



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Environment Exploration and Autonomous Adaptation in Embodied Agents

Dissertation

submitted to the University of Hamburg
with the aim of achieving a doctoral degree at

Department of Informatics
MIN-Faculty
University of Hamburg

Xufeng Zhao
Hamburg, 2025

Submission Date

Monday 23rd June, 2025

Oral Defense Date

Wednesday 8th October, 2025

Dissertation Committee

- ▶ Prof. Dr. Sören Laue (chair)
Dept. of Computer Science, University of Hamburg, Germany
- ▶ Prof. Dr. Jianwei Zhang (deputy chair, reviewer)
Dept. of Computer Science, University of Hamburg, Germany
- ▶ Prof. Dr. Stefan Wermter (advisor, reviewer)
Dept. of Computer Science, University of Hamburg, Germany

ABSTRACT

With the rapid advancement of Artificial Intelligence (AI), autonomous systems have gained increasing attention due to their growing potential across both virtual and real-world applications. Developing embodied agents that can follow human instructions requires not only semantic understanding but also efficient policy learning. To achieve further autonomy, an agent must explore its environment and adapt its capabilities beyond the initial design, which motivates research into world modeling and robotic self-determination.

This thesis begins by presenting a unified conceptual foundation for autonomous embodiment, followed by contributions that integrate multiple aspects of this foundation. First, the thesis introduces multimodal cues as intrinsic motivation to enable reinforcement learning agents to engage in self-determined exploration and representation learning, warming up their policies beyond immediate task demands. Second, the thesis proposes a decision-level interactive perception approach based on Large Language Models (LLMs), enabling agents to semantically reason about multimodal inputs for improved exploration and environmental understanding. Third, to strengthen the reasoning abilities of LLMs, the thesis explores logic-guided inference exploration to enhance performance on complex reasoning tasks without requiring additional fine-tuning. Fourth, the thesis addresses long-term embodied autonomy by enabling agents to reason about affordances in their environment and discover novel skills through self-determined policy learning. Finally, the thesis concludes with collaborative research on object-centric planning, bimanual coordination, and explainability in embodied systems, further extending and contextualizing the contributions within broader research on embodied intelligence.

ZUSAMMENFASSUNG

Mit dem rapiden Fortschritt der Künstlichen Intelligenz (KI) haben autonome Systeme zunehmende Aufmerksamkeit erlangt, bedingt durch ihr wachsendes Potenzial in virtuellen und realen Anwendungen. Die Entwicklung von verkörperten Agenten, die menschlichen Anweisungen folgen können, erfordert nicht nur semantisches Verständnis, sondern auch effizientes Erlernen der Aktionstrategie, dem sogenannten Policy-Learning. Um weitergehende Autonomie zu erreichen, muss ein Agent seine Umgebung erkunden und seine Fähigkeiten über seine ursprünglich gegebenen hinaus entwickeln, was die Modellierung der Welt und robotische Selbstbestimmung motiviert.

Diese Dissertation beginnt mit der Vorstellung einer vereinheitlichten konzeptionellen Grundlage für autonome Verkörperung, gefolgt von Beiträgen, die mehrere Aspekte mit dieser Grundlage untersuchen. Zunächst führt die Dissertation multimodale sensorische Reize als intrinsische Motivation für Agenten ein, die mittels Verstärkungslernen trainiert werden. Damit können sie selbstbestimmte Exploration und Repräsentationslernen durchführen, indem sie ihre Aktionsstrategien über unmittelbare Aufgabenanforderungen hinaus optimieren. Zweitens schlägt die Dissertation eine interaktive Wahrnehmung mittels großen Sprachmodellen (Large Language Models, LLMs) vor, die Agenten befähigt, semantisch über multimodale Eingaben zu argumentieren, um Exploration und Umweltverständnis zu verbessern. Drittens untersucht die Dissertation zur Stärkung der Argumentationsfähigkeiten von LLMs logikgesteuerte Inferenzexploration, um die Leistung bei komplexen Argumentationsaufgaben zu verbessern, ohne zusätzliches Feintuning zu benötigen. Viertens adressiert die Dissertation langfristige verkörperte Autonomie, indem sie Agenten befähigt, über Möglichkeiten in ihrer Umgebung nachzudenken und neue Fähigkeiten durch selbstbestimmtes Policy-Learning zu entdecken. Die Dissertation schließt mit kollaborativen Arbeiten zu objektorientierter Planung, bimanualer Koordination und Erklärbarkeit in verkörperten Systemen ab, die die Beiträge innerhalb des weiten Forschungsfeldes der verkörperten Intelligenz weiter ausbauen und kontextualisieren.

ACKNOWLEDGEMENTS

Completing this PhD has been a long journey made possible by the support, encouragement, and kindness of many people. I am deeply grateful to my colleagues, friends, and family for walking alongside me during both the challenges and the breakthroughs. Their presence has meant a great deal throughout these years.

First and foremost, I would like to express my sincere and heartfelt gratitude to my advisor, **Prof. Dr. Stefan Wermter**. His mentorship has shaped the way I think, work, and grow as a researcher. His passion for research and teaching, along with his patience and dedication, deeply inspired me. He taught me the importance of reflective thinking, always encouraging me to examine ideas from multiple perspectives and to approach research with thoughtful precision and profound clarity. I am especially grateful for his guidance and detailed feedback at every stage, which helped me refine my work and move forward with confidence. I am especially thankful for the many opportunities he provided for me to explore beyond the boundaries of our group, including his full support for attending international conferences and engaging with the wider research community. Through his leadership, I gained a valuable understanding of how a collaborative research group can be developed and guided with care and vision. It has been an honor to learn under his guidance.

I am equally grateful to **Dr. Cornelius Weber** for his invaluable advice during our meetings. He not only shared inspiring ideas that continually pushed me to think outside the box but also taught me the importance of maintaining a positive attitude toward life. His detailed comments, both on my writing and on broader topics, were immensely helpful. Beyond research, he provided insightful advice on collaboration, communication, and teaching, which has enriched my approach to working with colleagues and mentoring students. His support has been essential in shaping my academic journey and personal growth. I would also like to thank **Katja Kösters** for her invaluable support, efficiency, *e.g.* in handling administrative tasks, professionalism, and kindness.

My deep thanks go to my colleagues in the group, whose support, collaboration, and friendship have been invaluable throughout my PhD journey. **Dr. Mengdi Li** has been my invaluable research partner and a significant presence in my life. Throughout our time together in the group, we engaged in numerous profound discussions, ranging from professional endeavors to personal insights. Our collaboration has been incredibly enriching and enjoyable, and I eagerly anticipate our continued work together in the future. **Dr. Jae Hee Lee** has been a great source of inspiration. We share common interests from research to self-development, and our conversations have been rewarding. I learned much from him, especially in the areas of high-level thinking and presentation skills. **Kun Chu** and **Wenhao Lu**, thank you for the many meaningful moments we have shared, both in our work and throughout daily life. I truly cherished the time we spent together. They had a profound impact on my personal journey, introducing me to new friends and experiences, and encouraging me to take part in activities I might not have explored on my own. **Huajian Fang**,

his devotion to his work and life has been a true inspiration. Our interactions have always been a source of joy, and I have learned so much from his approach to life. Special thanks to **Xiaowen Sun, Mostafa Kotb, Josua Spisak, Dr. Di Fu, and Dr. Burhan Hafez**. Their friendship and research discussions brought many fun and new perspectives to my experience in the group. Our conversations about research and life were always enriching. I would also like to extend my gratitude to my other colleagues, past and present, for their support and camaraderie: **Dr. Matthias Kerzel, Dr. Philipp Allgeuer, Marie Bauer, Jan-Gerrit Habekost, Dr. Dennis Becker, Kyra Ahrens, Dipl.-Ing. Erik Strahl, Dr. Fares Abawi, Dr. Leyuan Qu, Burak Can Kaplan, Sergio Lanza, Dr. Ozan Özdemir, Connor Gäde, Hassan Ali, Dr. Johannes Twiefel, Dr. Hugo Cesar de Castro Carneiro, Theresa Pekarek-Rosin, Julia Gachot, and Martin Gromniak**. I am grateful for all the shared experiences.

I am especially thankful to my friends whose presence and support have been deeply meaningful throughout this journey. **Xiaoke Xin** helped me tremendously during the writing of my initial research proposal, which turned out to be the first step of my PhD journey. Some people have simply shown up in my life and made it better in their own unique ways, and I am sincerely thankful for the presence of **Jianzhi Lyu, Guanlue Li**, who made my life more delightful. This journey has not only been a research endeavor, but also a profound life experience—one through which I have learned how to balance rationality, emotional impulse, and responsibility.

Above all, I owe a great deal to my family, whose unwavering support has carried me through every stage of this journey. Their unconditional love and belief in me have been my solid foundation and enduring source of strength.

Lastly, I would like to acknowledge the generous support of the China Scholarship Council (CSC), the University of Hamburg, and the German Research Foundation (DFG) under project CML (TRR 169), whose generous support made this research possible. My sincere thanks also go to **Prof. Jianwei Zhang** and **Prof. Sören Laue** for their time and valuable feedback, which helped shape and strengthen this work.

Human beings are being shaped by their environments, and the people we encounter along the way play a significant role in this process. Though many can only accompany us for part of the journey, their presence is a gift that leaves a lasting imprint. I am thankful for every moment shared, and I carry the influence of these encounters with me as I move forward.

