirically characterize the noise in the discriminator and derive the effect of the reward function in reducing the bias of the estimated reward and the variance of the reward noise stemming from an underdeveloped discriminator.

3. We propose a simple yet effective reward function, the *clipped linear reward function*, which consistently stabilizes the training process and achieves faster convergence speed and higher performance on diverse IRRL tasks.

[2] Code for reproducing the experimental results reported in this chapter is open-sourced at https://github.com/mengdi-li/internally-rewarded-rl

6.3 Related Work

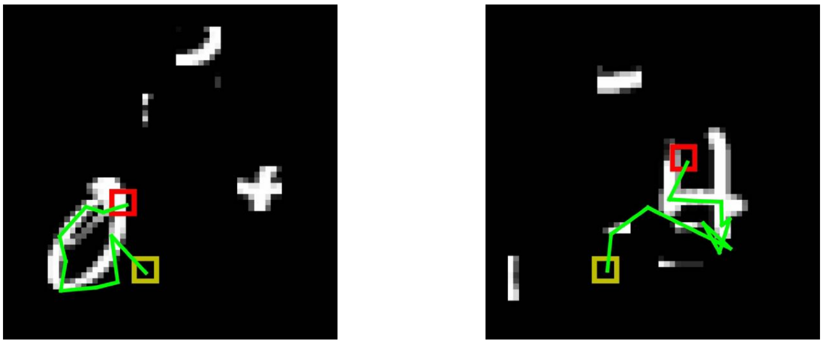
The RL process is notoriously unstable. Previous work has studied various techniques to stabilize training, such as reducing the bias and variance of gradient estimation for policy gradient methods [Greensmith et al., 2004, Schulman et al., 2015b], and value estimation for value-based methods [van Hasselt et al., 2016]. As another factor impacting RL training, reward noise that stems from various sources, e.g., sensors on robots, and adversarial attacks, is attracting attention because of the growing interest in applying RL to more realistic and complicated tasks [Huang et al., 2017, Everitt et al., 2017, Wang et al., 2020]. In cases where the noise directly resides in the reward, both policy gradient and value-based RL methods suffer. Everitt et al. [2017] and Wang et al. [2020] formulate RL with corrupted rewards and partially address the issue for cases with extra knowledge about the noise. Unlike the noise caused by reward corruption, the noise in IRRL comes from a discriminator and is subject to the learning process, so their approaches are not directly applicable to our scenarios in terms of both formulation and experimental emulation.

The issues of unstable training in IRRL have been mentioned in the literature, but they have not been systematically studied. Some works [Mnih et al., 2014, Ba et al., 2015, Li et al., 2017] ignore the impact of the unstable training loop at the expense of the training speed and the performance of the final model. Other works resort to elaborated training strategies, e.g., staged training [Gordon et al., 2018, Yang et al., 2019a, Lysa et al., 2022], curriculum training [Das et al., 2018, Li et al., 2021], imitation learning [Tan et al., 2020, Rangrej et al., 2022], or task-specific reward shaping [Deng et al., 2021]. However, extra efforts such as data collection or human ingenuity are needed in these methods.

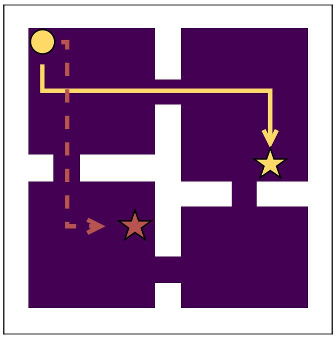
Strouse et al. [2022] study the pessimistic exploration problem in the context of unsupervised skill discovery (cf. Fig. 6.3b) where a skill discriminator is used to generate rewards. As the skill discriminator is subject to noise, this issue can be seen as a consequence of the unstable training loop under the framework of IRRL. Similar to our work, they also resort to modifying the reward function. They propose to train an ensemble of discriminators and reward the policy with their disagreement. Experimental results suggest that the proposed disagreement-based reward lets the agent learn more skills through optimistic exploration. However, this method introduces more model parameters and hyper-parameters than baseline methods that are not based on ensembles. In this chapter, we consider the issue in a more general context including but not limited to unsupervised skill discovery, and manage the issue in a more simple and efficient way.

6.4 Problem Formulation

We formulate the policy learning of IRRL as a Markov decision process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p_{\mathrm{E}}, \rho, r, \gamma \rangle$, where, $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $p_{\mathrm{E}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ the state transition probability, $\rho : \mathcal{S} \to \mathbb{R}$ the distribution of the initial state,



(a) **Hard attention for digit recognition on the Cluttered MNIST dataset** [Mnih et al., 2014]. A small glimpse (the squares) controlled by an attention policy sequentially changes its location to collect information for recognizing the digit. During training, the reward model is expected to produce rewards that reflect the sufficiency of information collected by the attention policy, and in turn, the policy is expected to attend to informative regions, i.e., pixels of the digit, to collect information for the classifier to learn digit recognition. The starting and stopping glimpses are represented by yellow and red boxes respectively. The green line indicates the positions of intermediate glimpses.

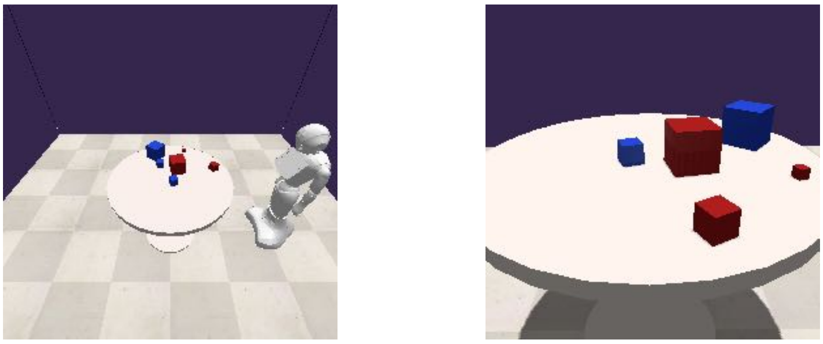


(b) **Unsupervised skill discovery in a four-room environment** [Strouse et al., 2022]. An agent spawned at the top-left corner is expected to learn a navigation policy that performs distinguishable skills without using any extrinsic rewards. In this task, a skill is represented by the final state of a trajectory. During training, the agent generates a trajectory conditioned on a randomly sampled skill label, and a discriminator estimates the posterior probability of the trajectory being the target skill, based on which the reward is produced. The policy and the discriminator are optimized simultaneously.

Figure 6.3: Example tasks of IRRL

$r : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ the reward on each transition, and $\gamma \in (0, 1)$ a discount factor.

Different from conventional RL settings, where reward r depends exclusively on the *external* environment, in IRRL reward r is determined by a *reward model*, which resides in the *internal* environments and interprets the supervision signals from the external environment to generate internal rewards (cf. Fig. 6.1). Here, we assume that the external environment, hence the observations an agent is making, is caused by a label y sampled from a prior distribution $p(y)$. The reward model depends on a trainable *discriminator* q_ϕ parameterized with ϕ. Given a trajectory $\tau \in (\mathcal{S} \times \mathcal{A})^n$ ($n \in \mathbb{N}$ is the trajectory length) sampled from a policy π_θ parameterized with θ,



(c) **Robotic object counting in occlusion scenarios**. A humanoid robot is trained to learn a locomotion policy to explore occluded space by rotating around the table and to terminate exploration to achieve efficient counting of specified objects, e.g., *small_blue_cube*. The robot performs the task solely based on its egocentric RGB view. During training, the policy uses the reward that is produced by a reward model containing an object counter, which is simultaneously updated with the policy. Similar to the task of hard attention, the reward should be able to evaluate the information sufficiency of observations for correct object counting.

Figure 6.3: Example tasks of IRRL (cont.)

the discriminator $q_\phi(y \mid \tau)$ computes the probability of label y being the cause of trajectory τ.[3]

Many existing works, which have been studied independently before, can be categorized as instances of IRRL. In the subsequent discussion, we present three lines of existing works as concrete examples of IRRL:

1. **Hard attention**. Hard attention mechanism [Mnih et al., 2014, Ba et al., 2015, Li et al., 2017, Rangrej et al., 2022] is essential when all available information is expensive or unrealistic to process, e.g., scene classification for high-resolution satellite images [Wang et al., 2019, Rangrej et al., 2022]. Fig. 6.3a shows the task of hard attention for digit recognition on the Cluttered MNIST dataset [Mnih et al., 2014].

2. **Intrinsically motivated RL**. In this setting, an agent is trained using dense intrinsic rewards to explore the environment based on its curiosity about encountered states [Pathak et al., 2017] or to discover diverse skills based on their novelty [Gregor et al., 2017, Eysenbach et al., 2019, Strouse et al., 2022]. Fig. 6.3b shows the task of unsupervised skill discovery in a four-room environment [Strouse et al., 2022].

3. **Task-oriented active vision**. This is an emerging research topic with the goal of endowing embodied agents with high-level perception and reasoning capabilities. The agent actively changes its egocentric view to collect information for achieving downstream tasks, e.g., question answering [Gordon et al., 2018, Deng et al., 2021, Li et al., 2021], object recognition [Yang et al., 2019a], or scene

[3]To simplify notations, we use lower-case letters (e.g., y) to both represent random variables and their realizations if the distinction is clear from the context. Similarly, we use $p(y)$ to represent both the distribution of y and the probability of y if the context is clear.

description [Tan et al., 2020]. Fig. 6.3c shows the task of robotic object counting in occlusion scenarios.

6.4.1 Model Optimization

In IRRL, the policy and the discriminator are optimized simultaneously with different optimization objectives.

Policy Optimization

The optimization objective of policy learning in IRRL can be formulated from two perspectives, which are accuracy maximization and mutual information maximization.

Accuracy Maximization This is an intuitive formulation, where the policy of the agent is optimized to maximize the expectation of an accuracy-based reward

$$r_{\text{acc}} = \mathbb{1}_y \left[\operatorname*{argmax}_{y' \in \mathcal{Y}} q_\phi(y' \mid \tau) \right], \tag{6.1}$$

where $\mathcal{Y}$ is a set of possible labels and $\mathbb{1}_y[x]$ is an indicator function that returns 1 if x is the target label y, 0 otherwise. This formulation has been widely used in existing works on hard attention [Mnih et al., 2014, Kingma and Ba, 2015, Li et al., 2017], embodied agents [Gordon et al., 2018, Yang et al., 2019a], and robotics [Lakomkin et al., 2018, Li et al., 2021]. However, an obvious disadvantage of the accuracy-based reward is that it cannot faithfully reflect the discriminator's uncertainty about the observations collected by the reinforcement learner, which makes learning slow and leads to suboptimal performance (cf. Sec. 6.6). Therefore, it will be analyzed only empirically in this work.

Mutual Information Maximization Mutual information is commonly used to estimate the relationship between pairs of random variables. The objective of mutual information maximization has been utilized in the realm of unsupervised skill discovery [Gregor et al., 2017, Eysenbach et al., 2019, Strouse et al., 2022]. We generalize it to the optimization objective of IRRL.

Given a target label y and a trajectory τ sampled from $p(y)$ and π_θ respectively, their mutual dependency can be obtained by the KL-divergence of their joint distribution $p(y, \tau)$ and the product of their marginal distributions $p(y)p(\tau)$:

$$\begin{aligned} I(y; \tau) &:= D_{\mathbb{KL}}(p(y, \tau) \parallel p(y)p(\tau)) \\ &= \mathbb{E}_{\tau \sim \pi_\theta, y \sim p(y)} \left[\log p(y \mid \tau) - \log p(y) \right], \end{aligned} \tag{6.2}$$

which is also known as Shannon's mutual information between y and τ and which reaches its maximum if the full knowledge of y can be deduced from τ. In this equation, $p(y \mid \tau)$ is the oracle posterior probability that reflects the information sufficiency of observations for discrimination. It can be interpreted as being generated by an *oracle discriminator*, a conceptual term utilized for the theoretical formulation. If $p(y \mid \tau)$ is known, then by defining

$$r^*_{\log} := \log p(y \mid \tau) - \log p(y) \tag{6.3}$$

as the reward for an RL algorithm involving π_θ, one can maximize $I(y;\tau)$, i.e., π_θ generates trajectories for an optimal discrimination of the target label y.

Because the oracle discriminator $p(y \mid \tau)$ is not available in practice, we can replace $p(y \mid \tau)$ with a neural network $q_\phi(y \mid \tau)$ with trainable parameters ϕ and define the reward as

$$r_{\log} = \log q_\phi(y \mid \tau) - \log p(y), \tag{6.4}$$

and maximize the Barber-Agakov variational lower bound of $I(y;\tau)$ [Barber and Agakov, 2003]:

$$I_{\mathrm{BA}}(y;\tau) := \mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)}[\log q_\phi(y \mid \tau) - \log p(y)].$$

Discriminator Optimization

Concurrent with policy learning, the discriminator $q_\phi(y \mid \tau)$ is trained to better approximate $p(y \mid \tau)$. To this end, instead of the cross-entropy loss

$$-\mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)}\left[p(y \mid \tau)\log q_\phi(y \mid \tau)\right],$$

which involves the oracle discriminator $p(y \mid \tau)$, a proxy cross-entropy loss $-\mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)} \log q_\phi(y \mid \tau)$ is used in practice, which is equivalent to assuming $p(y \,|\, \tau) = 1$, i.e., assuming that τ contains sufficient information for deducing y with the oracle discriminator.

6.4.2 The Issue of Reward Noise

As the trainable discriminator $q_\phi(y \mid \tau)$ only approximates the oracle discriminator $p(y \mid \tau)$, it inevitably introduces noise $\varepsilon_{\log}$ in the reward $r_{\log}$ in Eq. (6.4), which is given by

$$\varepsilon_{\log} = r_{\log} - r^\star_{\log} = \log q_\phi(y \mid \tau) - \log p(y \mid \tau).$$

To demonstrate the negative impact of reward noise on the learning process (cf. Fig. 6.2), we conduct *reward hacking* experiments, where we replace the trainable discriminator $q_\phi(y \mid \tau)$ with a pretrained one $q_{\tilde{\phi}}(y \mid \tau)$ that is obtained from a converged model to mimic the oracle discriminator $p(y \mid \tau)$. The setup of the reward hacking experiment is illustrated in Fig. 6.4. We choose the digit recognition task as the target task (cf. Fig. 6.3a) and use the recurrent attention model (RAM) [Mnih et al., 2014] (detailed information about this task and the model is given in Sec. 6.6.1 and Sec. 6.6.3).

Fig. 6.5 shows a plot of training curves when using different reward functions with and without reward hacking. As shown by the gap between the training curves when using an identical reward function with and without reward hacking, the noise of an under-optimized discriminator influences the training process negatively. In this work, we aim to narrow the gap by devising an effective reward. As we will see in the next section, a well-designed reward is key to stabilizing the learning process.

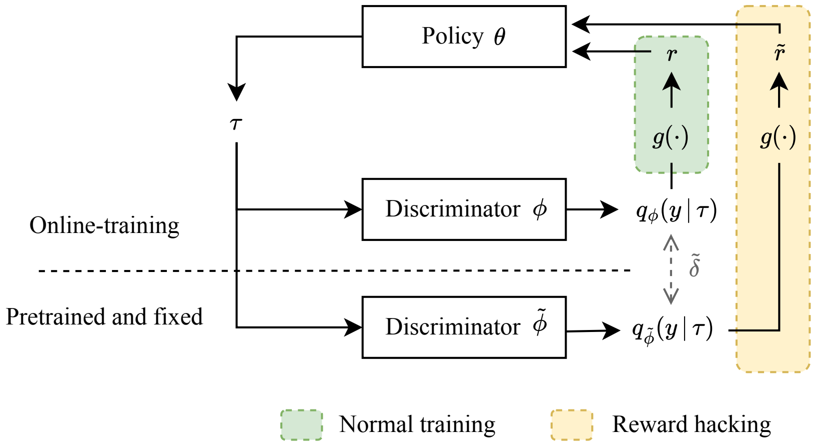


Figure 6.4: Illustration of the experimental setup of reward hacking. In normal training, the reward is produced based on the posterior probability estimated by an online-training discriminator ϕ. In training with reward hacking, the reward stems from a pretrained and fixed discriminator $\tilde{\phi}$. $\tilde{\delta}$ indicates the difference between a pair of posterior probabilities estimated by ϕ and $\tilde{\phi}$.