Hamburg, June 2025
Xufeng Zhao

CONTENTS

ABSTRACT	III
ZUSAMMENFASSUNG	IV
ACKNOWLEDGEMENTS	V
CONTENTS	VII
List of Figures	xi
List of Tables	xiii
List of Algorithms	xiv
I FOUNDATIONS AND CONTEXT	1
1 INTRODUCTION	2
1.1 Motivation	2
1.1.1 Agent, Embodiment, Robot and Autonomy	2
1.1.2 Problem Statement	2
1.1.3 Challenges	3
1.1.4 Research Scope	4
1.2 Research Objectives and Contributions	5
1.3 Thesis Organization	7
2 BACKGROUND AND RELATED WORK	9
2.1 Robotic Autonomy	9
2.1.1 Environment Exploration	9
2.1.2 Autonomous Adaptation	15
2.2 Robot Setting	18
2.2.1 NICOL	18
2.2.2 Franka Emika Panda	19
2.3 Simulation Setting	20
2.3.1 ThreeDWorld	20
2.3.2 CoppeliaSim	21
2.3.3 Isaac Sim	21
2.3.4 Other Simulators	22
2.4 Perspectives from the Literature	22
2.4.1 Early Conceptualizations	23
2.4.2 Evolving Perspectives in the Era of LLMs	24

3	CONCEPTUAL FOUNDATIONS	25
3.1	World Model	25
3.1.1	Sensing, Perception and Multimodal Fusion	26
3.1.2	Modeling and Representations	29
3.1.3	Utilization of World Models	34
3.2	Semantics	35
3.2.1	Abstract Concepts	35
3.2.2	Integration: Symbolic Grounding	37
3.3	Policy	38
3.3.1	Policy Function, Option, and Skill	38
3.3.2	Planning and Learning	40
3.3.3	Integration: Planning and Learning with Foundation Models	44
3.4	Self-Determination	47
3.4.1	Intrinsic Motivation	48
3.4.2	Self-Regulation	50
3.4.3	Integrations	52
II	CORE CONTRIBUTIONS	54
4	SOUND GUIDES REPRESENTATIONS AND EXPLORATIONS	55
4.1	Introduction	56
4.2	Related Work	57
4.2.1	Self-supervised Representation Learning	58
4.2.2	Active Exploration	58
4.3	Method: ISCM	59
4.3.1	Visual Representation Learning	60
4.3.2	Intrinsic Visual-Auditory Reward	62
4.3.3	Representation and Policy Learning	62
4.4	Experiments and Results	62
4.4.1	Experimental Setup	63
4.4.2	Implementations	63
4.4.3	Evaluation	65
4.4.4	Results and Discussion	65
4.5	Conclusion	68
4.6	Limitations and Future Work	68
5	INTERACTIVE MULTIMODAL PERCEPTION USING LARGE LANGUAGE MODELS	69
5.1	Introduction	70
5.2	Related Work	72
5.2.1	Multimodal Learning and Robotic Information Gathering	72
5.2.2	Large Language Models in Robotic Planning	73

5.3	Method: Matcha agent	73
5.3.1	Architecture	73
5.3.2	Multimodal Perception and Execution Policy	74
5.3.3	Prompt Engineering	75
5.4	Experiments and Results	76
5.4.1	Experimental Setup	77
5.4.2	Results	78
5.4.3	Case Studies	80
5.4.4	Discussion	81
5.5	Conclusion	82
5.6	Limitations and Future Work	82
6	ENHANCING REASONING VIA LOGIC-GUIDED INFERENCE SCALING	83
6.1	Introduction	85
6.2	Related Work	85
6.2.1	Chain-of-Thought Prompting	86
6.2.2	Variational Prompting	86
6.2.3	Neurosymbolic Models	88
6.3	Method: LoT	88
6.3.1	Reductio ad Absurdum	88
6.3.2	LoT Prompting	89
6.3.3	Chain Growth	91
6.4	Experiments and Results	91
6.4.1	Experimental Setup	92
6.4.2	Analysis	93
6.5	Conclusion	97
6.6	Limitations and Future Work	97
7	AGENTIC SKILL DISCOVERY	99
7.1	Introduction	100
7.2	Related Work	101
7.2.1	Skill Discovery	102
7.2.2	Code LLM Control	103
7.3	Method: ASD	103
7.3.1	Iterative Task Proposal and Skill Collection	104
7.3.2	Evolutionary Skill Learning with Fast and Slow Success Determination	105
7.3.3	On-demand Skill Learning with Quest Decomposition	109
7.4	Experiments and Results	110
7.4.1	Experimental Setup	110
7.4.2	Analysis	112
7.5	Conclusion	116
7.6	Limitations and Future Work	116

8	REWARD MODELING, EMBODIED PLANNING, AND EXPLAINABILITY	118
8.1	Reinforcement Learning with Derived Rewards	119
8.1.1	Reinforcement Learning with Inductive Rewards	119
8.1.2	Reinforcement Learning with Deductive Rewards	123
8.2	LLM-based Embodied Planning	124
8.2.1	Object State-Sensitive Agent	125
8.2.2	Bimanual Planning	126
8.3	Explainability in AI Agents	127
8.4	Conclusion	130
8.5	Limitations and Future Work	130
9	DISCUSSION, FUTURE DIRECTIONS, AND CONCLUSIONS	131
9.1	Discussion and Future Directions	131
9.2	Conclusions	135
	APPENDIX	137
A	PREDICTION ERROR AND GAUSSIAN MODELING	138
B	IMPACT SOUND SIMULATION	139
B.1	Physics-based Sound Simulation Theory	139
B.2	Sound Simulation Practice for Ceramic Objects	141
C	LOGICAL THOUGHTS (LoT) APPENDIX	144
C.1	ChatGPT Deduction Example with <i>vs</i> without Contraposition	144
C.2	Proof of Contraposition	144
C.3	Pseudo Codes for Cmps-LoT Prompting	145
C.4	Worsening and Improvement Rates	146
C.5	LoT Prompting Diagram in Detail	147
C.6	Prompts and Results for Cases	147
D	AGENTIC SKILL DISCOVERY (ASD) APPENDIX	158
D.1	Generated Functions Examples	158
D.2	Skill Learning Reports	161
D.3	Prompts	164
	BIBLIOGRAPHY	168
	LIST OF PUBLICATIONS	200
	NOTATION	202
	EIDESSTATTLICHE VERSICHERUNG	204
	ERKLÄRUNG ZUR VERÖFFENTLICHUNG	205

LIST OF FIGURES

1.1	Intelligent Agent Tetrahedron	4
2.1	Set illustration for ASLAM: localization, planning, mapping	10
2.2	Path planning illustration	11
2.3	Curiosity as intrinsic motivation to explore	14
2.4	Skill discovery illustration	17
2.5	NICOL's <i>head, arms</i> and <i>workspace</i>	18
2.6	NICOL in simulators and in real world	19
2.7	Panda robot in CoppeliaSim and Isaac Sim	19
3.1	Active perception	27
3.2	Contrastive Language-Image Pretraining (CLIP) training objective	36
3.3	Symbolic Grounding	37
3.4	Instructional control and planning	40
3.5	Reinforcement learning paradigm	41
3.6	Skill learning	45
3.7	Symbolic planning with PDDL	46
3.8	Agentic skill discovery	50
3.9	Model-based exploration, interactive scene graph and reasoning	53
4.1	<i>ManipulateSound</i> environments in ThreeDWorld multimodal simulator .	57
4.2	Overview of <u>I</u> ntrinsic <u>S</u> ound <u>C</u> uriosity <u>M</u> odule (ISCM) architecture . . .	60
4.3	Extrinsic reward accumulation in exploration phase	66
4.4	Episodic rewards in adaptation stage	67
4.5	Episodic rewards in adaptation stage with randomization	67
5.1	<u>M</u> ultimodal environment <u>c</u> hatting (Matcha) agent execution example . .	71
5.2	Overview of Matcha framework	74
5.3	A Matcha example with indistinct description	80
5.4	A Matcha example with distinct description	81
5.5	A Matcha example of failure	81
6.1	Overview of <u>L</u> ogical <u>T</u> houghts (LoT) method compared with Chain-of-Thought (CoT)	84
6.2	Diagram of the <i>think-verify-revision</i> loop in LoT	89
7.1	Example learned skill by <u>A</u> gentic <u>S</u> kill <u>D</u> iscovery (ASD) framework . . .	100
7.2	Overview of propose-learn-collect skill acquisition workflow in ASD . . .	104
7.3	Evolutionary search of reward functions for deterministic task	105
7.4	Skill learning and evolution in ASD	106
7.5	Skill stack and on-demand learning	109

7.6	Parallel simulation setup in Isaac Sim	111
7.7	Possible failure modes in evolution	115
8.1	Reinforcement Learning (RL) with derived rewards	119
8.2	Cold start in <u>I</u> nternally <u>R</u> ewarded <u>R</u> einforcement <u>L</u> earning (IRRL)	120
8.3	Hard attention with MNIST example	122
8.4	RL with deductive rewards	123
8.5	Lafite-RL learning paradigm	124
8.6	The <u>O</u> bject- <u>S</u> tate <u>S</u> ensitive <u>A</u> gent (OSSA) makes context-aware task plan- ning.	125
8.7	Bimanual coordination with spatiotemporal control types defined in <u>L</u> anguage-model-based <u>B</u> imanual <u>O</u> Rchestration (LABOR)	126
8.8	Bimanual manipulation with spatiotemporal coordination	126
8.9	LLM+MAP architecture	127
8.10	Explainable Q-Map	128
8.11	State disentanglement and reward decomposition	129
8.12	Mental modeling RL agents	129
B.1	Mass-spring-damper system	139
C.1	An example conversation where logic prompts help ChatGPT deduction	145
C.2	Detailed diagram of the LoT workflow with a real example on the AQuA dataset.	147
D.4	Task proposal prompts	164
D.5	Success function generation prompts	165
D.6	Reward function generation prompts	166
D.7	Code generation feedback prompts	167
D.8	Behavioral assessment prompts	167

LIST OF TABLES

3.1	Overview of core contributions and integrations into conceptual foundations	53
5.1	<u>M</u> ultimodal environment <u>chat</u> ting (Matcha) prompts	76
5.2	Descriptive properties of different materials used by Matcha.	79
5.3	Matcha task success rate with various LLMs.	80
6.1	Accuracy of <u>L</u> ogical <u>T</u> houghts (LoT) and Chain-of-Thought (CoT) over various datasets and language models	94
6.2	Step-wise revision frequency of LoT	95
6.3	Resultant reasoning steps with and without LoT applied	95
6.4	Ablation study of LoT variants	96
7.1	<u>A</u> gentic <u>S</u> kill <u>D</u> iscovery (ASD) task proposals	114
7.2	Quest completions with stacked ASD skills	116
C.1	Worsening rate ($\downarrow$) and improvement rate ($\uparrow$) when LoT is introduced	146
D.1	ASD learning details	162

LIST OF ALGORITHMS

1	Pseudo code for <u>I</u> ntrinsic <u>S</u> ound <u>C</u> uriosity <u>M</u> odule (ISCM)	64
2	Pseudo code for <u>L</u> ogical <u>T</u> houghts (LoT) reasoning	91
3	Agentic Skill Discovery (ASD) for open-skill learning	108
4	Agentic Skill Discovery (ASD) for quest completion	110
5	Pseudo code for Cmps-LoT reasoning	146

Part I

FOUNDATIONS AND CONTEXT

1.1 MOTIVATION

1.1.1 Agent, Embodiment, Robot and Autonomy

In establishing a common foundation for our discussion, it is essential to clarify several key concepts that will recur throughout this thesis. An *agent* is an entity that perceives its environment, processes information, and takes actions to achieve goals, characterized by perception, decision-making, and control. Building on this, the concept of *embodiment* emphasizes that cognitive processes are deeply rooted in the body's interactions with the world. An *embodied agent*, therefore, is one that possesses a body, either physical or virtual, enabling it to interact meaningfully with its environment. A *robot* is a specific form of embodied agent: a computational machine capable of performing physical actions, typically in the real world, based on its perception and internal decision-making processes. Within the context of this thesis, *autonomy* refers to an agent's ability to operate in complex, dynamic environments with minimal human intervention. Autonomy is not limited to control or navigation but encompasses the capacity to *learn, operate, and adapt* over time. In embodied agents, particularly robots, this includes the ability to explore environments and develop capabilities independently, a foundation for applications such as search and rescue, scientific exploration, industrial automation, and human-robot collaboration.

While all robots are physically embodied agents, the notion of an *embodied agent* also includes virtual entities that interact within simulated or digital environments. Thus, although the term *robot* is used frequently in this thesis, the proposed methods and insights are often applicable to a broader class of *agents*.

1.1.2 Problem Statement

Embodied autonomy spans a spectrum of capabilities, ranging from simple reactive behaviors to complex decision-making and long-term self-improvement. This thesis focuses on two intermediate yet foundational forms within this continuum: *active environment exploration* and *autonomous adaptation*. These forms are chosen because they mark critical transitions, from merely responding to stimuli toward proactively acquiring knowledge and adjusting behavior over time, which are essential stepping stones toward achieving full autonomous intelligence.

Environment Exploration entails an agent's capacity to actively seek and gather information about its surroundings to uncover future potentialities, such as identifying new objects, understanding spatial relationships, or identifying useful features for downstream tasks. These processes are fundamental to intelligent systems, as they

enable agents to operate in dynamic and uncertain environments without constant human supervision. Exploration helps build a structured understanding of the world, reveal hidden information, and improve decision-making.

Autonomous Adaptation is the capability to extend existing knowledge acquired during exploration, and to acquire novel behavior patterns when necessary, enabling the agent to adjust its strategies when encountering new challenges, changing conditions, or unexpected events. This form of autonomy centers on the agent's ability to adapt its own capabilities over time, which we also refer to as self-development. This process may involve adjusting internal estimation of the environment, generalizing skills to apply them to novel objects, or even learning new capabilities for novel manipulation.

Together, these capabilities are crucial for developing autonomous robots, particularly in robotic manipulation, which requires an awareness of how actions impact both the agent itself and its environment.

1.1.3 Challenges

Achieving effective environment exploration and autonomous adaptation is challenging, particularly for real-world robots, due to the complexity and unpredictability of environments. Many robotic systems are designed upon assumptions that the operations are performed in a known environment. Without this assumption, an autonomous robot, especially for long-term autonomy, faces many interrelated challenges such as hardware/software of robot platform design, and, moreover, the rapidly changing/unpredictable nature of dynamic environments.

From an environment modeling perspective, autonomous robotic systems must operate reliably in environments that neither they nor their designers have previously fully anticipated. This necessity arises from the inherent complexity of the real world, which makes it impractical for robots to fully model all possible scenarios in advance. From the control perspective, traditional symbolic planning methods explicitly model tasks and environments, solving the planning problem through optimization. However, these methods often fail or incur prohibitive computational complexity in complex and unforeseen environments, making them inflexible. A robust adaptive learning mechanism during operation is crucial for building autonomy in an open world, as encoding all necessary knowledge and skills into the system during the design phase is nearly impractical.

Challenges inevitably arise when dealing with uncertainty¹ that comes from diverse sources, *e.g.* noisy sensing, partial observability [LHP23], and instable learning [Li+23c], making it difficult to construct reliable representations for decision-

¹: Uncertainty in machine learning can be categorized into *aleatoric* and *epistemic* uncertainty [Mur22]. In the robotic context, aleatoric uncertainty arises from the inherent randomness in stochastic state transitions, while epistemic uncertainty stems from the agent's lack of full knowledge about the consequences of its actions in the environment. Addressing aleatoric uncertainty requires designing robust policies that can account for stochasticity, such as leveraging probabilistic models. On the other hand, reducing epistemic uncertainty usually necessitates active exploration strategies that prioritize information gain, enabling the agent to refine its world model and improve its predictive capabilities.

making. Adaptation further complicates the problem, as it demands the ability to generalize learned behaviors across different contexts, transfer knowledge to novel tasks, or even acquire new skills on demand. Furthermore, achieving long-term autonomy necessitates minimizing human intervention, meaning robots must develop self-directed learning strategies that enable continual improvement without explicit external supervision. Addressing these challenges is key of this thesis to building truly intelligent robotic systems capable of operating in diverse and evolving real-world settings.

1.1.4 Research Scope

To tackle the challenges raised by environment exploration and autonomous adaptation, my research focuses on four key fundamental concepts: *world model*, *semantics*, *policy*, and *self-determination* (see Figure 1.1):

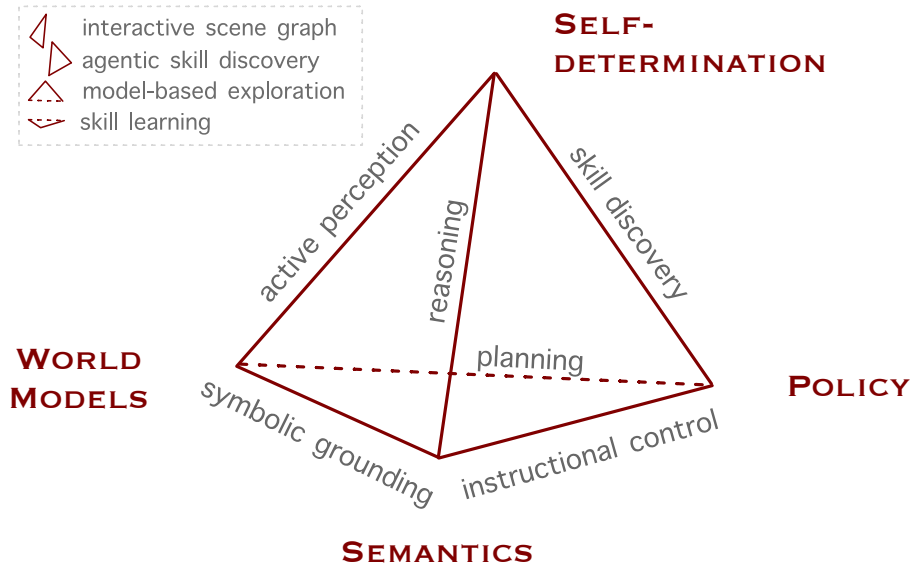


Figure 1.1: Intelligent Agent Tetrahedron. Conceptual foundation of environment exploration and autonomous adaptation, structured as a tetrahedron with four core components: *world models*, *semantics*, *policy*, and *self-determination*. Each edge represents the interaction between two concepts, while each triangular face (also shown in the top-left inset) denotes a three-way integration underlying specific capacities. For example, agentic skill discovery emerges at the intersection of *self-determination*, *semantics*, and *policy*. See Chapter 3 “Conceptual Foundations” for detailed discussion.

- **World Models** provide an agent with internal representations of its environment after prior exploration, either through fixed trajectories or active seeking, allowing it to predict the consequences of its actions, reason about uncertainty, and plan for future interactions. A robust world model helps mitigate the limitations of noisy or incomplete sensory data, improving planning and decision-making in complex environments.
- **Semantic** representations enhance the robustness of autonomy by abstracting away non-essential low-level details while preserving critical semantic and topological structures. This abstraction plays a key role in interpreting and organizing knowledge acquired through exploration. By leveraging semantics,

an agent can contextualize observations, generalize learned concepts across different tasks and environments, and communicate its understanding in a way that aligns with human expectations. Moreover, instructions may come as being casual, informal, and even incomplete to robot systems using a natural language interface; it is essential to leverage extensive knowledge and reasoning capabilities, *e.g.* from Large Language Models (LLMs), for interpretation and clarification for robust task planning.

- ▶ **Policy** learning enables an agent to develop effective strategies from exploration to operation by optimizing its actions based on past experiences and objectives. A well-trained policy controls state transitions through the decision-making process, allowing the agent to adapt its behavior to dynamic environments and achieve long-term goals. In the context of autonomous agents, policy learning bridges perception and action, ensuring that decisions are not only reactive but also proactive, aligning with both immediate feedback and strategic foresight.
- ▶ **Self-determination** in embodied context involves intrinsic motivation and self-regulated learning, enabling an agent to go beyond human-specified objectives and engage in open-ended exploration, thereby fostering continual learning and long-term autonomy. It empowers agents to set internal goals and evaluate their own progress, gaining abilities that are aligned with their own curiosities or preferences. By enabling agents to identify and pursue valuable information, and to acquire novel skills, on their own, self-determination reduces reliance on human supervision and enhances adaptability.

Together, these four components form a foundation for developing intelligent embodied systems that can explore effectively, operate robustly, and adapt autonomously in diverse and evolving environments.

1.2 RESEARCH OBJECTIVES AND CONTRIBUTIONS

By investigating the interplay between world modeling, semantic grounding, policy learning, and self-determination, this thesis aims to contribute to the foundation of self-developing autonomous embodied agents. Unlike conventional autonomy, which often prioritizes the efficient execution of predefined tasks, the proposed approach aims to develop autonomous agents with the following research objectives (O.):

OBJECTIVE I To construct self-deterministic agents that can leverage non-verbal multimodal cues to autonomously explore the environment and develop possible abilities beyond immediate task requirements.

Chapter 4 introduces **Intrinsic Sound Curiosity Module (ISCM)** to integrate cross-modal learning cues, specifically visual-auditory signal, for improved representation learning and exploration (early multimodal fusion) of Reinforcement Learning (RL) agents, with experiments addressing the following research questions:

- ▶ Does intrinsic sound curiosity help the agent to explore more actively and learn effective representations?

- ▶ Does unsupervised policy pre-training help the agent to adapt to new tasks?
- ▶ How does the choice of crossmodal prediction affect the performance?

OBJECTIVE II To develop an interactive multimodal perception framework in which the agent actively gathers, integrates, and semantically interprets diverse sensory inputs, enabling grounded semantic understanding and context-aware decision-making in complex environments.

ISCM discussed in [Chapter 4](#) investigates a learning-based (early) multimodal fusion approach to build a crossmodal predictive world model. [Chapter 5](#) further introduces a Multimodal environment chatting (Matcha) framework that integrates interactive perception with LLMs to enhance multimodal interpretation and decision-making in autonomous agents (late, decision-level multimodal fusion). Experiments in simulated multimodal manipulation scenarios study the following research questions:

- ▶ Can Matcha integrate multimodal perceptions at the decision level?
- ▶ How does the level of abstraction in the submodule outputs influence the performance?
- ▶ How do different scale LLMs affect the performance?

OBJECTIVE III To enhance agent reasoning abilities to interpret complex instructions and make informed decisions.