6.5 Reward Noise Moderation

In this section, we first analyze the reduction of the bias of the estimated reward and the variance of the reward noise and then propose a reward that alleviates the negative effect of reward noise and stabilizes the training process.

6.5.1 Generalized Reward

Since the noisy reward in Eq. (6.4) is a transformation of the posterior probability $q_\phi(y \mid \tau)$, it is reasonable to study the effect of a series of transformations of $q_\phi(y \mid \tau)$ as long as they agree on the same optimal objective. Based on the logarithmic transformation in Eq. (6.4), the *generalized reward* is defined as

$$r_g = g\left[q_\phi(y \mid \tau)\right] - g\left[p(y)\right] \tag{6.5}$$

and the *generalized oracle reward* as

$$r_g^* = g\left[p(y \mid \tau)\right] - g\left[p(y)\right],$$

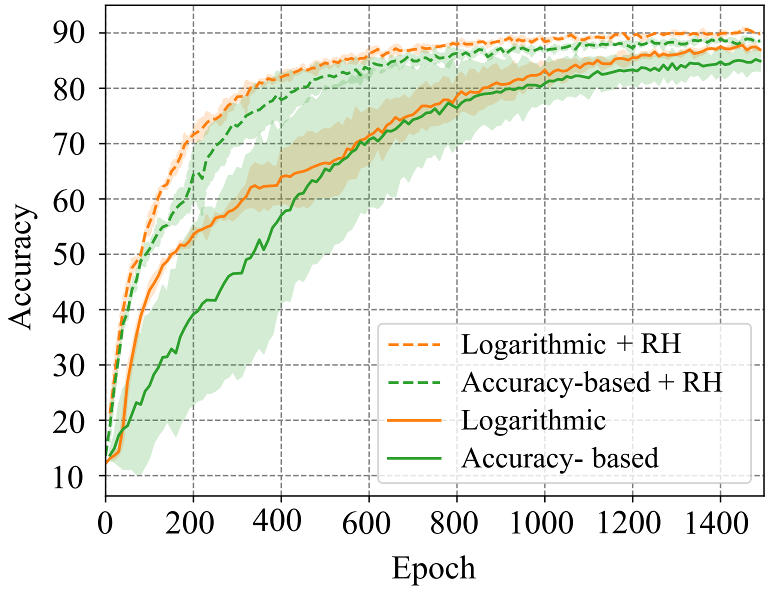


Figure 6.5: RAM trained using the accuracy-based and the logarithmic reward with and without reward hacking (RH). A model without reward hacking is subject to more noisy rewards and suffers from an unstable learning process, resulting in slower convergence and lower accuracy.

where g is an increasing function (e.g., log), such that maximizing $g(\cdot)$ leads to the maximization of the mutual information $I(y;\tau)$. When selecting the appropriate function, it is important to consider both its ability to transmit information and its ability to moderate noise. The former ensures that the maximization of mutual information can be achieved efficiently, while the latter helps to reduce the impact of reward noise.[4]

6.5.2 Generalized Reward Noise

To analyze the noise in the generalized reward r_g we apply the second-order Taylor approximation to the *generalized reward noise*

$$\varepsilon_g := r_g - r_g^* = g\left[q_\phi(y \mid \tau)\right] - g\left[p(y \mid \tau)\right]$$

at point $p(y \mid \tau)$. By defining $\delta := q_\phi(y \mid \tau) - p(y \mid \tau)$ as the *discriminator noise*, we have as the expectation of the reward noise (equivalently, the bias of the reward estimator)

$$\mathbb{E}[\varepsilon_g] \approx g'(p(y \mid \tau))\mathbb{E}[\delta] + \frac{1}{2!}g''(p(y \mid \tau))\mathbb{E}\left[\delta^2\right], \tag{6.6}$$

[4]The transformation $g(\cdot)$ can also be motivated by the f-mutual information objectives (see Sec. 6.7.1).

and as the variance of the reward noise

$$\mathbb{V}[\varepsilon_g] \approx (g'(p(y \mid \tau)))^2 \mathbb{V}[\delta] + (\frac{1}{2!} g''(p(y \mid \tau)))^2 \mathbb{V}\left[\delta^2\right] + g'(p(y \mid \tau)) g''(p(y \mid \tau)) \mathrm{Cov}\left[\delta, \delta^2\right]. \tag{6.7}$$

Our goal is to mitigate the impact of the reward noise by minimizing the expectation and the variance of the noise. This is expected to be achieved especially at the early learning stage when the issue of the unstable training loop is severe because both the discriminator and the policy are immature: the trajectory collected by the policy contains little information for discrimination, and the estimated posterior of the discriminator cannot reflect the sufficiency of information collected by the policy. To this end, in the following sub-sections, we theoretically and empirically analyze Eq. (6.6) and Eq. (6.7) and investigate reward functions.

6.5.3 Characterization of the Discriminator Noise

We make hypotheses regarding the distribution characteristics of the discriminator noise δ, which is necessary to analyze the expectation and variance of the reward noise ε_g according to Eq. (6.6) and Eq. (6.7). We hypothesize that the expectation of δ is zero, i.e., $\mathbb{E}[\delta] = 0$, and the distribution of δ is symmetric.

We conduct an empirical study of the distribution of the discriminator noise following the setup of the reward hacking experiment (cf. Fig. 6.4). Instead of using a pretrained discriminator to interfere in the training process, we visualize the approximated discriminator noise $\tilde{\delta}$ during normal training. $\tilde{\delta}$ is the difference between the posterior probabilities estimated by the online-training discriminator and the pretrained discriminator, i.e., $\tilde{\delta} = q_\phi(y \mid \tau) - q_{\tilde{\phi}}(y \mid \tau) \approx \delta$.

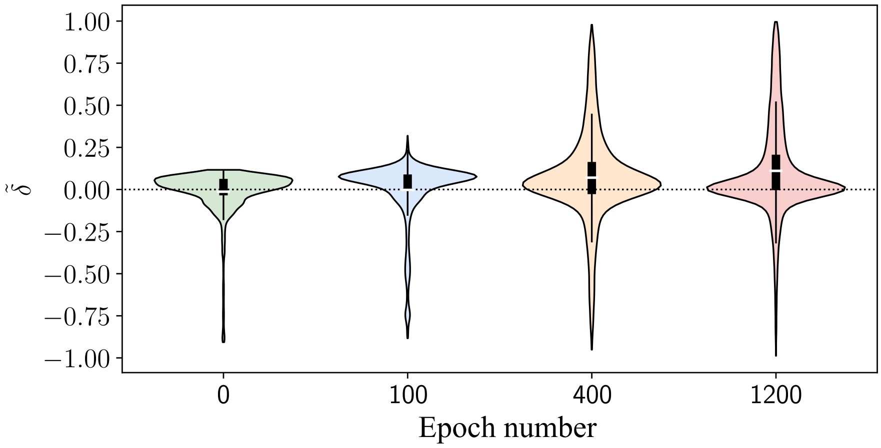
Fig. 6.7 demonstrates violin plots of the discriminator noise at four training epochs (the model converges at about 1200 epochs). Each violin plot is drawn from 1000 random samples from the testing dataset. We can observe that the mean of the noise is close to zero at different training stages, i.e., $\mathbb{E}[\delta] \approx 0$, and the plots are almost symmetrical with respect to the average noise except at the very beginning when the model weights are being updated after random initialization. We assume that the noise characteristics are generalized to other problems of IRRL because of the shared high-level abstraction among them (cf. Fig. 6.1).

Besides the visualization of the discriminator noise when the policy is trained using the logarithmic reward (see Fig. 6.7), we also visualize the discriminator noise when using the accuracy-based and the clipped linear reward in Fig. 6.6. We can see that the discriminator noise when using the clipped linear reward has a smaller bias and variance.

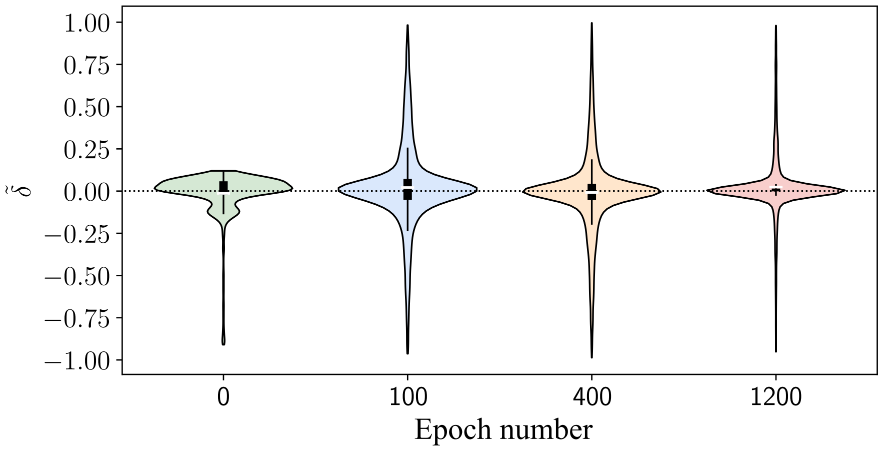
6.5.4 Linear Reward

Considering the impact of $g(\cdot)$ on the Taylor approximation to $\mathbb{E}[\varepsilon_g]$ and $\mathbb{V}[\varepsilon_g]$ in Eq. (6.6) and Eq. (6.7), we propose a linear reward

$$r_{\text{lin}} = q_\phi(y \mid \tau) - p(y), \tag{6.8}$$



(a) Accuracy-based reward



(b) Clipped linear reward

Figure 6.6: Visualization of the discriminator noise of RAM trained using the accuracy-based (cf. Eq. (6.1)) and the clipped linear reward (cf. Eq. (6.9)).

instead of the commonly applied logarithmic reward $r_{\log}$, to stabilize IRRL. The corresponding expectation and variance of the noise are $\mathbb{E}[\varepsilon_{\text{lin}}] = \mathbb{E}[\delta] = 0$ and $\mathbb{V}[\varepsilon_{\text{lin}}] = \mathbb{V}[\delta]$, respectively. The linear reward enjoys lower reward bias than the logarithmic reward since

$$|\mathbb{E}[\varepsilon_{\log}]| \approx \frac{1}{2!\, p^2(y \mid \tau)} \mathbb{E}[\delta^2] > 0 = |\mathbb{E}[\varepsilon_{\text{lin}}]|.$$

Furthermore, the variance of r_{lin} is low and stable compared with the variance of logarithmic reward $r_{\log}$, which suffers from high variance

$$\mathbb{V}[\varepsilon_{\log}] \approx p^{-2}(y \mid \tau)\mathbb{V}[\delta] + (\frac{1}{2!\, p^2(y \mid \tau)})^2 \mathbb{V}[\delta^2],$$

since $p(y \mid \tau) < 1$ in most cases and is dependent on the training policy. A detailed derivation is given in Sec. 6.5.5. The empirical evaluation of various g functions is given in Sec. 6.6.6.

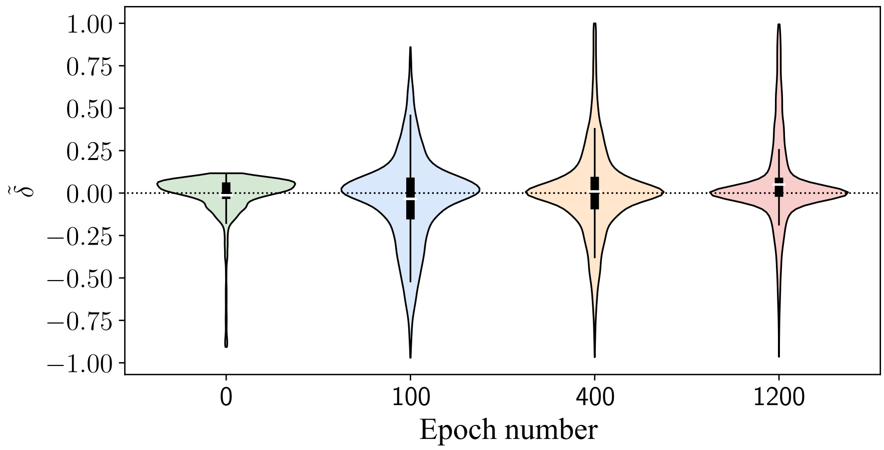


Figure 6.7: Violin plots of the approximated discriminator noise $\tilde{\delta}$ in the training process of RAM, of which the policy is trained using the logarithmic reward function (cf. Eq. (6.4)). The small white bar indicates the mean of the noise. The thick vertical line represents the interquartile range and the thin vertical line represents the area between the upper and lower adjacent values.

6.5.5 Noise of the Logarithmic Reward

Based on the formulation of Eq. (6.6) and Eq. (6.7), the expectation and variance of the reward noise when using the logarithmic reward ($\varepsilon_{\log}$) can be derived as follows:

$$
\begin{aligned}
\mathbb{E}[\varepsilon_{\log}] &= g'(p(y \mid \tau))\mathbb{E}[\delta] + \frac{1}{2!}g''(p(y \mid \tau))\mathbb{E}[\delta^2] + \mathbb{E}[o(\delta^2)] \\
&= \frac{1}{p(y \mid \tau)}\mathbb{E}[\delta] - \frac{1}{2!\, p^2(y \mid \tau)}\mathbb{E}[\delta^2] + \mathbb{E}[o(\delta^2)] \\
&\approx -\frac{1}{2!\, p^2(y \mid \tau)}\mathbb{E}[\delta^2],
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{V}[\varepsilon_{\log}] &= (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}[\delta^2] + g'(p(y \mid \tau))g''(p(y \mid \tau))\mathrm{Cov}[\delta, \delta^2] \\
&\quad + \mathbb{V}[o(\delta^2)] + 2g'(p(y \mid \tau))\mathrm{Cov}[\delta, o(\delta^2)] + g''(p(y \mid \tau))\mathrm{Cov}[\delta^2, o(\delta^2)] \\
&\approx (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}[\delta^2] + g'(p(y \mid \tau))g''(p(y \mid \tau))\mathrm{Cov}[\delta, \delta^2] \\
&\approx (g'(p(y \mid \tau)))^2\mathbb{V}[\delta] + (\frac{1}{2!}g''(p(y \mid \tau)))^2\mathbb{V}[\delta^2] \\
&= \frac{1}{p^2(y \mid \tau)}\mathbb{V}[\delta] + (\frac{1}{2!\, p^2(y \mid \tau)})^2\mathbb{V}[\delta^2].
\end{aligned}
$$

The variance is approximated using the fact that $\mathrm{Cov}[\delta, \delta^2] = \mathbb{E}[\delta^3] - \mathbb{E}[\delta]\mathbb{E}][\delta^2] = (\mu^3 + 3\mu\sigma^2 + \gamma\sigma^3) - \mu(\mu^2 + \sigma^2) = 2\mu\sigma^2 + \gamma\sigma^3 \approx 0$, where $\sigma^2 = \mathbb{E}[(\delta - \mu)^2]$ is the variance, and $\mu = \mathbb{E}[\delta]$ and $\gamma = \mathbb{E}[(\frac{\delta-\mu}{\sigma})^3]$ are the mean and skewness, which are both about zero due to the symmetry of the distribution of δ (see Sec. 6.5.3).

6.5.6 Clipped Linear Reward

The issue of reward noise is not fully tackled by using the linear reward. Given a target label y, it is intuitive to assume that the posterior probability $p(y \mid \tau)$ of an oracle discriminator should be, in most cases, equal or larger than the prior $p(y)$, as y is a cause of the trajectory τ. However, a discriminator q_ϕ may return a posterior probability $q_\phi(y \mid \tau)$ lower than $p(y)$, especially at the early training stage when both the policy and the discriminator are under-optimized.