Matcha discussed in [Chapter 5](#) investigates the use of LLMs for complex reasoning with proper abstraction and prompting for in-context reasoning. To further improve the reasoning abilities of LLMs, [Chapter 6](#) introduces Logical Thoughts (LoT), a logic-based symbolic method to improve zero-shot chain-of-thought reasoning, enabling improved inference-time reasoning and decision-making abilities of LLMs, with experiments addressing the following research questions:

- ▶ Does LoT outperform the original zero-shot CoT, *i.e.* logic-guided inference enhances reasoning ability in various domains as well as with LLMs of varying model scales?
- ▶ What is the impact of LoT on individual reasoning chains (*e.g.* revision frequency, resultant length)?
- ▶ Do post-hoc explanations help LLM self-check?

OBJECTIVE IV To construct autonomous agents with an advanced level of self-determination that can sense environment context verbally and discover meaningful skills from scratch in the pursuit of long-term embodied autonomy.

[Chapter 7](#) proposes a semantically motivated exploration framework for RL agents, Agentic Skill Discovery (ASD), that allows agents to autonomously identify and acquire useful skills from scratch in a self-determined manner, guided by LLMs, when faced with a novel environment. The accompanying experiments address the following research questions:

- ▶ What kind of tasks will be proposed?

- ▶ Can skills be acquired automatically?
- ▶ How do RL and learning context influence the learning efficiency?
- ▶ Can challenging tasks be completed by chaining learned skills?

1.3 THESIS ORGANIZATION

Part I. Foundations and Context

- ▶ **Chapter 1 “Introduction”**: Provides an overview of the research motivation, the fundamental problem setting, and the methodology adopted in this thesis. It also outlines the key research contributions and how they address the challenges of robotic autonomy.
- ▶ **Chapter 2 “Background and Related Work”**: Introduces essential concepts in environment exploration and self-development for robotic autonomy. It also presents the experimental robot platforms and simulation environments used in this research, along with a comprehensive review of related work in the field.
- ▶ **Chapter 3 “Conceptual Foundations”**: Establishes the core theoretical foundations that facilitate robotic autonomy, including the role of world models in enabling predictive reasoning, the significance of semantics in representation, the formulation of policies for behavior control, and the concept of self-determination as a driver (intrinsic motivation) and examiner (self-regulation) for exploration and adaptation. These elements form the conceptual backbone for the research contributions discussed in the following chapters.

Part II. Core Contributions

Building upon the conceptual foundations, this part presents the core research contributions in detail. Each chapter introduces a key advancement in autonomous robotic learning:

- ▶ **Chapter 4 “Sound Guides Representations and Explorations”**: Investigates how multimodal sensory feedback, specifically visual-auditory signals, can guide robotic exploration and improve learned representations of the environment, as well as a proactive policy for downstream task adaptation. This chapter addresses **O. I** with non-verbal motivations, correlating to **O. II** with learning-based multimodal fusion during exploration.
- ▶ **Chapter 5 “Interactive Multimodal Perception Using Large Language Models”**: Explores the integration of LLM-based interactive perception to enhance multimodal understanding, addressing (decision-phase) multimodal fusion (**O. II**), and relates to complex reasoning (**O. III**) with proper prompting.
- ▶ **Chapter 6 “Enhancing Reasoning via Logic-Guided Inference Scaling”**: Introduces logic-based symbolic verification of LLM inference to improve zero-shot chain-of-thought reasoning, enabling improved inference-time reasoning abilities of LLMs (**O. III**).
- ▶ **Chapter 7 “Agentic Skill Discovery”**: Proposes an agentic skill discovery frame-

work, enabling robots to autonomously identify and acquire useful skills from scratch in a self-determined manner when faced with a novel environment. This chapter addresses [O. IV](#) with verbal motivations, inherently resembling the human learning process.

- **Chapter 8 “Reward Modeling, Embodied Planning, and Explainability”:** Discusses collaborative research efforts that complement the core contributions, including:

- § 8.1 models intrinsically motivated RL within a unified framework, additionally addressing [O. I](#) by optimizing an information-seeking objectives for diverse skill exploration.

- § 8.2 addresses [O. III](#) with a focus on context-sensitive planning, proposing a novel planning framework including object-centric planning and advanced bimanual planning that leverage LLMs.

- § 8.3 further explores the role of explainability in robotic systems, enhancing the interpretability and trustworthiness of autonomous agents.

The proposed approaches seek to empower robots with the ability to autonomously reason, explore, and adapt, ultimately pushing the boundaries of embodied intelligence.

2.1 ROBOTIC AUTONOMY

Autonomous systems [LLA21; Kun+18; Azp+23; Jan+24; Wan+23a; Hon+24; Kim+24; Xi+25; Zen+23b] possess the capability to perceive, reason, and act independently in dynamic and uncertain environments across various domains, including *physical environments* (e.g. space, air, sea, field, and human environments), *simulated environments* (e.g. Isaac Sim [NVI25], MuJoCo [TET12], Coppeliasim [RSF13], and ThreeDWorld [Gan+21]), and *textual environments* (e.g. language model generations, automated office tasks, programming context, *etc.*), thereby reducing dependence on human intervention. These systems usually integrate a selective combination of advanced Artificial Intelligence (AI) techniques (e.g. world modeling, policy learning, multimodal fusion, *etc.*) that enable agents to interpret sensory inputs, predict future states, and generate purposeful actions.

Ultimately, autonomy seeks to bridge low-level control with high-level cognition, fostering agents that can explore, learn, and operate effectively in virtual- & real-world settings, resulting in capabilities that are essential for applications such as search and rescue, autonomous inspection, and planetary exploration, where both understanding and interacting with the environment are critical for mission success.

2.1.1 Environment Exploration

Within the scope of robotic autonomy, environment exploration is the systematic process by which agentic systems autonomously perceive, navigate, and interact with unknown or partially known environments to construct spatial representations and develop exploration strategies aimed at acquiring knowledge and adaptive capabilities. This subsection presents an integrated perspective that categorizes the main exploration approaches into *map-based* and *learning-based* methods, and reviews key works in areas such as representation learning, planning, and exploration that have shaped these paradigms.

Map-Based Exploration

Environment exploration has evolved from a primary focus on robotic navigation¹ where robots learn to autonomously map and traverse unknown environments [SB03; Zhu+18; Arm+23]. Traditional map-based methods rely on explicit representations of the environment, such as grid maps, topological maps, or semantic maps [LLA21].

¹: While this thesis primarily focuses on robotic manipulation scenarios, advancements in mobile robotics will also be discussed, particularly in the context of active perception for map building, which falls under the broader scope of environment exploration.

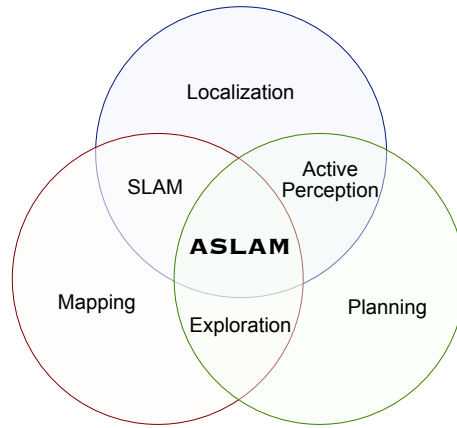


Figure 2.1: A set overlap illustration conceptualizing Active Simultaneous Localization and Mapping (ASLAM), an active robotic system that simultaneously localizes, plans paths, and builds maps (adapted from [LLA21]).

These approaches are primarily focused on improving navigation, localization, and path planning in both known and yet unknown environments. Beyond navigation, advanced exploration systems integrate manipulation capabilities, allowing robots to physically interact with their surroundings rather than merely observe. This interaction facilitates knowledge acquisition and the discovery of hidden information, such as resolving occlusions [Li+23c; Li24] or identifying invisible properties through interaction [Zha+23c; Gao+24a]. In such cases, techniques such as scene graphs can serve as a complement to traditional maps. The followings introduce fundamental concepts and related work in three key aspects: *map building*, *path planning*, and *exploration strategies*.

Map Building techniques are concerned with creating a dense representation of the environment based on sensory data. These methods are fundamental for any form of autonomous exploration, as they allow agents to build knowledge about their surroundings.

- ▶ *Simultaneous Localization and Mapping (SLAM)* [DB06; BD06] is a fundamental problem in robotics, enabling an autonomous agent to construct a map of an unknown environment while simultaneously locating its own position within the map. SLAM techniques integrate sensor data, typically from cameras, LiDAR, or other perception systems, with probabilistic estimation methods such as Kalman filters, particle filters, or graph-based optimization. Traditional SLAM focuses on geometric consistency and metric mapping, whereas more recent approaches incorporate semantic understanding to enhance navigation and interaction. In the context of autonomous exploration, *Active Simultaneous Localization and Mapping (ASLAM)* [Mu+15; LLA21] extends SLAM by incorporating decision-making strategies that guide the agent toward informative areas, balancing exploration and exploitation. This active approach enables efficient map construction while improving localization accuracy and adaptability in dynamic or partially observable environments (*cfr.* Figure 2.1).
- ▶ *Occupancy grid mapping* [ME85; LLA21] refers to approaches that divide the environment into a grid of cells (or Octree structure [Mea82] for 3D representations), each representing the probability of occupancy (*i.e.* whether the

cell is occupied or free). Occupancy grids are widely used in mobile robotics for creating real-time environment maps. This method is particularly effective for robots with limited computational power and can be enhanced to handle dynamic objects and obstacles in the environment.

While such approaches have been extensively employed in mobile navigation, they are equally pertinent to scenarios that demand manipulation-aware representations. In manipulation contexts, constructing dense 3D maps typically requires the integration of multimodal sensory inputs, such as visual, auditory, and tactile data, to generate detailed representations of objects and their properties. This process introduces increased computational complexity and necessitates a more sophisticated understanding of the environment than conventional mapping methods provide. Consequently, there is a heightened need for semantic understanding and, building upon it, high-level reasoning, both of which are central topics discussed in this thesis (cfr. [Chapter 5](#), [Chapter 6](#), and [Chapter 7](#)).

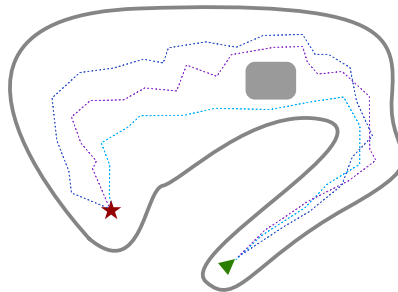


Figure 2.2: An illustration of path planning, in which the agent navigates toward a target while avoiding obstacles (shown in gray rectangle). The path is computed using a path planning algorithm, ensuring the agent avoids obstacles while reaching the target location.

Path Planning is concerned with finding a safe and efficient path through a map. It uses the built map (constructed through SLAM or other methods) to navigate the environment (cfr. [Figure 2.2](#)).

- *Sampling-based path planning* methods like Rapidly-exploring Random Trees (RRT) [LaV98], RRT* [KF11], and RRT-Connect [KL00] are used for real-time path planning in high-dimensional spaces. These methods allow agents to plan paths while avoiding obstacles, and RRT* guarantees asymptotic optimality, ensuring the paths are near-optimal. These techniques are particularly useful in dynamic environments where paths need to be recalculated on the fly.
- *Grid-based path planning* algorithms such as Dijkstra’s [Dij59; LaV06] and A* algorithm [HNR68; LaV06] are widely used to compute the optimal path from an agent’s current location to a target position on a discretized map. These algorithms are particularly efficient in environments where the map is fully known and can be represented as a grid or graph. Dijkstra’s algorithm guarantees the shortest path by exploring all possible routes in a breadth-first manner while considering edge costs. A*, an extension of Dijkstra’s algorithm, incorporates a heuristic function to guide the search more efficiently, significantly reducing computational overhead. These techniques are extensively applied in autonomous vehicles, mobile robots, and video games AI for navigation and path-finding tasks.

Planning algorithms are extensively utilized in both mobile and manipulation contexts. In manipulation tasks, planning often involves intricate constraints concerning object interaction and obstacle avoidance, treated as stringent safety measures. However, integrating numerous constraints escalates both design and computational complexity, diminishing overall adaptability and compromising real-time performance. Therefore, in scenarios where high-level decision-making supersedes fine-grained control, *e.g.* in [Chapter 5](#), trajectory planning is applied for low-level control, while high-level decision-making is managed by LLMs to ensure robust reasoning and generalization.

Map-based Exploration Strategies are methods that guide the agent’s movement within the environment to efficiently gather information². These methods often rely on the map being built so far and attempt to maximize the information gained during exploration.

- ▶ *Frontier-based exploration* [Yam97] is one of the most widely used strategies for exploring unknown environments. The agent identifies the boundaries, or “frontiers”, between the known and unknown areas of the environment. These frontiers represent regions where the map is incomplete, and the agent navigates toward these regions to expand its map. By focusing on these frontiers, the agent reduces redundant exploration and improves the efficiency of mapping large, unstructured environments.
- ▶ *Graph-structured exploration*, including *topological* [TB96; Ata15; LLA21; Mu+15] and *semantic mapping* [KG15; Yok+24], uses maps that represent the environment as a graph where nodes capture discrete locations or objects [Gu+24; Joh+15; Jia+24; Dai+24] at a high abstraction level, and edges encode their connectivity or relational context. Such graph-based representations not only provide structural (topological) guidance but also incorporate semantic context (object identities, affordances), allowing the robot to make more informed decisions during exploration.

Map-based exploration serves as a foundational strategy for robotic navigation in both familiar and novel environments, leveraging static or dynamically evolving maps to guide movement. However, such approaches typically depend on abstract, computation-intensive representations that often overlook the fine-grained details essential for precise, low-level manipulation. Environment symbolization has traditionally been handled through manual programming by domain experts, limiting adaptability and generalization across diverse environments. Recently, this process has been revisited through the integration of LLMs [Jia+19; Che+24; Chu+25; Din+23], as further discussed in [§ 3.3.3 “Integration: Planning and Learning with Foundation Models” on page 44](#) and [§ 8.2.2 “Bimanual Planning” on page 126](#), motivated by the fact that LLM-generated plans are not always reliable on their own but can be effectively combined with traditional planning methods, leveraging LLM’s strong code-generation capabilities. Learning-based approaches, in contrast, enable robots to autonomously acquire and adapt knowledge through interactive experiences and are mainly studied in this thesis.

²: The exploration strategies discussed here are not mutually exclusive, as they can complement each other depending on the context and goals of the exploration task.

Learning-Based Exploration

A prominent learning-based control approach is *Reinforcement Learning (RL)* [LZZ20; SB18], which enables autonomous agents to navigate and understand environments by acquiring knowledge through trial-and-error experience. This approach relies on *learning representations*, which encode meaningful features from high-dimensional sensory data, and *learning exploration*, which optimizes decision-making policy for environment exploration.

Learning Representations. Being fundamental to learning-based methods, representation learning enables autonomous agents to extract, encode, and leverage meaningful features from raw sensory inputs (e.g. visual, auditory, proprioceptive data). Well-structured representations help agents generalize across tasks, improve sample efficiency in RL, and make informed exploration decisions in real or virtual environments. With unrolled trajectories of an agent policy, agents are able to learn representations in self-supervised ways without the requirement of asking for human annotations. A variety of techniques have been developed to learn these latent representations, each with its own merits:

- ▶ *Autoencoders and Variational Autoencoders (VAEs)* [KW22; KW+19] compress sensory inputs into lower-dimensional embeddings by reconstructing the original data. Variational autoencoders add a probabilistic framework that encourages smooth, continuous latent spaces, which can be crucial for generating meaningful interpolations between observed states. These methods allow agents to capture the underlying structure of their environments, thereby facilitating more directed exploration.
- ▶ *Contrastive learning* methods [vLV18; WL21; LSA20; Eys+22; You+22] leverage self-supervised objectives to distinguish between similar and dissimilar observations. By maximizing agreement between augmented views of the same state while pushing apart representations of different states, contrastive methods yield embeddings that emphasize the discriminative features necessary for effective exploration. Such approaches have been shown to improve the sample efficiency of exploration policies, particularly in environments with sparse external rewards.
- ▶ *Dynamics Modeling* [Moe+23; Pat+17; DGI21] is another line of work that incorporates forward and inverse dynamics models to predict state transitions. These models not only provide a mechanism for learning robust representations but also generate *curiosity-driven* intrinsic rewards based on prediction errors (cfr. Figure 2.3). When an agent encounters states where its model poorly predicts the outcome, the resulting surprise serves as a signal to explore further, thereby enhancing the overall confidence of the environment model.

Learning effective representations is essential for building policies that generalize across tasks and environments, enabling agents to explore and adapt efficiently by capturing relevant features while filtering out irrelevant details. Recent advances in large-scale multimodal models reflect this principle by training on vast datasets spanning audio, web videos, images, 3D meshes, and more. Building on this success, recent work on Vision-Language-Action Model (VLA) [Bla+24; Int+25; Tea25]

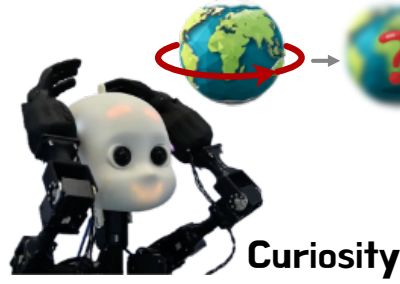


Figure 2.3: Uncertainty in state transition predictions generates *curiosity*, serving as an intrinsic reward signal that motivates the agent to explore further.

demonstrates the potential of pretraining large models on web-scale data and subsequently fine-tuning them on robotic data, yielding promising results for initializing multimodal representations and knowledge in robotics.

Despite these advances, multimodal models remain underexplored in robotics due to the high cost and complexity of collecting rich, synchronized data across modalities. Moreover, a significant domain gap exists between large-scale web datasets and the embodied, interactive settings in which robots typically operate. In such contexts, multimodal cues are often event-driven, sparse, and highly task-dependent, requiring more specialized representation learning. These challenges motivate our investigation into visual-auditory alignment in [Chapter 4](#) and interactive multimodal perception in [Chapter 5](#).

Learn to Explore. As RL agents learn within a trial-and-error paradigm, a balance between the exploration-exploitation trade-off is inevitable. While early approaches relied on simplistic methods like random action selection, modern techniques have evolved to incorporate more sophisticated strategies designed to cope with challenges such as sparse rewards, high-dimensional state spaces, and non-stationary dynamics.

- ▶ *ϵ -greedy exploration* [SB18] involves choosing a random action with probability ϵ and the best-known action otherwise. While effective in simple tasks, this strategy can be inefficient in complex environments where random exploration may rarely encounter novel states since it fails to utilize collected knowledge.
- ▶ *Curiosity-driven exploration* [Pat+17; DGI21; Gro+21; LPO18; Raj+21; Zha+22; Bur+19a; DTG20] is usually bound to dynamics modeling, whereas the intrinsic rewards are provided as the prediction error of a learned model (which can be a forward dynamics model or inverse dynamics that predicts action to make it less noise-sensitive). When the model’s prediction deviates significantly from the observed outcome, the resulting “curiosity” signal motivates the agent to further investigate that region of the state space (*cfr.* [Figure 2.3](#)). [Bou+02; VR23] maximizes knowledge gain by focusing on areas likely to reduce uncertainty. Robots evaluate potential exploration targets using metrics such as information gain, entropy reduction, or reward functions. This strategy can dynamically prioritize regions based on their expected informational value.
- ▶ *Diversity-driven exploration* methods do not necessarily model the environment dynamics but explicitly “count” the diversity of experienced states. For example, *count-based exploration* [Mar+17; Str+22; Li+23c; Lu+22] maintains visitation

counts to quantify state novelty, and *skill discovery* methods quantify empowerment, or “skill novelty” with information-theoretic measures [Wan+21; Eys+19; Li+23c]. As the agent strives to maximize the diversity of its visited states or skills³, the environment becomes thoroughly explored.

In contrast to planning-based exploration, learning-based strategies are better suited for scenarios where robot skills are not predefined but instead emerge through interaction. These strategies enable robots to adaptively discover and refine their abilities through active exploration, thereby extending their capabilities beyond fixed priors. Traditional learning approaches often operate in non-semantic spaces, relying on large amounts of data and computation to extract meaningful decision patterns, which limits their effectiveness and practicality. In comparison, advanced AI methods increasingly incorporate knowledge from models trained on large-scale datasets. The role of semantic understanding and reasoning, particularly through language models, in enhancing autonomous exploration is further examined in later chapters (see [Chapter 5](#) and [Chapter 7](#)).

2.1.2 Autonomous Adaptation

Autonomous adaptation refers to a system’s inherent capacity to modify its behavior, parameters, or structure in response to dynamic environments or novel stimuli without requiring explicit human intervention. Unlike environment exploration, which focuses on gathering and mapping external data, autonomous adaptation centers on agent self-development: the internal evolution of skills and strategies that enable the robot to meet unforeseen challenges and optimize its performance.

Learning and Generalization

Traditional machine learning frameworks focus on static datasets and batch training, where models learn to map inputs to outputs through supervised optimization. While these systems achieve high accuracy in controlled environments, their generalization capabilities are limited to the scope of their training data. Generalization focuses on how robots can extend learning from limited data or demonstrations to novel scenarios and tasks. This includes methods that allow a robot to learn underlying principles rather than merely memorizing examples. For autonomous systems operating in dynamic real-world environments (*e.g.* autonomous vehicles), this rigidity becomes a critical bottleneck. Models trained on fixed datasets struggle to adapt to novel scenarios, distribution shifts, or unseen task variations.