Since we expect $q_\phi(y \mid \tau)$ to be close to $p(y \mid \tau)$, we replace the term $q_\phi(y \mid \tau)$ of r_{lin} in Eq. (6.8) with $\max(q_\phi(y \mid \tau), p(y))$ to integrate the prior knowledge and define the *clipped linear reward* as

$$\begin{aligned} \overline{r_{\text{lin}}} &:= \max(q_\phi(y \mid \tau), p(y)) - p(y) \\ &= \max(q_\phi(y \mid \tau) - p(y), 0). \end{aligned} \tag{6.9}$$

Similar clipping techniques are empirically found to be beneficial when applied to the logarithmic reward [Strouse et al., 2022]. In this work, we go further with an analysis of reward functions from the perspective of noise moderation and achieve better performance with the proposed reward. The proposed clipped linear reward has a similar shape to the rectified linear unit (ReLU) activation function [Nair and Hinton, 2010] which preserves information about relative intensities in multiple layers of deep neural networks. Likewise, the clipped linear reward function can robustly preserve information that travels from an internal discriminator to the policy network.

6.6 Experiments

In this section, we introduce experiments conducted to evaluate the effectiveness of the proposed method on the three aforementioned tasks in Sec. 6.4. We first introduce experimental setups and baselines. Then, we compare the proposed clipped linear function with multiple baselines including state-of-the-art methods. Finally, we conduct reward hacking experiments on the clipped linear reward function to visualize its capability to reduce the impact of reward noise.

6.6.1 Experimental Setup

Hard Attention for Digit Recognition We adopt the dataset configuration of Mnih et al. [2014], and use two basic models for this task: the recurrent attention model (RAM) [Mnih et al., 2014] and the dynamic-time recurrent attention model (DT-RAM) [Li et al., 2017]. RAM performs a fixed number of movement steps before performing the final digit recognition, while the policy of DT-RAM learns to terminate the exploration before reaching a maximum number of movement steps. The performance of the agent is evaluated using the accuracy of the digit recognition.

Unsupervised Skill Discovery We use the same experimental setup and basic model on the four-room environment as in the work of the discriminator disagreement intrinsic reward (DISDAIN) [Strouse et al., 2022]. The performance of the agent is evaluated using the number of learned skills.

Robotic Object Counting The setup is based on the task of object existence prediction [Li et al., 2021]. We use their model and train it using PPO [Schulman et al., 2017] instead of REINFORCE for higher efficiency. The performance of the agent is evaluated using the accuracy of object counting.

Details of the environments and model implementations will be introduced below.

6.6.2 Environments

Cluttered-MNIST We generate the Cluttered MNIST dataset by a generator provided by the code repository[5] of Sønderby et al. [2015], where we adopt the dataset configuration from Mnih et al. [2014]. A Cluttered MNIST image is generated by randomly placing an original MNIST image (28×28) and 4 randomly cropped patches (8×8) from original MNIST images in an empty image (60×60). We generate 60k Cluttered MNIST images, of which 90% are used for training and the rest for validation.

Four-room Environment The four-room environment is adopted from Strouse et al. [2022] and is shown in Fig. 6.3b. There are four rooms and 104 states. The agent is initialized at the top-left corner at each episode and can select an action from {*left*, *right*, *up*, *down*, *no-op*} at each time step. The length of each trajectory is 20, by which the agent is able to reach all but one state, raising the maximum number of possible learned skills 103. The target skill label is uniformly sampled as an integer in $[0, 127]$ at each episode.

Object Counting We create a simulation environment for the task of object counting in occlusion based on the simulation environment provided by the code repository[8] of Li et al. [2021]. We use cubes of three different sizes (*small*, *medium*, and *large*) in two different colors (*red*, and *blue*) as objects on the table. The goal object is one of the *small* or *medium* objects. Each scene is initialized under the following constraints: 1) at least one large object is on the table as an abstraction; 2) the number of other objects N is sampled from a Poisson distribution ($\lambda = 4$) and is clipped at a maximum number of 6; 3) one of the goal objects is occluded by an object of a larger size with a probability of 80% to make occlusion happen frequently. The number of goal objects is uniformly sampled between 0 and N.

[5] https://github.com/skaae/recurrent-spatial-transformer-code

The agent is initialized in front of the table and takes as input an egocentric RGB image with a resolution of 256×256 (cf. Fig. 6.3c). The agent has three discrete actions: *rotate_right*, *rotate_left*, and *stop*. The agent circles around the table by 30 degrees with each rotation action. The maximum number of movement steps is 6, by which the agent can move to the opposite of its initial position. We generate offline datasets for training (100k scenes) and evaluation (1k scenes) because online occlusion checking including scene initialization in the CoppeliaSim simulator is slow.

6.6.3 Implementation

RAM We use an existing implementation of the original RAM model[6]. Given an image and the coordinate of the glimpse, a glimpse network extracts visual representations of the attended patch by an MLP. The coordinate is mapped into representations by another MLP. The two representation vectors have the same dimensionality of 256. They are added together to get glimpse representations. A simple RNN as the core network recurrently processes glimpse representations and produces hidden representations with a dimensionality of 256 at each time step. A policy network takes hidden representations of the core network as input to predict the location of the next glimpse. When the maximum number of movement steps is reached, a classification network takes hidden representations of the core network as input to produce the class prediction and finalize the task. The maximum number of movement steps is 18 in our experiments. The original RAM uses multi-resolution glimpses at each time step to achieve higher classification accuracy. The glimpse of the lowest resolution can cover almost the entire image. This setting compromises the quality of the attention policy. To focus on policy learning in this work, we use a single small glimpse of size 4×4 at each time step. The idea of not using multi-resolution glimpses has been used by Elsayed et al. [2019] for better interpretability. In our experiments, RAM models are trained using REINFORCE [Williams, 1992] and optimized by Adam [Kingma and Ba, 2015] for 1500 epochs with a batch size of 128 and a learning rate of 3e-4.

DT-RAM The DT-RAM model used in the experiments is from our own implementation. Instead of using two separate policy networks for location prediction and task termination respectively, which were designed for curriculum learning in the original DT-RAM, we use an integrated policy network for both location prediction and task termination. Same as RAM, the glimpse size is 4×4, and the maximum number of movement steps is 18 for DT-RAM. In our experiments, DT-RAM models are trained for 1500 epochs with the same optimization configuration as RAM models.

[6] https://github.com/kevinzakka/recurrent-visual-attention

Model for Unsupervised Skill Discovery The implementation of the model for unsupervised skill discovery is based on the code repository[7] of Strouse et al. [2022]. In this implementation, the model uses the last state as an abstraction of the trajectory. The model is trained using a distributed actor-learner setup similar to R2D2 [Kapturowski et al., 2019]. The Q-value targets are computed with Peng's $Q(\lambda)$ [Peng and Williams, 1996] instead of n-step double Q-learning. Following Strouse et al. [2022], performance of the agent is evaluated using the number of learned skills

$$n_{\text{skills}} = 2^{\mathbb{E}[\log q_\phi(y|\tau) - \log p(y)]}, \tag{6.10}$$

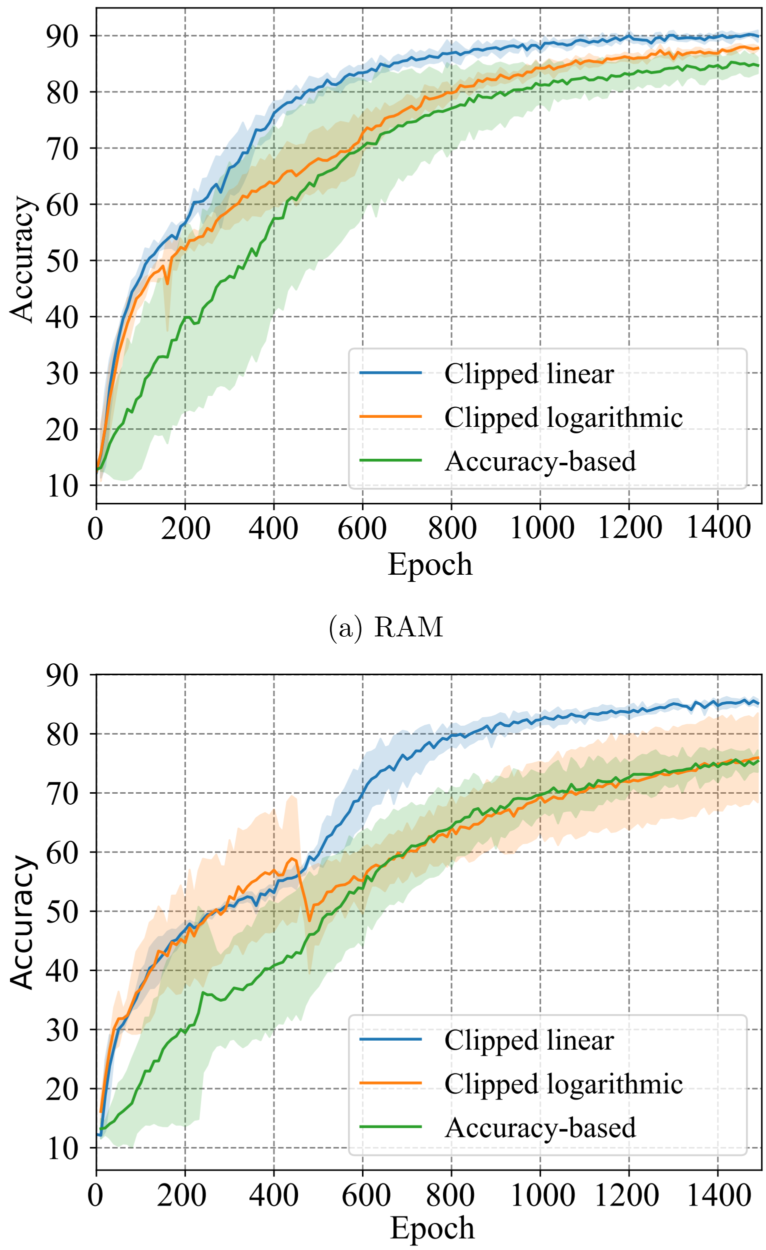
which can be understood as the measurement of the logarithmic reward in bits.

Model for Object Counting The implementation of the model for robotic object counting is based on the code repository[8] of Li et al. [2021]. We replace the REINFORCE algorithm with PPO for more efficient training. The implementation of the PPO algorithm is based on the code repository[9] of Chevalier-Boisvert et al. [2019]. The model consists of a pretrained and fixed ResNet18 [He et al., 2016] to extract feature maps from its *conv3* layer. The feature maps are then passed through two CNN layers and an average pooling layer to get visual representations of dimension 256. The index of the target object is mapped into a 10-dimensional embedding, which is called the goal representation. The visual and goal representations are concatenated together as the input of an RNN network, which recurrently produces hidden representations at each time step for the policy network and classification network. When the policy network selects the *stop* action, the classification network is triggered to produce the prediction of the number of the target object. We train the model for 2M episodes. Five processes are used to collect experience with a horizon of 40 steps. We train the model using Adam [Kingma and Ba, 2015] with a learning rate of 1e-4. Other hyperparameters of PPO are the same as the original implementation[9] except that we use 10 epochs of minibatch optimization and 5 parallelization processes.

6.6.4 Baseline Comparison

Baselines We compare the proposed clipped linear reward function with alternative reward functions. The first is the *accuracy-based reward function* r_{acc} in Eq. (6.1). The second is the *logarithmic reward function* based on Shannon's mutual information. Instead of using the original logarithmic reward function (Eq. (6.4)), we use a clipped variant, i.e., $\overline{r_{\log}} = \max(\log q_\phi(y \mid \tau) - \log p(y), 0)$, for fair comparison with our clipped linear reward function. We found that reward clipping generally results in similar or better performance in our experiments,

[7] https://github.com/deepmind/disdain
[8] https://github.com/mengdi-li/robotic-occlusion-reasoning
[9] https://github.com/mila-iqia/babyai



(b) DT-RAM

Figure 6.8: Comparison between the clipped linear reward function (■) with baselines, including the clipped logarithmic (■) and the accuracy-based (■) reward function, on the task of hard attention for digit recognition using RAM and DT-RAM. All the experiments in this work ran over three random seeds. Lines and shaded areas show the mean and standard deviation over multiple runs.

which is consistent with the empirical finding by Strouse et al. [2022]. The empirical study of reward clipping is provided in Sec. 6.6.7. On the skill discovery task, we additionally compare our reward function with the state-of-the-art *DISDAIN* *reward function* [Strouse et al., 2022], which was designed specifically to mitigate the pessimistic exploration issue in this task. The reward of the DISDAIN method is $r = r_{\log} + \lambda r_{\text{DISDAIN}}$, where $r_{\log}$ is the logarithmic reward function (cf. Eq. (6.4)), λ is a weighting coefficient, and r_{DISDAIN} is an auxiliary ensemble-based reward calculated as

$$r_{\text{DISDAIN}} = \mathbb{H}\left[\frac{1}{N}\sum_{i=1}^{N} q_{\phi_i}(y \mid \tau)\right] - \frac{1}{N}\sum_{i=1}^{N} \mathbb{H}\left[q_{\phi_i}(y \mid \tau)\right], \tag{6.11}$$

where N is the number of discriminators of the ensemble, and $\mathbb{H}[X]$ is the entropy of random variable X. The DISDAIN reward is essentially the estimation of the epistemic uncertainty of the discriminator.

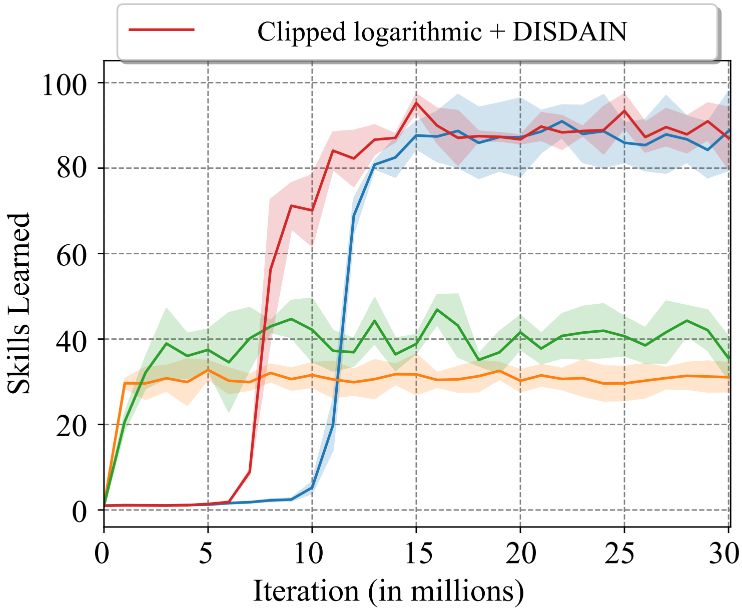
Comparison Results Fig. 6.8 shows that both RAM and DT-RAM trained using the clipped linear reward function achieve the highest accuracy and fastest training speed. Furthermore, the small blue shaded areas indicate that multiple runs using the clipped linear reward function are consistent with each other, which suggests high stability of the training process. Fig. 6.9a demonstrates that the clipped linear reward function outperforms both the clipped logarithmic reward function and the accuracy-based reward function by a large margin and achieves almost the same performance as DISDAIN. We note that the DISDAIN method depends on an ensemble of discriminators and needs more hyper-parameters to tune, e.g., the weight of the DISDAIN reward and the number of ensemble members, while our method is much simpler. Fig. 6.9b shows that the clipped linear reward function also benefits the challenging robotic object counting task by making the model converge faster and achieve the highest final accuracy.

We can see that the clipped linear reward function generally outperforms the logarithmic and the accuracy-based reward function. The improvement is significant on the skill discovery task, which makes sense according to our theoretical analysis in Sec. 6.5.4. Since the number of possible discrimination classes in the skill discovery task (128 classes) is much larger than that of other tasks (10 classes in the digit recognition task, and 7 classes in the robotic object counting task), $p(y \mid \tau)$ tends to be closer to zero when trajectory τ contains a small amount of information for discrimination in the skill discovery task. Thus the expectation and variance of the noise of the logarithmic reward function are larger, resulting in a severer unstable training issue, while our clipped linear reward function resulting in low expectation and variance of the reward noise still performs well.