The No Free Lunch Theorem (NFL) [WM97] in the context of machine learning also indicates that the choice of a model should be problem-specific, considering the data distribution and assumptions⁴. As a result, approaches to augment existing data

³: In the context of skill discovery, visited states and skills are highly correlated and often interchangeable, as skills are typically defined and distinguished by the states they traverse, particularly the initial and final states.

⁴: *Do LLMs escape these limitations?* While LLMs leverage statistical patterns to perform well on many

[Höf+21; YKF21] and assimilate new data beyond the training distribution [Par+19] are necessary.

Continual Learning

Continual learning [Par+19] aims to let models learn incrementally from streaming data while retaining prior knowledge, *i.e.* avoiding catastrophic forgetting. It focuses on model and learning paradigm design to tackle the stability-plasticity dilemma, balancing the maintenance of old knowledge (stability) with the integration of new information (plasticity).

In robotics, continual learning approaches are valuable for developing models that support lifelong environment modeling and multi-task policy learning. They are particularly beneficial for resource-constrained devices and time-sensitive control scenarios where robots must adapt to changing tasks. A major challenge, however, emerges when shifting from human-curated task sequences to self-directed learning (*cfr.* Chapter 7), where the system must autonomously decide what and when to learn, and how to integrate new capabilities into existing models. While the methods explored in this thesis mitigate forgetting by storing and switching between multiple models, scaling to more complex behaviors, particularly in self-supervised settings, underscores the importance of continual learning for effective knowledge management and long-term adaptation. A detailed investigation of such continual learning mechanisms, however, lies beyond the scope of this thesis.

Skill Discovery

Skill discovery is the process of identifying structured, reusable behaviors that allow an agent to solve tasks efficiently [Las+21b; Yan+25; BSK21; Rho+25; Kam+22; KPK21; Las+21a; Sha+20; Eys+19]. These behaviors, often referred to as “skills” or “options” [SB18], are temporally extended actions that go beyond primitive actions like moving forward or turning left. It typically involves two key components:

- ▶ Representation learning, which identifies latent structures in unlabeled data (*e.g.* object affordance in a cluttered room).
- ▶ Policy learning, where optimizing algorithms rapidly acquire new skills by distinguishing them from accumulated ones. Skills are often encoded as latent variables in a policy, allowing the agent to generalize across tasks by reusing learned behaviors.

In a Markov Decision Process (MDP) setting, skill discovery can be modeled as learning a policy $\pi(a|s, z)$, where $s \in \mathcal{S}$ is the current state, $a \in \mathcal{A}$ is the action, and $z \in \mathcal{Z}$ is a latent variable representing the skill. The objective is to discover diverse and distinguishable skills by maximizing the mutual information $I(\tau; z)$ between the latent variable z and trajectory $\tau = \{s, a\}_T$, *i.e.* skills are distinguishable in terms of

problems, their adaptability remains constrained by training data priors. Their emergent abilities do not exempt them from the NFL theorem; instead, methods like fine-tuning and in-context adaptation are needed to extend their capabilities beyond their original distribution.

the state-action pairs visited by the agent:

$$I(\tau; z) = H(z) - H(z|\tau),$$

where $H(z)$ is the entropy of the skill distribution (ensuring diversity), and $H(z|\tau)$ is the conditional entropy of the skill given the trajectory (ensuring distinguishability among skills, *cfr.* Figure 2.4). Expanding $I(\tau; z)$ with probability integral leads to

$$I(\tau; z) = \mathbb{E}_{\tau, z \sim p(\tau, z)} [\log p(z|\tau) - \log p(z)],$$

where $p(\tau, z)$ is the joint distribution of trajectories and skills; $p(z|\tau)$ is the posterior probability of the skill given the trajectory; and $p(z)$ is the marginal probability of the skill.

To approximate $p(z|\tau)$, a learnable discriminator $q_\phi(z|\tau)$ with parameters ϕ can be introduced to optimize the Barber-Agakov lower bound [BA03; Li+23c] of $I(\tau; z)$, leading to RL objective

$$\mathcal{J}(\pi) = \mathbb{E}_{z \sim p(z), \tau \sim p(\tau|z)} [\log q_\phi(z|\tau) - \log p(z)],$$

where $p(\tau|z)$ is the trajectory distribution induced by the policy $\pi(a|s, z)$ for a chosen skill z .

For long-term autonomy, it is promising for intelligent systems to pursue learning outcomes that are novel to the initial setting. Techniques like intrinsic motivation [Pat+17] or compositional skill libraries [Zha+23a] enable agents to autonomously chain primitive skills into complex behaviors. Skill discovery via Unsupervised Reinforcement Learning (URL) presents a promising approach by generating trajectories and clustering them into distinct skill categories without requiring human supervision. Despite its appeal, its practical application remains limited due to substantial data requirements and current feasibility only in simulation, constrained by the sim-to-real gap. Moreover, the discovered skills often exhibit non-determinism across training runs and lack direct interpretability, posing challenges for understanding and systematic reuse. To address these issues, this thesis investigates self-determined mechanisms (*cfr.* § 3.4) of robot learning and further explores the integration of LLMs to support semantic reasoning over environment-centric goals (*cfr.* Chapter 7), aiming for more effective and interpretable skill discovery processes.

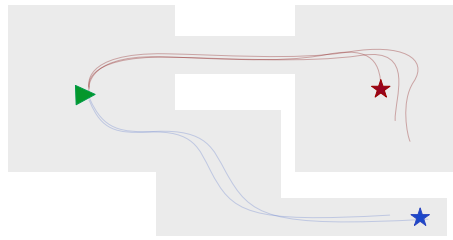


Figure 2.4: Skill discovery illustration, where the upper red and lower blue trajectories indicate skill distinction $z_i \neq z_j$ defined by novel state visiting, which can be unsupervisedly discovered.

2.2 ROBOT SETTING

This section introduces two robots that are used in later chapters: (1) NICOL (§ 2.2.1), a robot built by our group *Knowledge Technology*⁵, and (2) Franka Emika Panda (§ 2.2.2), which has been widely used in research due to its high flexibility.

2.2.1 NICOL

The Neuro-Inspired COLlaborative semi-human robot (NICOL) [Ker+23; Zha+23c] is a semi-humanoid robot designed to enhance human-robot interaction and collaboration beyond verbal communication. It consists of a *head*, an upper body with two *arms*, and a structured table *workspace*. See Figure 2.5 for individual robot parts and Figure 2.6 for the whole robot setting in both real- and virtual environments. Research works introduced later in Chapter 5 and Chapter 8 mainly rely on this platform.

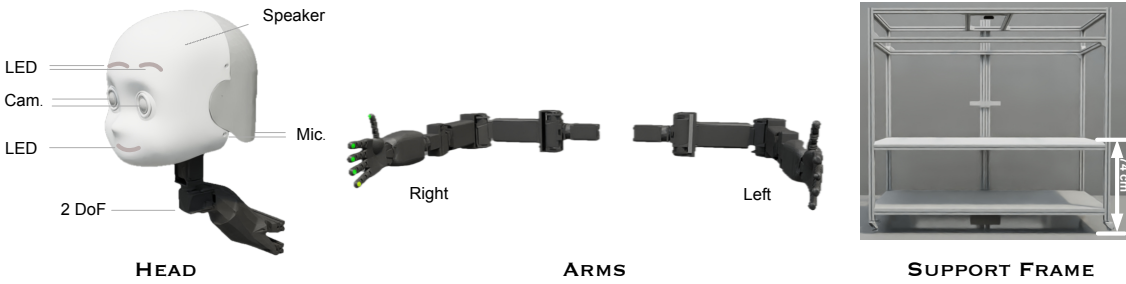


Figure 2.5: NICOL’s head, arms and workspace shown as component parts respectively. See also Figure 2.6 on the facing page for the whole robot, both in the real world and simulation.

Head. NICOL’s head is adapted from iCub [Met+10] and features two Degrees of Freedom (DoF) for pitch and yaw movements. Being different from iCub, NICOL’s head is equipped with two See3CAM CU135 cameras for vision and two Soundman OKM II binaural microphones for auditory perception, along with an internal speaker for audio output. Stylized facial expressions are displayed using three LED arrays: two 8x8 arrays behind the eyes and a 16x8 array behind the mouth, facilitating expressive and interactive visual communication.

Arms. The robot’s upper body incorporates two OpenManipulator-P⁶ arms, each with 6 DoF and a payload capacity of 3kg. These arms are fitted with SeedRobotics RH8D adult-sized robotic hands⁷, which serve as end-effectors with a 750g manipulation payload. Each hand comprises five tendon-operated fingers, with every three-segment finger controlled by a single tendon.

Workspace. NICOL is centrally mounted above a 100x200 cm white table, positioned at a height of 74 cm. An aluminum profile frame securely supports the structure, ensuring stability and precision.

5: <https://www.inf.uni-hamburg.de/en/inst/ab/wtm.html>

6: https://github.com/ROBOTIS-GIT/open_manipulator_p

7: <https://www.seedrobotics.com/rh8d-adult-robot-hand>

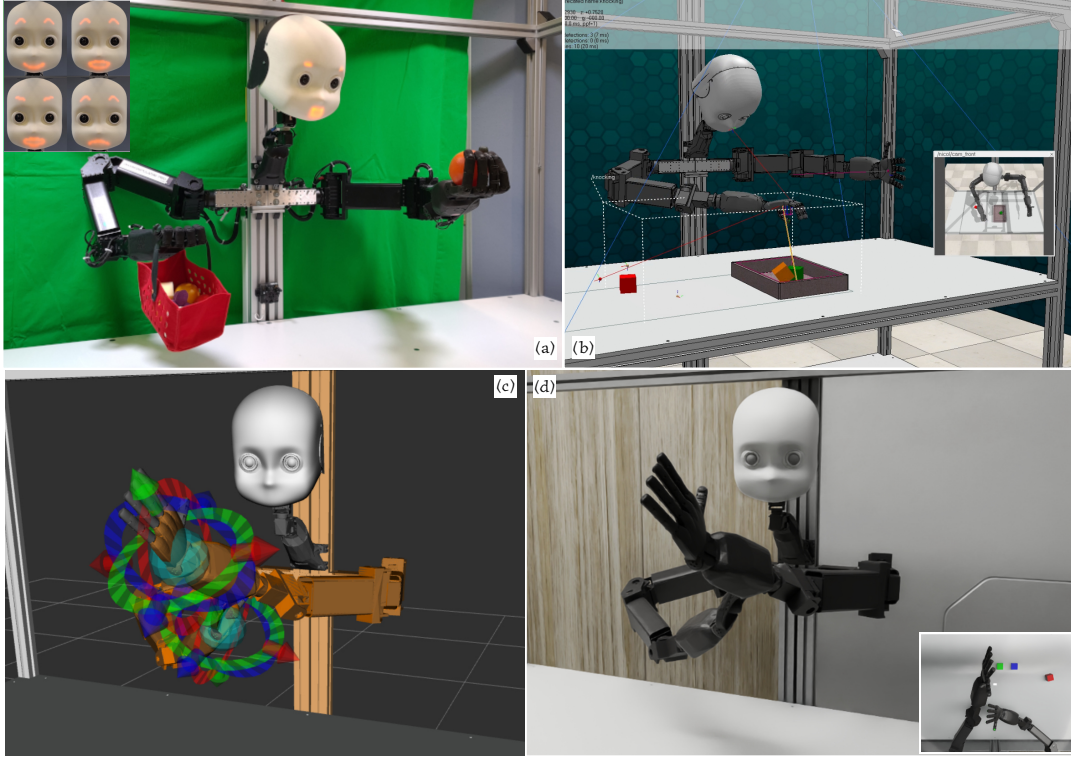


Figure 2.6: Semi-humanoid robot NICOL, the Neuro-Inspired COLlaborator. (a) NICOL in the real world, displaying various facial expressions with LED lights (image adapted from [Ker+23]); (b) NICOL in the Coppeliasim (formerly V-REP) simulator [RSF13]; (c) Bimanual planning with MoveIt2 [Con25], using both palms as end-effectors for planning; (d) NICOL in Isaac Sim [NVI25] with realistic rendering (side and top view), communicating with MoveIt2 via ROS2 support [NVI25; Qui+09; Ric22].