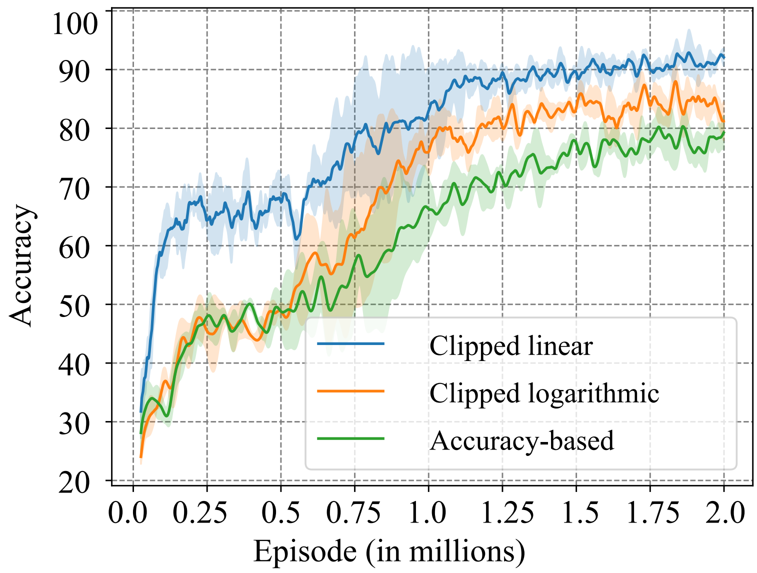
Interesting case studies for the digit recognition task, the object counting task, and an intuitive comparison of state occupancy in the unsupervised skill discovery task are given in Sec. 6.6.8.

6.6.5 Effect of Noise Moderation

Following the experimental setup in Sec. 6.4.2, we conduct reward hacking experiments using the clipped linear reward to visualize its capability in narrowing the



(a) Unsupervised skill discovery



(b) Robotic object counting

Figure 6.9: Comparison with baselines on the tasks of unsupervised skill discovery and robotic object counting. On the unsupervised skill discovery task, the auxiliary DISDAIN reward (■) is compared. Legends are shared between the two sub-figures.

gap between training processes with and without reward hacking (cf. Fig. 6.4). Fig. 6.10 shows the training curves. In order to facilitate a comprehensive comparison, we incorporate training curves when using the accuracy-based and the logarithmic reward (cf. Fig. 6.5) into the figure. We can see from Fig. 6.10 that when using reward hacking, all three rewards perform similarly (see dashed lines).

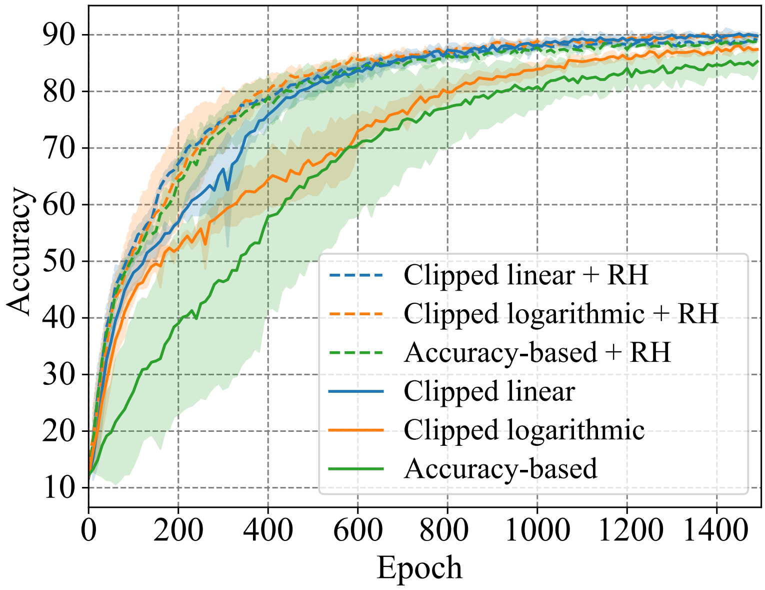


Figure 6.10: RAM trained using the three kinds of reward functions with and without reward hacking (RH). The clipped linear reward function achieves a much smaller gap between the training processes with and without reward hacking.

This suggests that the linear function performs as well as the logarithmic function in terms of information transmission. However, when not using reward hacking, the training curve of the clipped linear reward is much closer to the training curve of using reward hacking, compared to the other two rewards. This suggests that the advantage of the clipped linear reward function is due to the reduction of the impact of reward noise.

6.6.6 Evaluation of Various g Functions

We evaluate several other g functions in addition to the linear and logarithmic functions using the RAM model on the digit recognition task. Fig. 6.11 illustrates the clipped generalized reward with respect to the estimated posterior probability when using different g functions (cf. Eq. (6.5)). The reward is clipped at $q_\phi(y \mid \tau) = p(y) = 0.1$ (cf. Eq. (6.9)). Fig. 6.12 shows training curves when using different g functions. We can see that the linear function results in the best performance, and g functions that are similar in shape to the linear function generally perform well. The logarithmic function and function $g(x) = x^6$ perform worse than others, which can be explained from the perspective of the requirements of g functions. Though the logarithmic function works ideally in information transmission in theory where noise is not an issue, it suffers from noisy rewards as discussed in Sec. 6.5. Function $g(x) = x^6$, on the other hand, leads to a small bias and variance of the estimated reward, which suggests a favorable ability in noise moderation. However, it cannot transmit information with high fidelity. Its incompetence in information transmission can be observed from the shape of the corresponding plot in Fig. 6.11, where a wide range of values, e.g., [0, 0.5], is compressed to values close to zero, leading to a substantial ignorance of information in various observations. In contrast, the linear function achieves a trade-off between these two abilities and exhibits the best performance among all the g functions considered.

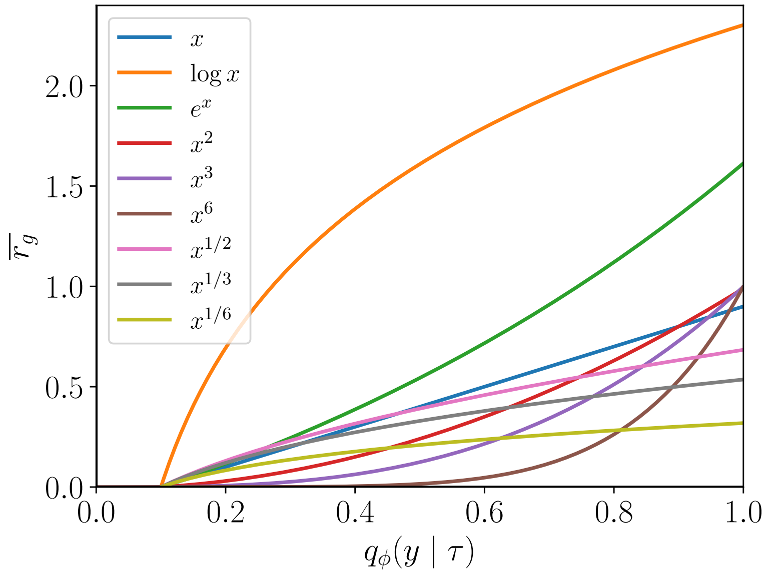


Figure 6.11: Clipped generalized reward with respect to the estimated posterior probability.

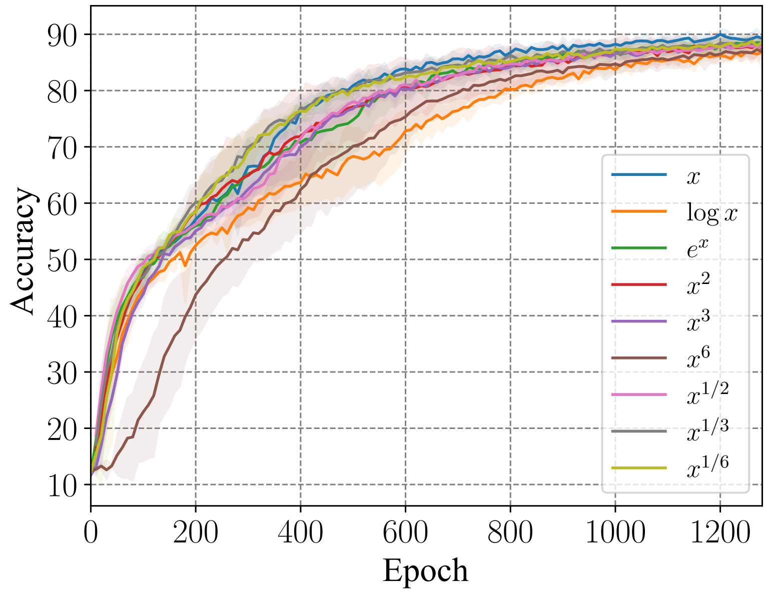


Figure 6.12: Evaluation of various g functions using the RAM model on the digit recognition task.

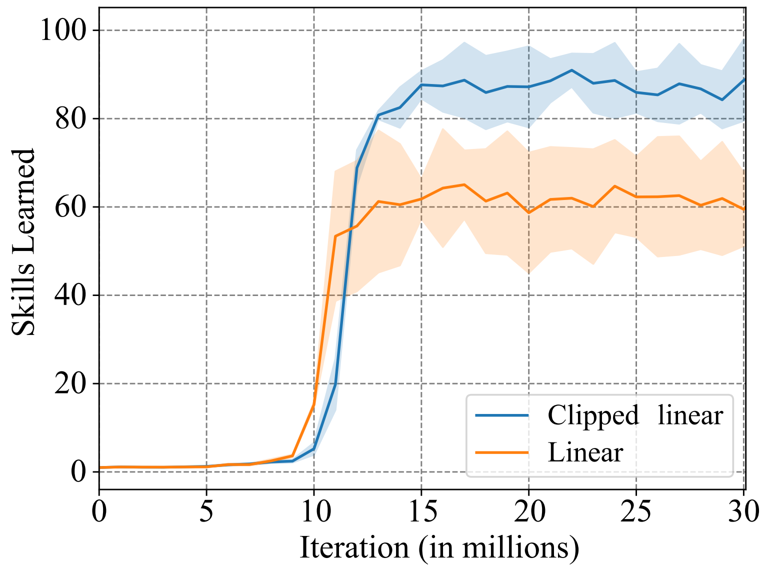


Figure 6.13: Reward clipping on the unsupervised skill discovery task.

6.6.7 Reward Clipping

We compare the performance of models trained using the logarithmic and the linear reward with and without reward clipping. Experimental results are that the clipped logarithmic reward achieves almost the same performance on the digit recognition task on both RAM and DT-RAM, slightly better performance on the skill discovery task ($\sim$ 1.5 more learned skills), and slightly worse performance ($\sim$ 3.5% lower accuracy) on the object counting task. The clipped linear reward achieves almost the same performance on the object counting task, slightly better performance ($\sim$ 1% and $\sim$ 1.5% higher accuracy on RAM and DT-RAM respectively) on the digit recognition task, and considerable improvement ($\sim$ 23 more learned skills) on the unsupervised skill discovery task (see Fig. 6.13). These results suggest that reward clipping is a generally beneficial technique, which is consistent with our theoretical analysis in Sec. 6.5.6.

6.6.8 Case Study

Hard attention for digit recognition

In Fig. 6.14, we provide cases of the DT-RAM model on the digit recognition task for intuitive comparison between models trained using different reward functions. All the cases are randomly sampled without any cherry-picking. We can see that trajectories generated by the model trained using the clipped linear reward can cover sufficient information for recognizing the digit, while trajectories generated by the model trained using the logarithmic reward function tend to be pessimistic, e.g., trajectories in cases of digit 9 and digit 6 in the first row, digit 0 in the second row, and digit 4 in the third row. The exploration trajectories generated by the model trained using the accuracy-based reward tend to sample less informative areas, e.g., trajectories in cases of digit 6 in the first row, and digit 2 in the third row and second column, which may account for its low accuracy.

Robotic object counting

Fig. 6.15 shows examples of the pessimistic exploration issue when using the logarithmic reward function and the accuracy-based reward function. The agent trained using the accuracy-based reward function chooses not to move, and the agent trained using the logarithmic reward function terminates exploration too early to acquire sufficient information for predicting the number of the target object. They guess the number of target objects based on insufficient observations, while the agent trained using the clipped linear reward function learns to choose a reasonable number of movement steps to explore the environment.

Unsupervised Skill Discovery

Fig. 6.16 demonstrates state occupancy reached using different reward functions at initialization, at the intermediate stage, and at convergence during training. We can see that using the clipped linear reward function, the agent learns to reach all states as using the DISDAIN reward, while the agent mainly explores the first room when using the clipped logarithmic reward function.

6.7 Discussion

6.7.1 Interpretation from the Information-theoretic Perspective

The linear reward function has specific meanings from an information-theoretic perspective. It can be derived from the optimization objective of maximizing the χ^2-divergence, one of the f-mutual information measures [Csiszár, 1972, Esposito et al., 2020], instead of the commonly used KL-divergence corresponding to Shannon's mutual information [Shannon, 1948] (cf. Eq. (6.2)). The derivation is provided below.

We use the optimization objective of maximizing the f-mutual information between the observation trajectory τ and the target class y in place of the objective of Eq. (6.2) and obtain

$$\begin{aligned} I_f(y;\tau) &:= D_f(p(y,\tau)\,||\,p(y)p(\tau)) \\ &= \mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)} F\left(\frac{p(y\mid\tau)}{p(y)}\right), \end{aligned}$$

where $D_f(P\,||\,Q) := \mathbb{E}_{q(x)} f\left(\frac{p(x)}{q(x)}\right) = \mathbb{E}_{p(x)} F\left(\frac{p(x)}{q(x)}\right)$ is the f-divergence of two probability distributions P and Q on X, with $f : \mathbb{R}^+ \to \mathbb{R}$ being a generic convex function satisfying $f(1) = 0$, $F(x) := f(x)/x$ for simplicity of expectation over P

instead of Q for later use, and $p(x)$ and $q(x)$ are probability density functions of P and Q respectively. By choosing $f(x) = x \log x$, f-divergence becomes the well-known Kullback–Leibler divergence and, correspondingly, the f-mutual information is then Shannon's mutual information [Shannon, 1948, Kinney and Gurinder S. Atwal, 2014, Belghazi et al., 2018]. Other typically used f-divergences and their expected mutual information over $p(x, y)$ are listed in Table 6.1. When using the χ^2-divergence, i.e., $f(x) = (x-1)^2$, f-mutual information becomes

$$I_f(y;\tau) = \mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)}\left[\frac{p(y \mid \tau)}{p(y)} - 1\right].$$

When y is sampled from a uniform distribution, i.e., $p(y)$ is a constant, we have $I_f(y;\tau) = \alpha\, \mathbb{E}_{\tau\sim\pi_\theta, y\sim p(y)}\left[p(y \mid \tau) - p(y)\right]$, where $\alpha = 1/p(y)$. Following the derivation of Eq. (6.3), this optimization objective induces the linear reward function in Sec. 6.5.4.

Table 6.1: f-mutual information and the corresponding convex functions

f-divergence	$f(x)$	$I_f(x;y)$
Kullback–Leibler	$x \log x$	$\mathbb{E}_{p(x,y)} \log \frac{p(y\|x)}{p(y)}$
χ^2	$(x-1)^2$	$\mathbb{E}_{p(x,y)} \frac{p(y\|x)}{p(y)} - 1$
Total Variance	$\frac{1}{2}\|x-1\|$	$\mathbb{E}_{p(x,y)} \frac{1}{2}\left\|1 - \frac{p(y)}{p(y\|x)}\right\|$
Squared Hellinger	$(1-\sqrt{x})^2$	$\mathbb{E}_{p(x,y)}\left[2 - 2\sqrt{\frac{p(y)}{p(y\|x)}}\right]$
Le Cam	$\frac{1-x}{2x+2}$	$\mathbb{E}_{p(x,y)} \frac{[p(y\|x)-p(y)]^2}{2p(y\|x)+2p(y)}$
Jensen Shannon	$x \log \frac{2x}{x+1} + \log \frac{2}{x+1}$	$\mathbb{E}_{p(x,y)}\left[\log \frac{2p(y\|x)}{p(y\|x)+p(y)} + \frac{p(y)}{p(y\|x)} \log \frac{2p(y)}{p(y\|x)+p(y)}\right]$
Reverse KL	$-\log x$	$\mathbb{E}_{p(x,y)}\left[\frac{p(y)}{p(y\|x)} \log \frac{p(y)}{p(y\|x)}\right]$

In recent years, f-mutual information has been studied in many deep learning applications, such as generative models [Nowozin et al., 2016, Gimenez and Zou, 2022], representation learning [Lotfi-Rezaabad and Vishwanath, 2020, Mittal et al., 2023], image classification [Wei and Liu, 2021], imitation learning [Zhang et al., 2020], etc. Wei and Liu [2021] suggested that a properly defined f-divergence measure is robust with label noise in a classification task, which is related to our finding that the χ^2-mutual information is a more robust information measure against the inherent noise in the policy learning of IRRL compared to Shannon's mutual information. This leads to interesting future work on investigating principles for selecting the optimal f-mutual information measure, and the possibility of using other f-mutual information measures for achieving more stable IRRL.

6.7.2 Limitations and Future Work

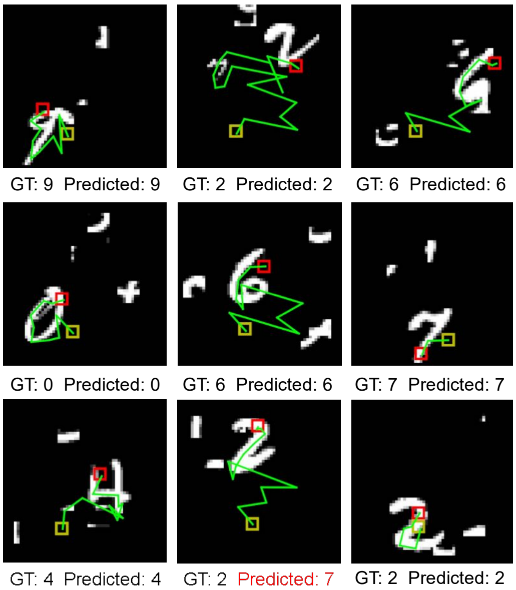
This work is an early step towards stabilizing IRRL. Some identified limitations potentially lead to interesting future work. First, we evaluated the efficiency of the suggested reward functions within a subset of IRRL scenarios. It is appealing to study the generalizability and explore the potential adaptions across a wider spectrum of applications, e.g., RLHF for large language models finetuning. Second, we only considered classification-based reward models but not regression-based ones. A unified guideline for designing reward functions in both cases would be significant. Third, we stabilized the training process of IRRL from the perspective of reducing the impact of reward noise without explicitly considering reducing the impact of insufficient observations (see Fig. 6.1). An integrated method considering both issues should lead to a more optimal solution.

6.8 Summary

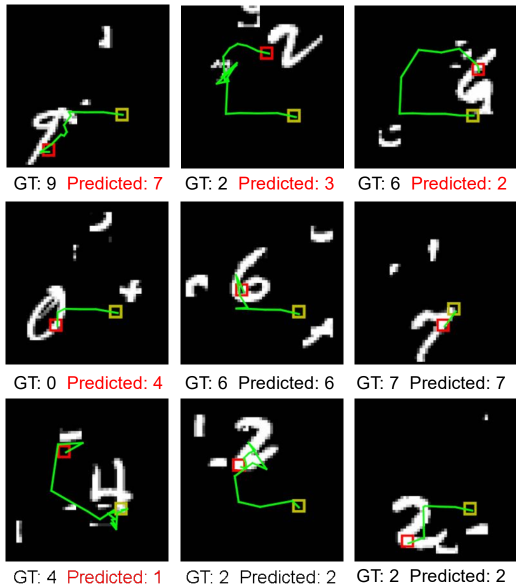
In this chapter, we attempt to answer the third research question "*How to stabilize the reinforcement learning process of the active vision control policy to make the training more efficient?*" We generalized the reinforcement learning process of the active vision control policy into a class of reinforcement learning problems, namely Internally Rewarded Reinforcement Learning (IRRL), where the policy is trained using reward signals from the feedback of a discriminator-based reward model, which is simultaneously optimized in a supervised learning manner using information collected by the policy. The inherent issues of noisy rewards and insufficient observations in the training process of IRRL lead to an unstable training loop where neither the policy nor the discriminator can learn effectively. Based on theoretical analysis and empirical studies, we proposed the clipped linear reward function to reduce the impact of reward noise.

Extensive experimental results suggested that the proposed method consistently stabilizes the training process and achieves faster convergence and higher performance compared with baselines in diverse tasks. Additionally, we gave an interpretation of the use of the linear reward function from the information-theoretic perspective, which suggested an interesting future research direction. As interest grows in integrating the capability of high-level prediction and low-level control of behaviors into a single model, for instance in embodied AI, robotics, and unsupervised RL, stable and efficient training of IRRL will be particularly relevant.

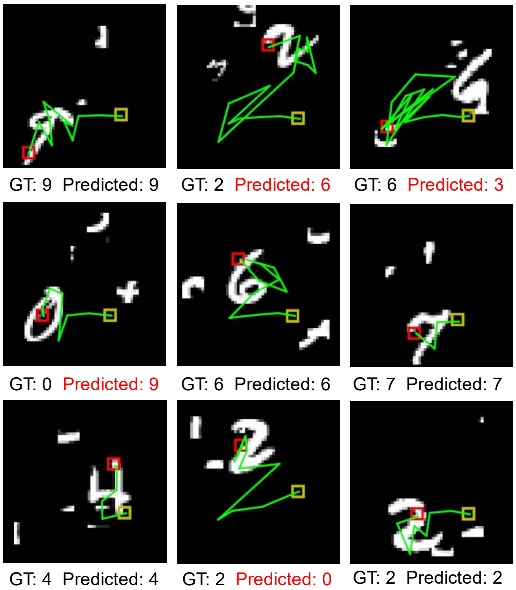
We acknowledge that the proposed reward function is not guaranteed the optimal solution for the unstable training issue of IRRL. Future research is necessary to enhance the stable training of such models, for example on more theoretical analysis of the influence of the variance and bias of rewards on the training stability.



(a) Clipped linear



(b) Clipped logarithmic



(c) Accuracy-based

Figure 6.14: Comparison of DT-RAM models trained by different reward functions. Trajectories generated by the model trained using the *clipped linear reward* can cover sufficient information for recognizing the digit, while trajectories generated by the model trained using the *logarithmic reward* function tend to be pessimistic. The exploration trajectories generated by the model trained using the *accuracy-based reward* tend to sample less informative areas. The starting and stopping glimpses are represented by yellow and red boxes respectively. The green line indicates the positions of intermediate glimpses. GT: the ground-truth class; Predicted: the predicted class (red indicates incorrect predictions).

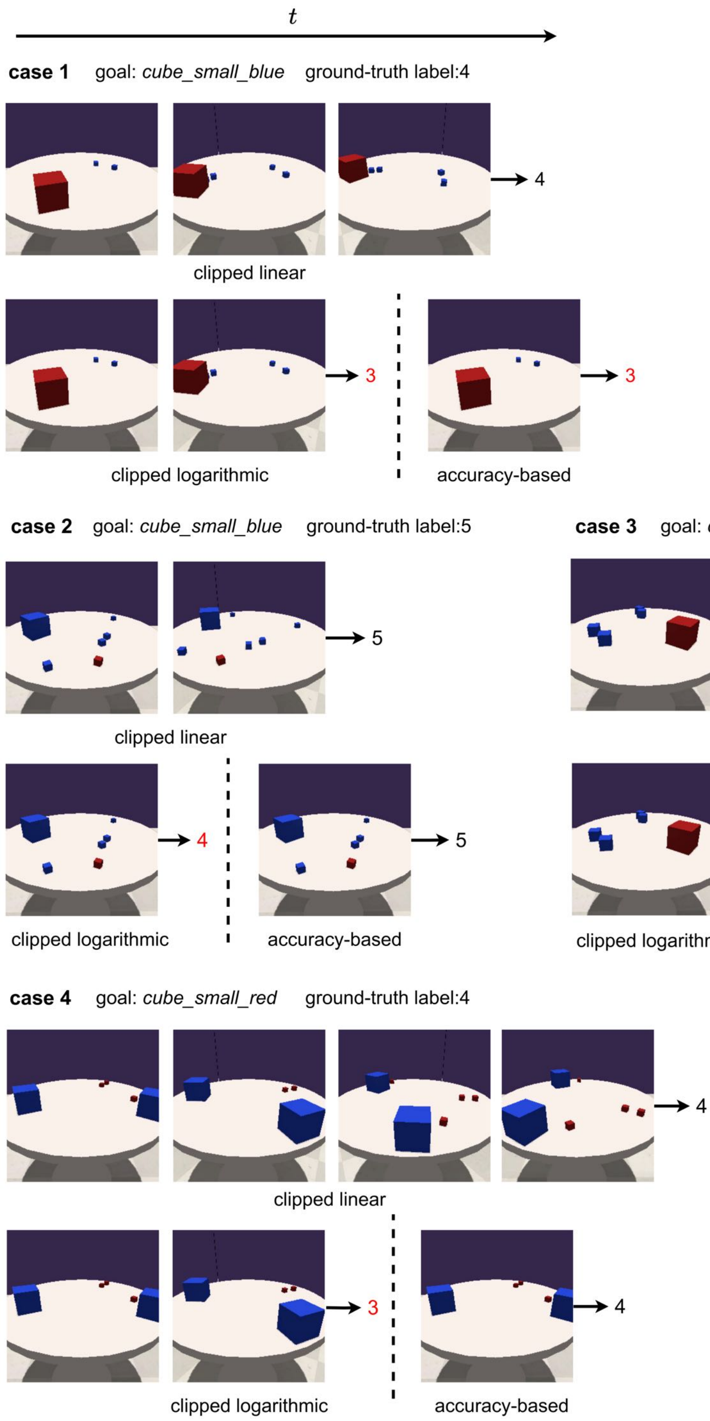
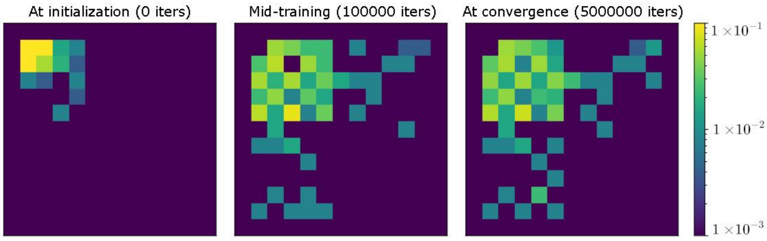
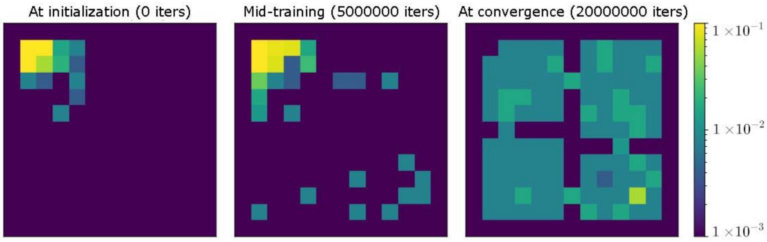


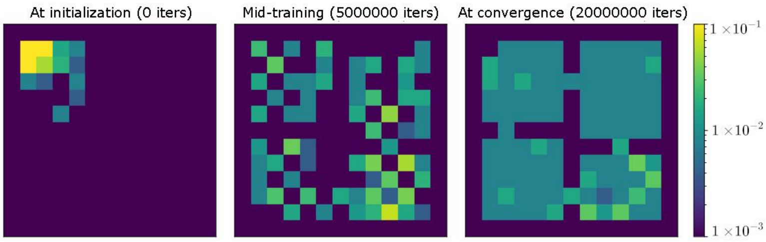
Figure 6.15: Comparison of models trained by different reward functions on the robotic object counting task. The number next to the arrow after a sequence of egocentric views is the number of goal objects predicted by the agent. Red numbers indicate wrong predictions.



(a) Clipped logarithmic



(b) DISDAIN



(c) Clipped linear

Figure 6.16: States reached by agents that are trained with different reward functions. Plots depict ratios of final states reached after performing 10 trajectories per skill (the ratio is clipped between 0.001 and 0.1 for the sake of visualization). The diversity of states reached by the agent reflects the diversity of skills the agent learns. We expect that the agent reaches as diverse states as possible. The agent trained with the *clipped logarithmic* reward function mostly reaches the states of its starting room (the top-left room) and cannot reach the bottom-right room even when the training is converged (see the figure in the first row and third column). However, the agent trained using the *DISDAIN* and *clipped linear* reward is able to near-uniformly reach all the states of the four rooms (see figures in the last two rows and third column). The agent using the *clipped linear* reward learns faster compared to the agent using *DISDAIN*, as it can reach diverse states faster during training (see figures in the last two rows and second column).

Chapter 7

Discussions and Conclusions

7.1 Reinforcement Learning for Embodied Agents

It has been demonstrated in this thesis that the technique of reinforcement learning still has great potential to improve. The motivation to create embodied AI agents is a driving force towards advancement in this area. The integration of sensory perception, attention, and value evaluation into the reinforcement learning framework facilitates the emergence of embodied agents that are generalizable, versatile, and robust enough to be deployed into realistic human environments.

This thesis suggests that reinforcement learning is a versatile computational approach that can be employed in the development of the active vision control policy of action-response agents. Conventional reinforcement learning methods are not suitable for achieving this goal, due to the complexity of tasks involving embodied agents, compared with tasks on game environments that were extensively studied in the literature. A more generalized reinforcement learning framework, which considers the internal processing of external stimuli, attention mechanisms, and goal-oriented value evaluation, was proposed and we showcased its effectiveness in the learning of active vision control.

This extension of reinforcement learning incurs specific challenges in model optimization. Compared to the conventional reinforcement learning framework, where the reward signals are directly produced by the external environment according to predefined roles with human expertise, reinforcement learning utilizing automatically generated reward signals, which depend on perception and value evaluation components, is more complicated and hard to converge. We demonstrated that the shape of the reward function of the reward model in the internal environment impacts the learning stability of the reinforcement learning process and proposed methods to mitigate this issue. We believe that more work is needed to improve the proposed learning framework, to improve its efficiency and scalability, and to extend the application scope of this framework.

In this thesis, we only used small-scale tasks to evaluate the effectiveness of the proposed approach. This leads to another point of our discussion about the scalability of the proposed method in the next section.

7.2 Scalability

In this thesis, we aimed to design models for action-response embodied agents and train the active vision control policy using reinforcement learning. The feasibility of this idea is demonstrated in small-scale environments, however, we question the scalability of this approach in large-scale settings when applied in isolation. Developing action-response agents within large-scale environments poses a significant challenge than in small-scale settings, as the more comprehensive capabilities of the agents are necessary. On the other hand, compared to the development of action-only agents, like the point-goal navigation agent where the scalability of an RL-based method has been demonstrated [Wijmans et al., 2020], action-response agents in large-scale environments require more comprehensive high-level capabilities, which are pretty challenging to acquire through pure RL.

The paradigm of unsupervised learning followed by supervised learning and reinforcement learning could be the ideal pathway to robust and generalized AI agents. Compared to RL, unsupervised learning is more suitable for learning robust representations and behaviors, particularly when processing multi-modal information in complex, large-scale environments. Supervised learning, on the other hand, offers higher data efficiency in acquiring task-specific skills. Nevertheless, the role of RL still remains essential. Its benefits and advantages are indispensable in the learning of low-level interaction behaviors guided by a high-level task objective.

However, though remarkable success has been achieved in pretrained foundation models in natural language processing and computer vision, developing pretrained embodied agents through unsupervised learning is still a big challenge and a long-standing goal in the community.

7.3 Trustworthy AI Models and Agents

Beyond the specific goal of this thesis on developing action-response embodied agents with active vision, there lies a more general objective which is to develop trustworthy AI models and agents. They are expected to produce helpful, harmless, and ethical responses in realistic applications. As studied in this thesis, a VQA model should be able to abstain from answering when the visual question from the user is irrelevant to the given visual content to avoid the producing of potentially harmful answers; an embodied agent should be able to evaluate the information sufficiency and perform actions to actively collect more necessary information for the given task when the collected information is deemed inefficient.