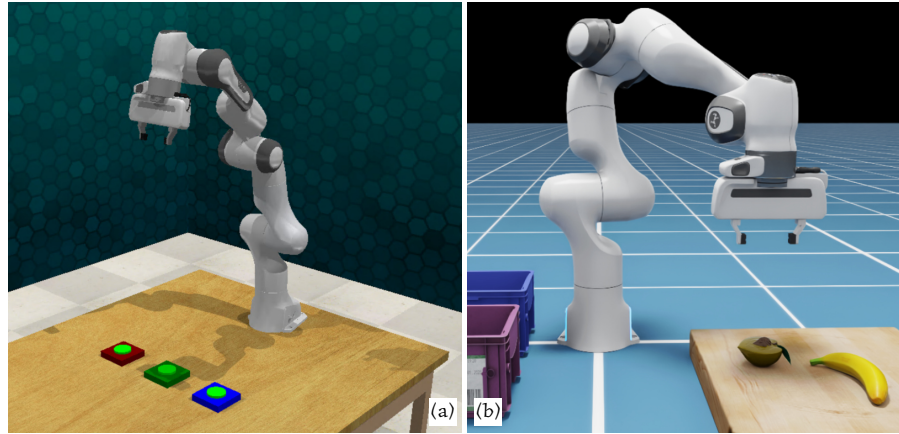


Figure 2.7: Panda robot in simulation. (a) a “push button” task in Coppeliasim (image adapted from our work [Chu+24b]); (b) a “pick-place” task in Isaac Sim.

2.2.2 Franka Emika Panda

The Franka Emika Panda [Rob25] is a 7-DoF collaborative robotic arm designed for precision manipulation and safe human-robot interaction. Each joint is equipped with integrated torque sensors, enabling compliant control and force-sensitive operations. With a 3kg payload, 850mm reach, and 0.1mm repeatability, the Panda is well-suited for tasks requiring dexterity and accuracy. The Panda robot is equipped with the Franka Hand, a two-finger gripper with a 70N continuous grasping force and 80mm

stroke length, allowing versatile object handling across research and industrial applications.

The Panda robot serves as a dedicated platform for manipulation tasks, though it has limitations in multimodal perception and communication capabilities as seen in NICOL. It has been widely adopted in research due to its open-source control interface, comprehensive documentation, and strong community support, all of which enhance the reproducibility and accessibility of robotics research. Works using the Panda robot are discussed in [Chapter 7](#) and [Chapter 8](#).

2.3 SIMULATION SETTING

2.3.1 ThreeDWorld

ThreeDWorld [Gan+21]⁸ is an advanced, multi-modal simulation platform that blends near-photorealistic rendering with advanced physics to create highly realistic 3D environments. ThreeDWorld utilizes the Unity3D engine to generate detailed indoor and outdoor scenes enriched with dynamic lighting and high-quality textures. Its physics engines support fast, accurate rigid-body interactions alongside sophisticated soft-body, cloth, and fluid simulations.

A distinguishing feature that sets ThreeDWorld apart from many other simulators is its support for physics-based auditory simulation. Its high-fidelity audio subsystem, powered by tools such as PyImpact [TCM19], enables the real-time synthesis of realistic impact and environment sounds (*e.g.* reverberation). This unique support serves as the simulating foundation for the visual-auditory experiments that will be introduced later in [Chapter 4 on page 55](#).

Simulating Impact Sound

As impact sound generally exists in reality and provides rich information, integrating it within simulated environments to guide embodied agents is promising. In [Chapter 4](#), the research shows how visual-auditory correspondence guides RL in terms of both representation learning and exploration. The experiment is conducted in the ThreeDWorld simulator. However, this does not necessarily mean a simulator with sound-simulating ability is mandatory for such experiments. Below are two possible alternatives.

Online Simulation. It is also possible to simulate sound in real-time outside the physics simulator, provided that collision information, such as contact force, object mass, and geometry, is available. Impact sounds can be computed simultaneously in a separate thread, independent of the physics simulator. [Appendix B “Impact Sound Simulation” on page 139](#) presents both the theoretical foundation and practical implementation of physics-based impact sound synthesis, which underpins sound simulations in

⁸: See also its webpage <https://www.threedworld.org/> for a demo and <https://github.com/threedworld-mit/tdw> for partially open-sourced codes.

environments like ThreeDWorld and can be applied in other systems where the physics engine lacks native sound support.

Offline Simulation. Pre-recorded audio can be played when collisions are detected. These recordings may originate from computationally intensive synthesis processes or real-world datasets. Offline simulation is straightforward to implement and offers flexibility in controlling data quality. The research experiment in [Chapter 5 on page 69](#) utilizes this approach to simulate audio.

Although ThreeDWorld offers high-fidelity vision and sound simulation, it is not widely adopted in the robotics community as other simulators introduced in this section. One reason is its lack of native support for parallel simulation. Additionally, converting both robot models and scene assets into the Unity3D format can be cumbersome and error-prone, particularly for large-scale experiments.

2.3.2 Coppeliasim

Coppeliasim [RSF13] is a robust robotics simulation platform widely used in academic research and education. Its intuitive interface and support for multiple physics engines (Bullet, Mujoco, Newton *etc.*) provide flexibility for diverse simulation requirements. Researchers benefit from extensive pre-built assets (and also configured well-known robots such as UR-series) and various programming interfaces through embedded scripts and community-supported Python APIs⁹. The platform has proven reliable through implementation in numerous peer-reviewed studies, while extensions like RLbench [Jam+20] have created standardized environments for RL research and easy tools for recording demonstrations. The research works in [Chapter 5](#) and some in [Chapter 8](#) are conducted in Coppeliasim, leveraging RLbench’s built-in motion planning capabilities and straightforward task creation framework. While maintaining these advantages, the standard Panda robot (§ 2.2.2) is replaced with the NICOL robot (§ 2.2.1) for this investigation.

Coppeliasim’s strengths lie in its ease of use, extensive community support, and compatibility with various physics engines, making it a versatile choice for many robotics applications. However, Coppeliasim is limited in rendering photorealistic images compared to modern simulators, posing a constraint as high-fidelity visuals become increasingly vital for vision-based robotics research. Additionally, Coppeliasim was not originally designed for parallel learning with multiple instances, which constrains its applicability for data-intensive machine learning approaches, *e.g.* RL, that require parallel simulation. These limitations may impact its suitability for certain cutting-edge research applications that demand high visual fidelity or large-scale parallel training.

2.3.3 Isaac Sim

NVIDIA Isaac Sim [NVI25] is a state-of-the-art robotics simulation platform that provides photorealistic rendering (*cfr.* [Figure 2.6](#) and [Figure 2.7](#)) and parallel sim-

9: See <https://github.com/stepjam/PyRep>.

ulation capabilities for robotics research and development. For perception tasks, it supports the simulation of various sensors, including cameras, LiDAR, and depth sensors, making it highly suitable for perception-based robotics research. Furthermore, it enables large-scale domain randomization and synthetic data generation, which are particularly beneficial for data-intensive deep-learning applications. Isaac Sim integrates well with ROS/ROS2, bridging the gap between simulation-based development and real-world deployment. Additionally, its GPU-accelerated parallel simulation significantly enhances the efficiency of RL for training robotic agents. Parallel simulation enables researchers to run numerous training environments simultaneously, substantially reducing convergence time and improving the overall efficiency of the development pipeline.

Isaac Sim gains increasing popularity in the robotics community, particularly for RL and perception tasks, due to its high-fidelity rendering, extensive sensor support, and parallel simulation capabilities. Parallelization is particularly beneficial for iterative trial-and-error processes, such as designing and fine-tuning reward functions in RL tasks. For instance, when reward functions are generated and refined using LLMs [Ma+24b; ZWW24], rapid testing and evaluation of multiple variations become essential. The research introduced in [Chapter 7 on page 99](#) leverages Isaac Sim to study skill learning under various LLM-generated reward functions. Parallel simulation support significantly accelerates training, facilitating exhaustive exploration of learning parameters.

2.3.4 Other Simulators

Several other simulators are widely used in the robotics and AI research community, including PyBullet [CB21], MuJoCo [TET12], AI2-THOR [Kol+22; Ehs+21], Gazebo [KH04], *etc.*. PyBullet and MuJoCo are particularly known for their efficient physics-based simulations, making them popular choices for RL tasks. AI2-THOR specializes in interactive 3D environments, making it well-suited for embodied AI research, such as navigation and object manipulation at an abstract level. Gazebo, a long-established simulator, is frequently used in conjunction with ROS for robotics development and testing. While these simulators offer powerful capabilities and are widely adopted across various research domains, this thesis focuses exclusively on ThreeDWorld, CoppeliaSim, and Isaac Sim. Therefore, detailed discussions of other simulators are beyond the scope of this work. For comprehensive comparisons and analyses of simulation platforms, readers can refer to existing surveys [Dua+22; Col+21].

2.4 PERSPECTIVES FROM THE LITERATURE

The field of autonomous agents has been extensively studied, resulting in numerous surveys that categorize existing works based on different embodiments, applied methodologies, and application domains. The advent of LLMs has significantly transformed the design and development of autonomous agents. This shift warrants a reassessment of literature surveys in the field, distinguishing between those conducted

before and after the integration of LLMs into autonomous systems. This section categorizes literature surveys accordingly.