The capabilities of *information sufficiency evaluation* and *interactive information collection* play essential roles in achieving trustworthy AI models and agents. In this thesis, the capability of information sufficiency evaluation is obtained through supervised learning, and the evaluation result is represented by the uncertainty of the prediction of the ground-truth label. The capability of interactive information collection is developed through reinforcement learning, through which an interactive information collection policy is trained. This methodology is actually aligned with recent work on trustworthy chatbots [Ouyang et al., 2022, Touvron et al., 2023]. Researchers employed supervised learning to train a reward model to evaluate the human preference for responses given by the chatbot, analogous to information sufficiency evaluation, and used reinforcement learning to finetune the chat policy, aiming at improving human preference, as indicated by a preference score predicted by the reward model. The resulting chat policy thus adopts strategies like abstaining from responding to questions that could lead to illegal answers, asking for more details in the case of ambiguous queries, or directly pointing out the conflict with the truth when questions involve counterfactual premises. Different from our methodology, the fundamental models for these chatbots are obtained through unsupervised pretraining on large-scale unlabeled data. The success of this methodology in the application of chatbots points out a promising direction in the development of trustworthy embodied agents.

7.4 Reinforcement Learning with Reward Models

In this thesis, stemming from our primary interest in the application of RL for active vision control learning, we also studied the fundamental challenge of unstable training in RL with discriminator-based reward models. The framework of RL with reward models has been applied in tasks across diverse domains involving interaction between AI agents and environments or users. As application scenarios of AI models extend, this framework appears to have a broader range of applications.

Beyond the methods proposed in this thesis, there remains considerable potential for exploration in addressing the issue of unstable training. We discuss two research areas below where the framework of RL with reward models is applied.

Unsupervised Skill Discovery As already discussed in Chapter 6, existing methods in the task of unsupervised skill discovery involve the paradigm of RL with reward models: a discriminator is simultaneously optimized with the skill policy in supervised learning and provides reward signals for policy learning. Efficient optimization of these methods is still challenging. For example, recent works in this area argue that mutual information maximization is not an appropriate training objective based on the observation that this objective does not differentiate skills in different scales thus limiting the diversity of learned skills [Park et al., 2022, 2023]. Based on the findings of this thesis, it is reasonable to hypothesize that this challenge could be possibly alleviated from the perspective of mitigating the unstable training loop that is caused by the issue of noisy rewards and insufficient observations.

RLHF for AI Alignment AI alignment, with the goal of aligning AI models with human values, has been attracting increasing attention recently [Norvig, 2019, Casper et al., 2023]. Reinforcement learning from human feedback (RLHF) has been demonstrated as an effective method for achieving this goal, as seen in its successful application in chatbots such as ChatGPT [Ouyang et al., 2022] and Llama 2 [Touvron et al., 2023]. RLHF for LLMs finetuning involves RL training of the chat policy guided by reward signals from a reward model that is trained in a supervised learning manner using human-annotated data. While effective, it has been recognized as a challenging task to jointly train a chat policy and a reward model [Rafailov et al., 2023, Casper et al., 2023]. This challenge poses barriers to human control over the behavior of large language models and other pretrained large-scale models. It would be significant to propose guidelines for simplifying the training of RLHF in such applications.

7.5 Conclusions

In conclusion, this thesis contributed to the development of the active vision control policy of action-response embodied agents. An embodied agent with active vision that is capable of actively collecting visual information for producing reliable responses can be modeled using a modular network, of which the active vision control policy is optimized using reinforcement learning guided by reward signals from the response module, and the response module is trained in a supervised learning manner using visual information collected by the active vision control policy. This method can be generalized and formulated as a novel reinforcement learning framework, which is characterized by a jointly optimized discriminator-based reward model that provides reward signals for policy learning. The application of this framework is widespread, however, its learning process suffers from the inherent issue of an unstable training loop, where an under-optimized reward model yields noisy rewards, and in turn, an immature policy yields insufficient observations. It was demonstrated empirically and theoretically that the use of a clipped linear reward function instead of the commonly used logarithmic reward function alleviates this issue, stabilizing the training process.

Appendix A

LLM-based Active Vision for Robotic Object Existence Prediction

A.1 Introduction

In this part, we implement a robot demo controlled by a large language model (LLM)-based system for active vision control. The system diagram is illustrated in Fig. 2.5 in Chapter 2. In line with our previous work on the reinforcement learning-based method for active vision control, which is introduced in Chapter 5, we utilize the task of Robotic Object Existence Prediction (ROEP) for evaluation. In this task, a robot standing next to a round table is asked to predict the existence of an object on the table. The robot can move around the table to observe the tabletop scene from various perspectives. The ROEP task is introduced in detail in Sec. 5.2. Specifically, we use a Pepper robot[1] equipped with a LiDAR scanner for the demo implementation. We test the robot on some ROEP task scenarios (see an example scenario in Fig. A.2) to evaluate the effectiveness of an LLM-based approach.

A.2 Methodology

The framework of our active-vision agent has a similar structure to the Matcha (**M**ultimod**a**l environmen**t** **cha**tting) agent [Zhao et al., 2023], where both of them use natural language as the intermediate representations for multimodal sensations and use LLMs to generate control commands for active information gathering. The diagram of our method is illustrated in Fig. A.1.

An **LLM** functions as the core of this method. The input of the LLM includes three sources, 1) the task-specific initial prompts, 2) users' questions, and 3) scene descriptions of the current view. The **initial prompt** contains four parts:

[1] https://www.softbankrobotics.com/emea/en/pepper

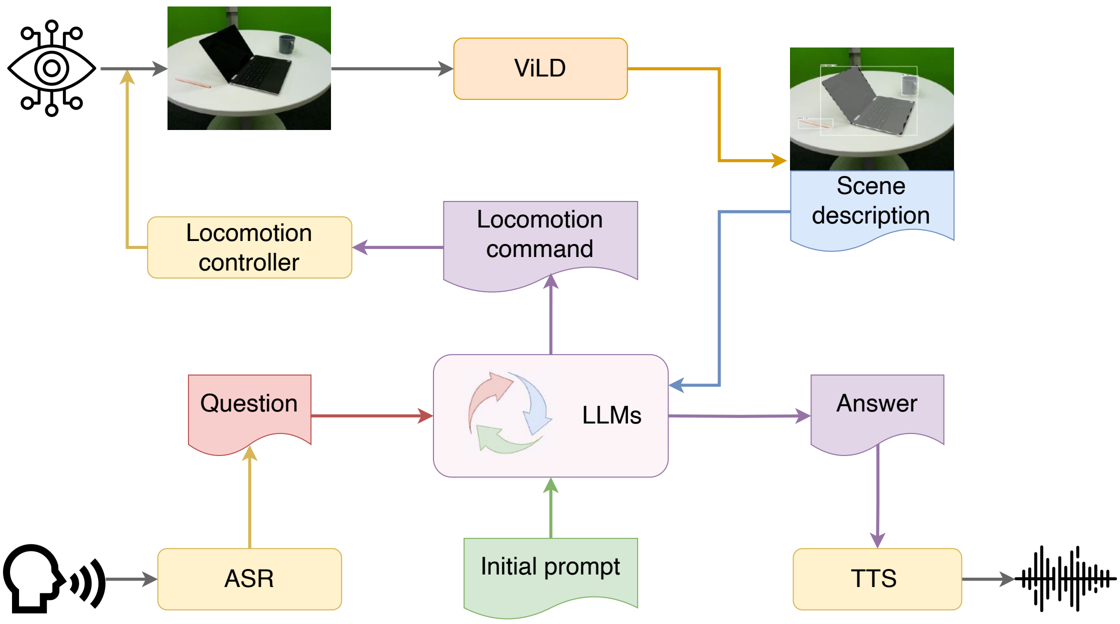


Figure A.1: Framework of the proposed LLM-based active-vision agent for the task of Robotic Object Existence Prediction (ROEP).

a) task descriptions; b) the robot's action space descriptions; c) examples; and d) additional prompts for fine-grained control. These prompts are shown in the following prompt snippets. The prompt of task descriptions mainly introduces the task setup, the general concept of occlusion between objects, and the expected behavior of the robot (see Prompt Snippet A). The prompt of the robot's action space descriptions introduces possible actions of the robot and the meaning of each action (see Prompt Snippet B). This informs the LLM of the set of possible actions to choose and the meaning of each action for better action planning. However, only given this general information, the reasoning chain generated by the LLM could be very diverse. Thus, we also give some examples to explicitly encourage the LLM to reason in a specific pattern. We give three examples in the prompt, each of which represents a class of scenarios (see Prompt Snippet C). These example scenarios are distinct from those scenarios used for evaluation, to avoid the contribution of memorization in task solving. We note that too many examples could harm the performance of the LLM in task-specific reasoning, as the LLM could be biased on the examples and overlook the general definition of the task and commonsense knowledge. Further, we add some additional prompts to impose some constraints on the output of the LLM based on our preliminary experiments (see Prompt Snippet D).

Users' **questions** are produced by an off-the-shelf Automatic Speech Recognition (ASR) model. The questions are queries about the existence of a type of object, e.g., "Is there an apple on the table". **Scene descriptions** are generated by an object detecter based on the first-person view of the Pepper robot regarding the tabletop. Specifically, visual information represented by RGB images is con-

Prompt Snippet A: Task Descriptions

A 120 centimeters-tall humanoid robot stands next to a 1-meter-high round table at a distance of 1 meter. The robot is expected to predict whether a target object is on the table or not.

Objects can be visually occluded by larger objects, for example, a key can be occluded by a mug because a key is visually smaller than a mug in size. Objects cannot be visually occluded by smaller objects, for example, a pear cannot be visually occluded by a pencil because a pear is visually larger than a pencil in size.

If the target object is observed, the robot should directly stop and predict "Yes".

If the target object is not observed, the robot needs to compare the size of the target object with each of the observed objects.

If the target object is smaller than one or some observed objects, the target object could be visually occluded by them from the current viewpoint of the robot. In this case, the robot should rotate to check the visually occluded space.

If the target object is larger than all the observed objects, the target object cannot be visually occluded by any of the observed objects, so the robot should directly stop and predict "No".

Prompt Snippet B: Robot's Action Space Descriptions

The robot can perform three actions:

1. "rotate_left": rotate 30 degrees around the table to the left.
2. "rotate_right": rotate 30 degrees around the table to the right.
3. "stop_and_predict": stop rotating around the table and predict the existence of the target object.

verted to language descriptions using a pretrained object detection model ViLD [Gu et al., 2022] as a list of detected objects, e.g., "a pencil, a mug, a laptop" (see Fig. A.3 for an example). Compared to conventional object detectors, ViLD works in an open-vocabulary manner, i.e., it is able to detect objects that are specified in a set of target objects. And because it was pretrained on large-scale datasets, it is highly robust to diverse scenes.

Taking advantage of the commonsense knowledge, multi-step reasoning capability, and in-context learning capability, the LLM reasons potential occlusion between the queried target object and observed objects, and generates a reasoning chain in natural language, including an **action command** selected from robot's action space descriptions in the initial prompt. Examples of the reasoning chain are shown in Sec. A.4. When the "stop_and_predict" action command is chosen, the LLM will generate the final **answer**. In our practical implementation, we let the robot speak out its reasoning chain in natural language using the built-in Text-To-Speech (TTS) service of the Pepper robot to make the system more transparent and trustworthy to the user.

The action commands "rotate_left" and "rotate_right" are executed by the **locomation controller** module to control the robot to move around the table. This module is hardcoded, utilizing the LiDAR signal for accurate rotation action control. The robot collects new observations after the execution of each rotation

Prompt Snippet C: Examples

Example 1
Question: Is there an apple on the table?
AI: Initialized the robot and observe the scene.
Observation: a coin, an apple, a bank card
AI: The robot observes three objects: [a coin, an apple, an bank card]. The target object is the apple. An apple is observed. So the robot chooses to stop to not waste time in further exploration. Therefore, the robot should take the action: stop_and_predict => The answer is: Yes
Terminate

Example 2
Question: Is there a pear on the table?
AI: The robot is just initialized and observing the scene.
Observation: a button battery, a bank card, a key
AI: The robot observes three objects: [a button battery, a bank card, a key]. The target object is the pear. A pear is not observed. Because pears are larger than all the observed objects [button batteries, bank cards, keys], a pear cannot be visually occluded by any of the observed objects. So the robot should choose to stop to not waste time in further exploration. Therefore, the robot should take the action: stop_and_predict => The answer is: No
Terminate

Example 3
Question: Is there a bank card on the table?
AI: Initialized the robot and observe the scene.
Observation: a laptop, a key, an electric kettle, an eraser
AI: The robot observes four objects: [a laptop, a key, an electric kettle, an eraser]. The target object is the bank card. A bank card is not observed. Among observed objects, bank cards are smaller than [laptops, electric kettles], so a bank card can be visually occluded by [laptops, electric kettles]. Bank cards have a similar size to [keys, erasers], so a bank card cannot be visually occluded by [keys, erasers]. So the robot should change its viewpoint to check the visually occluded space of the laptops and the electric kettle to see if there is a bank card. Therefore, the robot should take the action: rotate_right
Terminate

Prompt Snippet D: Additional Prompts

You will be the "AI" to assist the robot to achieve the task. The user will provide the "Observation". Stop generating when the first robot action is clear to execute and wait for the user to give you feedback. Do not generate fake "Observation".

action, and new scene descriptions will be generated by the ViLD module for the LLM to perform further occlusion reasoning.

A.3 Evaluation

We evaluate our method on scenarios with object settings (types and combinations) different from those used in the examples in the initial prompt to evaluate its generalizability. We use three experimental scenarios:

1. The target object can be directly observed;
2. The target object is not observed, however, it may be occluded by observed objects according to common sense;
3. The target object is not observed, and it cannot be occluded by observed objects according to common sense.

Detailed configurations of each of these experimental setups are introduced in Table A.1, where "Visible objects" are objects that are visible to the robot from its initial viewpoint, "Occluded object" is the object that is occluded by one of the visible objects from the initial viewpoint of the robot, "Queried object" is the object whose existence is queried by the user, "GT answer" is the ground-truth answer of the existence prediction.

Table A.1: Setups of three experimental scenarios

No.	Visible objects	Occluded object	Queried object	GT answer
1	Pencil, mug, laptop	Eraser	Pencil	Yes
2	Pencil, mug, laptop	Eraser	Eraser	Yes
3	Pencil, mug	Eraser	Laptop	No

A.4 Results

In Scenario 1, the initial state and observation of the robot are shown in Fig. A.2 and Fig. A.3, respectively. To answer the question of whether there is a "pencil" on the table, the robot performs the following reasoning chain and produces the final answer "Yes".

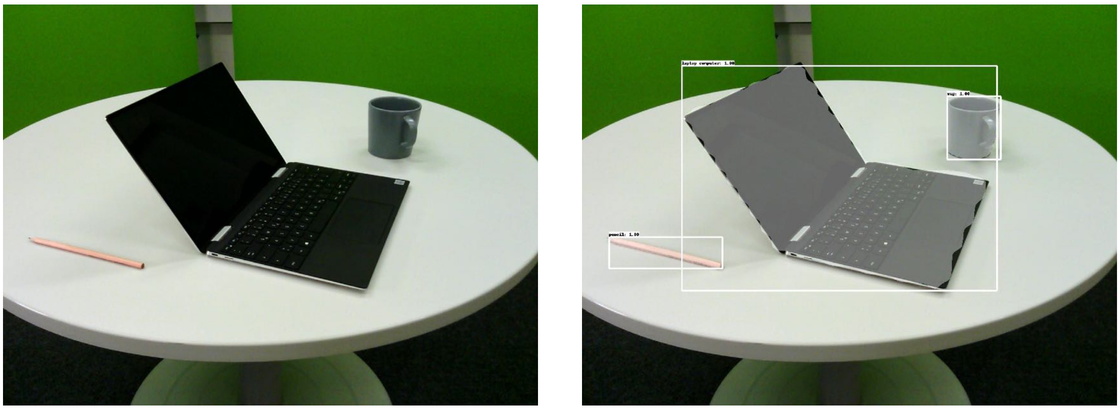
Scenario 1: Reasoning Chain

The robot observes three objects: [a pencil, a mug, a laptop computer]. The target object is a pencil. A pencil is observed. So the robot chooses to stop to not waste time in further exploration. Therefore, the robot should take the action: stop_and_predict => The answer is: YES!

In Scenario 2, the initial state and observation of the robot are identical to those in Scenario 1, since the only difference between them is the queried object (cf. Table. A.1). To answer the question of whether there is an "eraser" on the table, the robot performs the reasoning chain reported in "Scenario 2: Reasoning Chain (1/2)" and chooses to execute the action "rotate_right".

Instructed by the chosen action "rotate_right", the robot moves and observes the scene from a novel viewpoint. The state and observation of the robot after executing the chosen action are shown in Fig. A.4 and Fig. A.5, respectively. The robot performs the reasoning chain reported in "Scenario 2: Reasoning Chain (2/2)" and produces the final answer "Yes".

Figure A.2: The initial state of the robot in Scenario 1 and Scenario 2.



(a) First-person view of the robot

(b) Object detection result

Figure A.3: The initial observation of the robot in Scenario 1 and Scenario 2.

Scenario 2: Reasoning Chain (1/2)

The robot observes three objects: [a pencil, a mug, a laptop computer]. The target object is an eraser. An eraser is not observed. Among the observed objects, an eraser is smaller than a laptop computer. So an eraser could be visually occluded by a laptop computer from the current viewpoint of the robot. Therefore, the robot should take the action: rotate_right.

In Scenario 3, the initial state and observation of the robot are shown in Fig. A.6 and Fig. A.7, respectively. To answer the question of whether there is a "laptop" on the table, the robot performs the following reasoning chain reported in "Scenario 3: Reasoning Chain" and produces the final answer "No".

Scenario 2: Reasoning Chain (2/2)

The robot observes four objects: [a mug, a pencil, a laptop computer, an eraser]. The target object is an eraser. An eraser is observed. So the robot chooses to stop to not waste time in further exploration. Therefore, the robot should take the action, stop_and_predict => The answer is: YES!

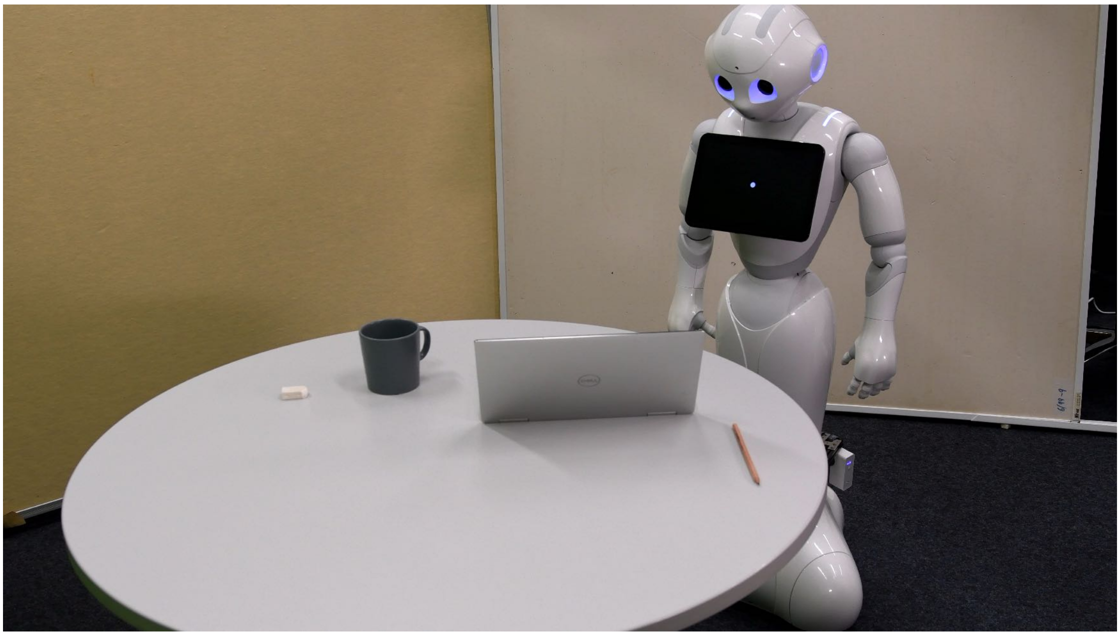


Figure A.4: The state of the robot after executing the action "rotate_right" in Scenario 2.

Scenario 3: Reasoning Chain

The robot observes two objects: [a pencil, a mug]. The target object is a laptop. A laptop is not observed. Among the observed objects, laptops are larger than both the pencil and the mug. Therefore, a laptop can not be visually occluded by any of the observed objects. So the robot chooses to stop to not waste time in further exploration. Therefore, the robot should take the action: stop_and_predict => The answer is: NO!

A.5 Conclusion and Discussion

Without the need for task-specified finetuning, the robot controlled by the LLM-based active vision system demonstrates effectiveness and generalizability in the task of ROEP. However, there are several challenges hindering the application of LLM-based approaches to scenarios in the wild. The first one is the achievement of more efficient vision understanding. Existing methods of vision-to-language conversion through separate models, such as object detection models or image captioning models, possess significant information loss. Natural language is not a high-fidelity representation of visual observations because of its discrete property. This poses a driving force for the advance of end-to-end visual-language models, where visual information and language information have a shared representation

(a) First-person view of the robot

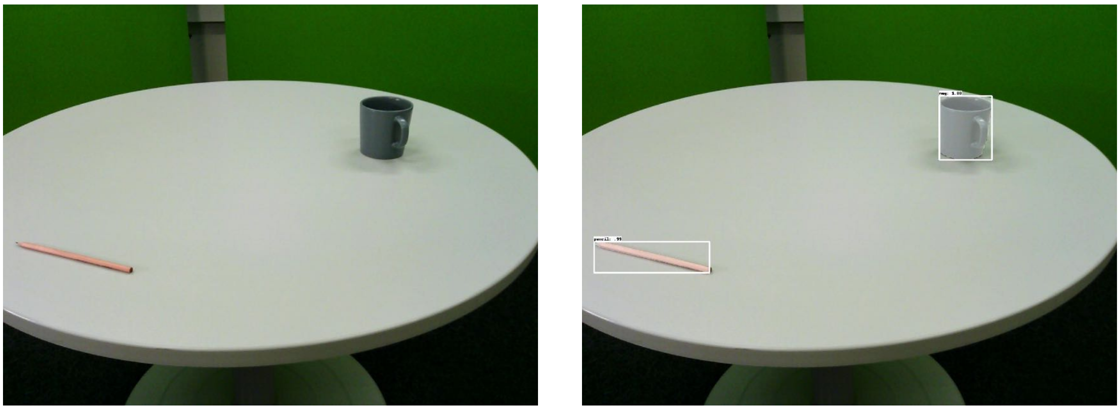
(b) Object detection result

Figure A.5: The observation of the robot after executing the action "rotate_right" in Scenario 2.

Figure A.6: The initial state of the robot in Scenario 3.

space. However, improving the reasoning capability of visual-language models is challenging [Chen et al., 2024].

The second challenge is enabling robots to perform more versatile physical interactions with their environments. Existing methods normally utilize hardcoded action policies [Zhao et al., 2023] or pretrained action policies through imitation learning [Ahn et al., 2022] to control robots based on action commands from LLMs. These methods have fixed action sets that are developed by human experts, which limits the generalization of LLM-based methods to diverse tasks and scenarios. Pretrained language-conditioned action policies that are highly generalizable to novel scenarios are appealing [Peng et al., 2022]. A unified pretrained model that



(a) First-person view of the robot (b) Object detection result

Figure A.7: The initial observation of the robot in Scenario 3.

can directly generate low-level control signals, e.g., the state of robotic graspers and arms, is another research direction to explore [Zitkovich et al., 2023].

Appendix B

Resulting Publications

Some of the concepts, methodologies, datasets, simulation environments, experimental designs, and findings introduced in this thesis were published in various international conferences.

- Mengdi Li, Xufeng Zhao, Jae Hee Lee, Cornelius Weber, Stefan Wermter. Internally Rewarded Reinforcement Learning. Proceedings of the International Conference on Machine Learning (ICML), pp. 20556-20574, Honolulu, USA, 2023.

- Mengdi Li, Cornelius Weber, Matthias Kerzel, Jae Hee Lee, Zheni Zeng, Zhiyuan Liu, Stefan Wermter. Robotic Occlusion Reasoning for Efficient Object Existence Prediction. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2686-2692, Prague, Czech Republic, 2021.

- Mengdi Li, Cornelius Weber, and Stefan Wermter. Neural Networks for Detecting Irrelevant Questions during Visual Question Answering. Proceedings of the International Conference on Artificial Neural Networks (ICANN), pp. 786–797, Bratislava, Slovakia, 2020.

Some of the ideas introduced in this thesis also contributed to the publications in other domains apart from the realm of active vision of embodied agents.

- Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, Stefan Wermter. Enhancing Zero-Shot Chain-of-Thought Reasoning in Large Language Models Through Logic. Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING), Torino, Italia, 2024.

- Wenhao Lu, Xufeng Zhao, Thilo Fryen, Jae Hee Lee, Mengdi Li, Sven Magg, Stefan Wermter. Causal State Distillation for Explainable Reinforcement Learning. Proceedings of the Conference on Causal Learning and Reasoning (CLeaR), Los Angeles, USA, 2024.

- Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, Stefan Wermter. Accelerating Reinforcement Learning of Robotic Manipulations via Feedback from Large Language Models. Workshop at the Conference on Robot Learning (CoRL), Atlanta, USA, 2023.

- Wenhao Lu, Xufeng Zhao, Sven Magg, Martin Gromniak, Mengdi Li, Stefan Wermter. A Closer Look at Reward Decomposition for High-Level Robotic Explanations. Proceedings of the IEEE International Conference on Development and Learning (ICDL), pp. 429-436, Macau, China, 2023.

- Xufeng Zhao, Mengdi Li, Cornelius Weber, Burhan Hafez, Stefan Wermter. Chat with the Environment: Interactive Multimodal Perception using Large Language Models. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3590-3596, Detroit, USA, 2023.

- Yuan Yao, Tianyu Yu, Ao Zhang, Mengdi Li, Ruobing Xie, Cornelius Weber, Zhiyuan Liu, Hai-Tao Zheng, Stefan Wermter, Tat-Seng Chua, Maosong Sun. Visually Grounded Commonsense Knowledge Acquisition. Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), pp. 6583-6592, Virtual Events, 2022.

- Xiaowen Sun, Cornelius Weber, Matthias Kerzel, Tom Weber, Mengdi Li, Stefan Wermter. Learning Visually Grounded Human-Robot Dialog in a Hybrid Neural Architecture. Proceedings of the International Conference on Artificial Neural Networks (ICANN), pp. 258-269, Bristol, United Kingdom, 2022.

- Jae Hee Lee, Yuan Yao, Ozan Özdemir, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter. Spatial Relation Learning in Complementary Scenarios with Deep Neural Networks. Frontiers in Neurorobotics, 16, p. 844753, 2022.

- Yuan Yao, Ao Zhang, Xu Han, Mengdi Li, Cornelius Weber, Zhiyuan Liu, Stefan Wermter, Maosong Sun. Visual Distant Supervision for Scene Graph Generation. Proceedings of the International Conference on Computer Vision (ICCV), pp. 15816-15826, Virtual Events, 2021.

Appendix C

Acknowledgements

As I reach the end of my doctoral research journey, I would like to express my deepest gratitude to those who have accompanied me throughout. First and foremost, I want to express my sincere thanks to my supervisor Prof. Dr. Stefan Wermter. Your constant support and thoughtful guidance have been essential to the success of this research. I deeply appreciate your patience, encouragement, and trust in my abilities, as well as the effort you have dedicated to reviewing my work and providing valuable feedback.

I would also like to thank Dr. Cornelius Weber for guiding me throughout my research projects. Your enthusiasm for research, dedication to mentoring young researchers, and attitude toward life will have a lifelong impact on me. Thanks to Katja Kösters for your unfailing kindness and generosity in offering assistance. Thanks to Dr. Jae Hee Lee for the invaluable lessons I learned during our collaborations, such as scientific writing and critical thinking.

I would also like to thank my colleagues from the WTM group. We had numerous interesting talks during the lunches and meetings. Special thanks to Dr. Leyuan Qu, Dr. Di Fu, Xufeng Zhao, Huajian Fang, Xiaowen Sun, Wenhao Lu, Kun Chu, Jianzhi Lyu, Mostafa Kotb, Hassan Ali, Fares Abawi, and Dr. Dennis Becker for your companionship. I am certain I will cherish the memories of our time together in Hamburg. Thanks to Erik Strahl, Dr. Matthias Kerzel, Dr. Burhan Hafez, Dr. Johannes Twiefel, and Dr. Philipp Allgeuer for your supportive assistance when I encountered difficulties during my research.

Finally, I am profoundly grateful to my family for their unwavering love, endless encouragement, and constant support. To my parents, what you taught me is the foundation for what I achieved today. To my brother, Mengnan, your support means a lot to me. To my love, Chen, I could not make it without you.

I would like to acknowledge that this work was supported by the China Scholarship Council (CSC) and the German Research Foundation DFG under project CML (TRR 169).

Bibliography

Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, and et al. Do as I can, not as I say: Grounding language in robotic affordances. In *Proceedings of The Conference on Robot Learning (CoRL)*, 2022.

John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1:333–356, 1988.

Ankesh Anand, Eugene Belilovsky, Kyle Kastner, Hugo Larochelle, and Aaron Courville. Blindfold baselines for embodied QA. In *Advances in Neural Information Processing Systems (NeurIPS) Workshops*, 2018.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. SPICE: Semantic propositional image caption evaluation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 382–398, 2016.

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, 2018b.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3674–3683, 2018c.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2425–2433, 2015.

Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. Multiple object recognition with visual attention. In *International Conference on Learning Representations (ICLR)*, 2015.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv: 2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv: 2212.08073*, 2022b.

Ruzena Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

David Barber and Felix Agakov. Information maximization in noisy channels: A variational approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 201–208, 2003.

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning (ICML)*, volume 80, pages 530–539, 2018.

Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics*, pages 679–684, 1957.

Hedi Ben-Younes, Remi Cadene, Nicolas Thome, and Matthieu Cord. BLOCK: Bilinear superdiagonal fusion for visual question answering and visual relationship detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8102–8109, 2019.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pages 41–48, 2009.

Serena Booth, W. Bradley Knox, Julie Shah, Scott Niekum, Peter Stone, and Alessandro Allievi. The perils of trial-and-error reward design: Misdesign through overfitting and invalid task specifications. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5920–5929, 2023.

Remi Cadene, Hedi Ben-Younes, Matthieu Cord, and Nicolas Thome. MUREL: Multimodal relational reasoning for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1989–1998, 2019.

Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set. *IEEE Robotics and Automation Magazine*, 22(3):36–52, 2015.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research (TMLR)*, 2023.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning (ICML)*, pages 1430–1440, 2021.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2819–2826, 2018.

Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4247–4258, 2020.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Danny Driess, Pete Florence, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. *arXiv preprint arXiv:2401.12168*, 2024.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 15084–15097, 2021.

Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active Vision in Robotic Systems: A Survey of Recent Developments. *The International Journal of Robotics Research (IJRR)*, 30(11):1343–1377, 2011.

Weizhe Chen, Roni Khardon, and Lantao Liu. AK: Attentive kernel for information gathering. In *Proceedings of Robotics: Science and Systems (RSS)*, 2022.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4299–4307, 2017.

Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, and Stefan Wermter. Accelerating reinforcement learning of robotic manipulations via feedback from large language models. In *Proceedings of The Conference on Robot Learning (CoRL) Workshops*, 2023.

Imre Csiszár. A class of measures of informativity of observation channels. *Periodica Mathematica Hungarica*, 2:191–213, 1972.

Charles Darwin. *On the Origin of Species by Means of Natural Selection*. Murray, London, 1859.

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.

Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–10, 2018.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

Yuhong Deng, Di Guo, Xiaofeng Guo, Naifu Zhang, Huaping Liu, and Fuchun Sun. MQA: Answering the question via robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2021.

Marco Dorigo and Marco Colombetti. Robot shaping: developing autonomous agents through learning. *Artificial Intelligence*, 71(2):321–370, 1994. ISSN 0004-3702.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, and et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, pages 8469–8488, 2023.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning (ICML)*, pages 8657–8677, 2023.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied AI: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

Onno Eberhard, Jakob Hollenstein, Cristina Pinneri, and Georg Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Gamaleldin F. Elsayed, Simon Kornblith, and Quoc V. Le. Saccader: Improving accuracy of hard attention models for vision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 700–712, 2019.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations (ICLR)*, 2019.

Amedeo Roberto Esposito, Michael Gastpar, and Ibrahim Issa. Robust generalization via f-mutual information. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2723–2728, 2020.

Tom Everitt, Victoria Krakovna, Laurent Orseau, and Shane Legg. Reinforcement learning with a corrupted reward channel. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4705–4713, 2017.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations (ICLR)*, 2019.

Patrick Foo, William H Warren, Andrew Duchon, and Michael J Tarr. Do humans integrate routes into a cognitive map? map-versus landmark-based navigation of novel shortcuts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(2):195, 2005.

Di Fu, Cornelius Weber, Guochun Yang, Matthias Kerzel, Weizhi Nan, Pablo Barros, Haiyan Wu, Xun Liu, and Stefan Wermter. What can computational models learn from human selective attention? A review from an audiovisual unimodal and crossmodal perspective. *Frontiers in Integrative Neuroscience*, 14:10, 2020.

Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4438–4446, 2017.

Jaime Roquero Gimenez and James Y. Zou. A unified f-divergence framework generalizing VAE and GAN. *arXiv preprint arXiv:2205.05214*, 2022.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

Leonardo Gomes, Olga Regina Pereira Bellon, and Luciano Silva. 3D reconstruction methods for digital preservation of cultural heritage: A survey. *Pattern Recognition Letters*, 50:3–14, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. IQA: visual question answering in interactive environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4089–4098, 2018.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6904–6913, 2017.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 5:1471–1530, 2004.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. In *International Conference on Learning Representations (ICLR) Workshops*, 2017.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations (ICLR)*, 2022.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. VizWiz Grand Challenge: Answering visual questions from blind people. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3608–3617, 2018.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2455–2467, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv: 2301.04104*, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

Martin Heidegger. *The Basic Problems of Phenomenology*, volume 478. Indiana University Press, 1988.

Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, Marcus Wainwright, Chris Apps, Demis Hassabis, and Phil Blunsom. Grounded language learning in a simulated 3D world. *arXiv preprint arXiv: 1706.06551*, 2017.

Felix Hill, Andrew Lampinen, Rosalia Schneider, Stephen Clark, Matthew Botvinick, James L. McClelland, and Adam Santoro. Environmental drivers of systematicity and generalization in a situated agent. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Felix Hill, Olivier Tieleman, Tamara von Glehn, Nathaniel Wong, Hamza Merzic, and Stephen Clark. Grounded language learning fast and slow. In *International Conference on Learning Representations (ICLR)*, 2021.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573, 2016.

Hengyuan Hu and Dorsa Sadigh. Language instructed reinforcement learning for human-AI coordination. In *International Conference on Machine Learning (ICML)*, pages 13584–13598, 2023.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615, 2023.

Sandy H. Huang, Nicolas Papernot, Ian J. Goodfellow, Yan Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. In *International Conference on Learning Representations (ICLR)*, 2017.

Drew A Hudson and Christopher D Manning. Compositional attention networks for machine reasoning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Nikolai Ilinykh, Yasmeen Emampoor, and Simon Dobnik. Look and answer the question: On the role of vision in embodied question answering. In *Proceedings of the International Conference on Natural Language Generation (INLG)*, pages 236–245, 2022.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3D reconstruction. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484. IEEE, 2016.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0. 1: The winning entry to the VQA challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, 2017.

Kushal Kafle and Christopher Kanan. An analysis of visual question answering algorithms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1965–1973, 2017.

Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Justin B. Kinney and Gurinder S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences (PNAS)*, 111(9):3354–3359, 2014.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3294–3302, 2015.

Martin Klissarov, Pierluca D'Oro, Shagun Sodhani, Roberta Raileanu, Pierre-Luc Bacon, Pascal Vincent, Amy Zhang, and Mikael Henaff. Motif: Intrinsic motivation from artificial intelligence feedback. In *International Conference on Learning Representations (ICLR)*, 2023.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

Yves Kompis, Luca Bartolomei, Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Informed sampling exploration path planner for 3D reconstruction of large scenes. *IEEE Robotics and Automation Letters*, 6(4):7893–7900, 2021.

Vijay R. Konda and John N. Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1008–1014, 1999.

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.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1106–1114, 2012.

Alexander Kuhnle and Ann Copestake. ShapeWorld - A new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*, 2017.

Obin Kwon, Jeongho Park, and Songhwai Oh. Renderable neural radiance map for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9099–9108, 2023.

Egor Lakomkin, Mohammad Ali Zamani, Cornelius Weber, Sven Magg, and Stefan Wermter. EmoRL: Continuous acoustic emotion classification using deep reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4445–4450, 2018.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. RLAIF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv preprint arXiv: 2309.00267*, 2023.

Soomin Lee, Le Chen, Jiahao Wang, Alexander Liniger, Suryansh Kumar, and Fisher Yu. Uncertainty guided policy for active robotic 3D reconstruction using neural radiance fields. *IEEE Robotics and Automation Letters*, 7(4):12070–12077, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. Igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Proceedings of The Conference on Robot Learning (CoRL)*, pages 455–465, 2022.

Mengdi Li, Cornelius Weber, and Stefan Wermter. Neural networks for detecting irrelevant questions during visual question answering. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 786–797, 2020.

Mengdi Li, Cornelius Weber, Matthias Kerzel, Jae Hee Lee, Zheni Zeng, Zhiyuan Liu, and Stefan Wermter. Robotic occlusion reasoning for efficient object existence prediction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2686–2692, 2021.

Mengdi Li, Xufeng Zhao, Jae Hee Lee, Cornelius Weber, and Stefan Wermter. Internally rewarded reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 20556–20574, 2023.

Zhichao Li, Yi Yang, Xiao Liu, Feng Zhou, Shilei Wen, and Wei Xu. Dynamic computational time for visual attention. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2017.

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 *Proceedings of European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

Xiao Liu, Tian Xia, Jiang Wang, Yi Yang, Feng Zhou, and Yuanqing Lin. Fully convolutional attention networks for fine-grained recognition. *arXiv preprint arXiv:1603.06765*, 2016.

Ali Lotfi-Rezaabad and Sriram Vishwanath. Learning representations by maximizing mutual information in variational autoencoders. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2729–2734, 2020.

Yuliia Lysa, Cornelius Weber, Dennis Becker, and Stefan Wermter. Word-by-word generation of visual dialog using reinforcement learning. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 123–135, 2022.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Aroma Mahendru, Viraj Prabhu, Akrit Mohapatra, Dhruv Batra, and Stefan Lee. The Promise of Premise: Harnessing question premises in visual question answering. *arXiv preprint arXiv:1705.00601*, 2017.

Maurice Merleau-Ponty and Colin Smith. *Phenomenology of Perception*, volume 26. Routledge London, 1962.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 405–421, 2020.

Sarthak Mittal, Korbinian Abstreiter, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion based representation learning. In *International Conference on Machine Learning (ICML)*, pages 24963–24982, 2023.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS) Workshops*, 2013.

Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2204–2212, 2014.

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 *International Conference on Machine Learning (ICML)*, pages 1928–1937, 2016.

Raghav Nagpal, Achyuthan Unni Krishnan, and Hanshen Yu. Reward engineering for object pick and place training. *arXiv preprint arXiv:2001.03792*, 2020.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

Andrew Y. Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In Pat Langley, editor, *International Conference on Machine Learning (ICML)*, pages 663–670, 2000.

Stuart Russell; Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Higher Education, 4 edition, 2019.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 271–279, 2016.

James Olds and Peter Milner. Positive reinforcement produced by electrical stimulation of septal area and other regions of rat brain. *Journal of Comparative and Physiological Psychology*, 47(6):419, 1954.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 27730–27744, 2022.

Xuran Pan, Zihang Lai, Shiji Song, and Gao Huang. Activenerf: Learning where to see with uncertainty estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 230–246, 2022.

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. Language model self-improvement by reinforcement learning contemplation. In *International Conference on Learning Representations (ICLR)*, 2023.

Seohong Park, Jongwook Choi, Jaekyeom Kim, Honglak Lee, and Gunhee Kim. Lipschitz-constrained unsupervised skill discovery. In *International Conference on Learning Representations (ICLR)*, 2022.

Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. Controllability-aware unsupervised skill discovery. In *International Conference on Machine Learning (ICML)*, 2023.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, pages 2778–2787, 2017.

Jing Peng and Ronald J. Williams. Incremental multi-step Q-learning. *Machine Learning*, 22(1-3):283–290, 1996.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 53728–53741, 2023.

Marie-Julie Rakotosaona, Fabian Manhardt, Diego Martin Arroyo, Michael Niemeyer, Abhijit Kundu, and Federico Tombari. Nerfmeshing: Distilling neural radiance fields into geometrically-accurate 3d meshes. In *International Conference on 3D Vision (3DV)*, 2023.

Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. PONI: Potential functions for ObjectGoal navigation with interaction-free learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18868–18878, 2022.

Yunlong Ran, Jing Zeng, Shibo He, Jiming Chen, Lincheng Li, Yingfeng Chen, Gimhee Lee, and Qi Ye. NeurAR: Neural uncertainty for autonomous 3D reconstruction with implicit neural representations. *IEEE Robotics and Automation Letters*, 8(2):1125–1132, 2023.

Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *International Conference on Machine Learning (ICML)*, volume 98, pages 463–471, 1998.

Samrudhdhi B. Rangrej, Chetan L. Srinidhi, and James J. Clark. Consistency driven sequential transformers attention model for partially observable scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2508–2517, 2022.

Arijit Ray, Gordon Christie, Mohit Bansal, Dhruv Batra, and Devi Parikh. Question relevance in VQA: Identifying non-visual and false-premise questions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 919–924, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015.

E. Rohmer, S. P. N. Singh, and M. Freese. V-REP: a versatile and scalable robot simulation framework. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1321–1326, 2013.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4035–4045, 2018.

Yao Rong, Wenjia Xu, Zeynep Akata, and Enkelejda Kasneci. Human attention in fine-grained classification. In *Proceedings of the British Machine Vision Conference (BMVC)*, page 150, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

Gavin A Rummery and Mahesan Niranjan. *On-Line Q-learning Using Connectionist Systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust Region Policy Optimization. In *International Conference on Machine Learning (ICML)*, pages 1889–1897, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2015b.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Wolfram Schultz. Neuronal reward and decision signals: From theories to data. *Physiological Reviews*, 95(3):853–951, 2015.

Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. igibson 1.0: A simulation environment for interactive tasks in large realistic scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7520–7527, 2021.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Satinder Singh, Andrew G. Barto, and Nuttapong Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1281–1288, 2004.

Søren Kaae Sønderby, Casper Kaae Sønderby, Lars Maaløe, and Ole Winther. Recurrent spatial transformer networks. *arXiv preprint arXiv:1509.05329*, 2015.

DJ Strouse, Kate Baumli, David Warde-Farley, Vlad Mnih, and Steven Hansen. Learning more skills through optimistic exploration. In *International Conference on Learning Representations (ICLR)*, 2022.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction.* Adaptive computation and machine learning. MIT Press, 1998.

Sinan Tan, Huaping Liu, Di Guo, Xinyu Zhang, and Fuchun Sun. Towards embodied scene description. In *Proceedings of Robotics: Science and Systems (RSS)*, 2020.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and at al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv: 2307.09288*, 2023.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2094–2100, 2016.

Francisco J Varela, Evan Thompson, and Eleanor Rosch. *The Embodied Mind, Revised Edition: Cognitive Science and Human Experience.* MIT press, 2017.

Chaoqun Wang, Jiyu Cheng, Jiankun Wang, Xintong Li, and Max Q-H Meng. Efficient object search with belief road map using mobile robot. *IEEE Robotics and Automation Letters*, 3(4):3081–3088, 2018.

Jingkang Wang, Yang Liu, and Bo Li. Reinforcement learning with perturbed rewards. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6202–6209, 2020.

Qi Wang, Shaoteng Liu, Jocelyn Chanussot, and Xuelong Li. Scene classification with recurrent attention of VHR remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):1155–1167, 2019.

Ranxiao Frances Wang and Elizabeth S Spelke. Human spatial representation: Insights from animals. *Trends in cognitive sciences*, 6(9):376–382, 2002.

Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge United Kingdom, 1989.

Jiaheng Wei and Yang Liu. When optimizing f-divergence is robust with label noise. In *Conference on Learning Representations (ICLR)*, 2021.

Spencer Whitehead, Suzanne Petryk, Vedaad Shakib, Joseph Gonzalez, Trevor Darrell, Anna Rohrbach, and Marcus Rohrbach. Reliable visual question answering: Abstain rather than answer incorrectly. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 148–166, 2022.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. DD-PPO: Learning near-perfect PointGoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E. Gonzalez, and Trevor Darrell. See, say, and segment: Teaching lmms to overcome false premises. *arXiv preprint arXiv:2312.08366*, 2023.

Yi Wu, Yuxin Wu, Aviv Tamar, Stuart Russell, Georgia Gkioxari, and Yuandong Tian. Bayesian relational memory for semantic visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2769–2779, 2019.

Dongyu Yan, Jianheng Liu, Fengyu Quan, Haoyao Chen, and Mengmeng Fu. Active implicit object reconstruction using uncertainty-guided next-best-view optimization. *IEEE Robotics and Automation Letters*, 8(10):6395–6402, 2023.

Jianwei Yang, Zhile Ren, Mingze Xu, Xinlei Chen, David J. Crandall, Devi Parikh, and Dhruv Batra. Embodied amodal recognition: Learning to move to perceive objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2040–2050, 2019a.

Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual semantic navigation using scene priors. In *International Conference on Learning Representations (ICLR)*, 2019b.

Xin Yang, Yuanbo Wang, Yaru Wang, Baocai Yin, Qiang Zhang, Xiaopeng Wei, and Hongbo Fu. Active object reconstruction using a guided view planner. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4965–4971, 2018.

Xin Ye, Zhe Lin, Haoxiang Li, Shibin Zheng, and Yezhou Yang. Active object perceiver: Recognition-guided policy learning for object searching on mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6857–6863, 2018.

Adams Wei Yu, Hongrae Lee, and Quoc V. Le. Learning to skim text. In Regina Barzilay and Min-Yen Kan, editors, *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1880–1890, 2017.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. Pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021.

Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, Brian Ichter, Ted Xiao, Peng Xu, Andy Zeng, Tingnan Zhang, Nicolas Heess, Dorsa Sadigh, Jie Tan, Yuval Tassa, and Fei Xia. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv: 2306.08647*, 2023.

Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling ImageNet: From single to multi-labels, from global to localized labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Rowan Zellers, Ari Holtzman, Matthew E. Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. PIGLeT: Language grounding through neuro-symbolic interaction in a 3D world. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2040–2050, 2021.

Xin Zhang, Yanhua Li, Ziming Zhang, and Zhi-Li Zhang. f-GAIL: Learning f-divergence for generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12805–12815, 2020.

Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. Chat with the environment: Interactive multimodal perception using large language models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3590–3596, 2023.

Kaiwen Zhou, Kaizhi Zheng, Connor Pryor, Yilin Shen, Hongxia Jin, Lise Getoor, and Xin Eric Wang. ESC: Exploration with soft commonsense constraints for zero-shot object navigation. In *International Conference on Machine Learning (ICML)*, pages 42829–42842, 2023.

Xiaohui Zhou, Ke Xie, Kai Huang, Yilin Liu, Yang Zhou, Minglun Gong, and Hui Huang. Offsite aerial path planning for efficient urban scene reconstruction. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. Analyzing and mitigating object hallucination in large vision-language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, 2017.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, pages 2165–2183, 2023.

Erklärung der Urheberschaft

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

Hamburg, 05.06.2024 Mengdi Li

Ort, Datum Unterschrift

Erklärung zur Veröffentlichung

Ich erkläre mein Einverständnis mit der Einstellung dieser Dissertation in den Bestand der Bibliothek.

Hamburg, 05.06.2024 Mengdi Li

Ort, Datum Unterschrift