2.4.1 Early Conceptualizations

Before the advent of LLMs, research on autonomous agents primarily focused on classical robotics, RL, and a narrower scope of AI approaches. The following surveys provide insights into various aspects of autonomous agent design prior to the emergence of large foundation models:

- ▶ The survey by Lluvia *et al.* [LLA21] specifically reviews key SLAM and ASLAM research for indoor mobile robots.¹⁰ It includes a comparative analysis of various approaches based on factors such as robot platforms, sensor modalities, world representations, core contributions, optimization objectives, and testing environments (real-world or simulated).
- ▶ Azpurua *et al.* [Azp+23] summarize robotic exploration techniques, especially for subterranean environments, with a focus on various sensors used for localization and SLAM (*e.g.*, event cameras, stereo cameras, and active RGB-D cameras). Their research provides a taxonomy of exploration methods based on map representation (2D or 3D) and deployment strategy (single- or multi-robot systems).
- ▶ Latyshev *et al.* [LP23] provide a technical overview of intrinsic motivation in model-based RL, covering data collection strategies, loss formulations, major types of intrinsic signals from world models, and the incorporation of intrinsic rewards into RL frameworks.
- ▶ Kunze *et al.* [Kun+18] review the application of AI methods to long-term robot autonomy—operations lasting days, months, or even years. They introduce key domains and describe how AI contributes to robotic modules such as navigation and mapping, perception, reasoning, planning, human-robot interaction, and learning.
- ▶ Kroemer *et al.* [KNK21] formalize the problem of robot manipulation learning and highlight major challenges. Their survey categorizes manipulation learning methods across five dimensions: (1) transition models, (2) object-centric representations, (3) skill learning, (4) hierarchical task decomposition, and (5) preconditions and postconditions.

Early surveys in the field of autonomous agents primarily focused on classical robotics, offering valuable insights into the hardware and software aspects of autonomous system design and development. However, they often lack a comprehensive perspec-

[LLA21]: Lluvia *et al.* (2021), ‘Active Mapping and Robot Exploration’

¹⁰: Discussed also in § 2.1 “Robotic Autonomy” on page 9

[Azp+23]: Azpurua *et al.* (2023), ‘A Survey on the Autonomous Exploration of Confined Subterranean Spaces’

[LP23]: Latyshev *et al.* (2023), ‘Intrinsic Motivation in Model-Based Reinforcement Learning’

[Kun+18]: Kunze *et al.* (2018), ‘Artificial Intelligence for Long-Term Robot Autonomy: A Survey’

[KNK21]: Kroemer *et al.* (2021), ‘A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms’

tive on the broader implications of advanced AI techniques and their integration into autonomous agents.

2.4.2 Evolving Perspectives in the Era of LLMs

With the rapid development of LLMs, these models have played an increasingly central role in the implementation of autonomous agents due to their extensive knowledge and powerful reasoning capabilities. This has led to a surge of literature surveys on LLM-based agents. Many of these surveys primarily focus on general AI agents and may overlook the specific challenges and methodologies relevant to robotics.

- ▶ Xi *et al.* [Xi+25] provide a broad survey on LLM-based agents, covering fundamental concepts, system architectures, real-world applications, societal implications, and emerging trends. They introduce a high-level conceptual framework for AI agents, structured around three primary components: brain (cognitive processing), perception (sensory input), and action (decision-making and control).
- ▶ Jang *et al.* [Jan+24] explore the integration of foundation models into robotic systems and their impact on perception, planning, and control. The study discusses relevant datasets, augmentation strategies, and real versus simulated robot experiments. Although the survey adopts an application-oriented perspective, it lacks a detailed discussion of methodological innovations.
- ▶ Wang *et al.* [Wan+24b] review recent advancements in applying LLMs to robotics, particularly in planning, manipulation, and reasoning. They highlight how LLM-based planning methods leverage general-purpose knowledge and reasoning to enable embodied agents to generalize across tasks and adapt to unforeseen challenges.
- ▶ Zeng *et al.* [Zen+23b] survey LLM-driven innovations in robotics, introducing various LLM models and their benefits. Their review focuses on techniques for developing four core modules: perception, decision-making, control, and sim-to-real interaction.

These surveys reflect the growing influence of LLMs in the field of autonomous agents, signaling a paradigm shift in how intelligent systems are designed and deployed. While existing surveys typically categorize research by agent components and applications, their conceptual frameworks often remain too abstract for practical robotic platform development. In contrast, this thesis adopts a more structured approach focused on environment exploration and autonomous adaptation, establishing a concrete conceptual foundation built on four key pillars: world modeling, semantic grounding, policy learning, and self-determination mechanisms.

[Xi+25]: Xi et al. (2025), 'The Rise and Potential of Large Language Model Based Agents: A Survey'
 [Jan+24]: Jang et al. (2024), 'Unlocking Robotic Autonomy: A Survey on the Applications of Foundation Models'

[Wan+24b]: Wang et al. (2024), 'Large Language Models for Robotics: Opportunities, Challenges, and Perspectives'

[Zen+23b]: Zeng et al. (2023), 'Large Language Models for Robotics: A Survey'

CONCEPTUAL FOUNDATIONS

Exploration and adaptation are fundamental challenges in robotics, where autonomous agents are typically designed to interact with their environment in order to learn how to perform specific, predefined tasks. However, in more advanced settings, exploration is not limited to task execution but extends to uncovering what kinds of tasks or skills an agent could potentially acquire within a given environment. For instance, instead of simply learning to push a box to a goal location, a curious robot might investigate whether it can roll, stack, or throw the box, discovering new capabilities that were not explicitly programmed or instructed. This shift moves exploration from being goal-driven to being agentic, where the robot actively probes the environment to infer its own affordances and latent competencies, enabling agents to understand spatial structures, identify key features, and adapt to dynamic conditions.

In this chapter, the four conceptual foundations for autonomous agents, *world model*, *semantics*, *policy*, and *self-determination*, will be introduced, both individually and in terms of their interconnected research overlaps. Collectively, these foundations offer a theoretical framework that guides agents beyond reactive behaviors toward deliberate, adaptive interactions with their environment, thereby expanding their capabilities.

3.1 WORLD MODEL

Environment exploration necessitates the storage of accumulated knowledge for potential exploitation, which can take distinct forms depending on the task context. The choice of representation reflects different levels of abstraction suited to the agent's operational needs. In navigation scenarios, representations such as occupancy maps [LLA21; DB06; BD06; Wu+20; Mu+15], potentially enriched with semantic annotations, enable spatial awareness and efficient path planning. For manipulation, particularly in cluttered environments, scene graphs provide structured knowledge representations that facilitate object interaction and task execution [Gu+24; Joh+15; Ran+23; Jia+24; Dai+24].

At the core of these representations lies the concept of world models [HS18], which serve as internal frameworks for intelligent agents, ranging from robots and autonomous vehicles to simulated entities, to comprehend, anticipate, and interact with their environments. By encoding high-dimensional sensory input into compressed, predictive structures, world models allow agents to simulate possible future states and make informed decisions. As a result, they have become central to model-based Reinforcement Learning (RL), autonomous navigation, and complex task planning, effectively linking perception with action and enhancing an agent's ability to adapt and operate autonomously.

3.1.1 Sensing, Perception and Multimodal Fusion

The capacity to accurately sense and perceive the environment forms the first crucial step in building effective world models. Modern autonomous systems rely on a diverse array of sensors, including RGB(-D) cameras, event cameras, lidars, radars, tactile and proximity sensors, microphones, GPS, and inertial measurement units (IMUs), to capture rich and complementary data streams. Integrating these heterogeneous sources of information, a process known as multimodal fusion, is essential for generating a reliable and complete internal representation of the world. This subsection discusses the techniques, challenges, and importance of multimodal sensing and perception in the context of autonomous systems and world models.

Sensing

Autonomous agents utilize various sensors, each offering distinct advantages:

Vision sensors (RGB, RGB-D, event cameras, Lidar, and Radar). RGB cameras provide high-resolution spatial information with rich color and texture details, serving as the foundation for object recognition, scene understanding, and visual navigation. RGB-D cameras extend this capability by adding depth information, enabling more accurate 3D perception and spatial reasoning. Event cameras, on the other hand, operate asynchronously by detecting changes in brightness at high temporal resolutions. They offer low-latency, high-dynamic-range data, making them particularly effective in high-speed or high-contrast environments. Together, these vision sensors offer complementary strengths for robust and efficient perception in dynamic real-world settings. Lidar and radar provide precise distance measurements and 3D point clouds. Lidar excels at detailed spatial mapping, while radar offers robustness in adverse weather. These sensors complement vision modalities: vision captures rich semantic details under good lighting, whereas Lidar and radar provide reliable depth and structure regardless of illumination.

Audio sensors capture auditory cues from the environment, complementing visual data and aiding in detecting events that might not be visible, such as alarms or verbal commands, which can be critical in complex navigation [Che+20] or manipulation [Zha+22; Gan+20a] environments.

Tactile and proximity sensors offer fine-grained feedback about immediate physical interactions and distances to nearby objects. Tactile sensors enable robots to sense texture, force, and slip during manipulation tasks, while proximity sensors help in collision avoidance during close-range operations.

Locating sensors (GPS, IMUs). GPS provides global positioning data essential for large-scale navigation and localization, especially in outdoor environments. It serves as a reference to correct drift from other sensors. In contrast, IMUs measure acceleration and angular velocity, offering high-frequency updates for ego-motion estimation and stabilization. Combined, GPS and IMUs enable precise and reliable localization of the robot and its components.

Different sensor modalities provide complementary information, collectively enhanc-

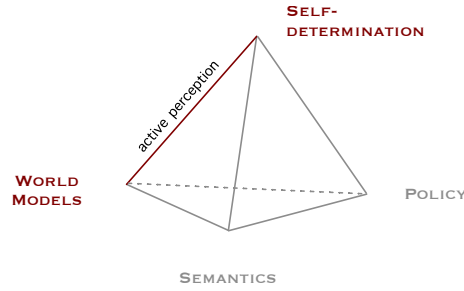


Figure 3.1: Active Perception = World Models + Self-determination. Embodied agents engage in active perception by interacting with their environment to construct world models, which are then used to make predictions and guide exploratory actions. This process is driven by robotic self-determination (cfr. § 3.4), wherein the agent autonomously selects what information to gather based on intrinsic motivations and internal measures of relevance or uncertainty.

ing the agent’s perception and understanding of its environment. Each modality offers a distinct trade-off between information richness and computational or energy budget. For example, in [Chapter 4](#), visual and auditory sensing are utilized; [Chapter 5](#) expands this by integrating visual, auditory, and tactile modalities. The unique aspects of the environment captured by each modality form the foundation for effective multimodal fusion, enabling the agent to build richer and more robust representations.

Perception

Perception is the process of transforming raw sensory data into structured representations that can be used for decision-making and planning. Perception methods can be categorized into *passive perception*, where the system passively observes and processes incoming sensor data, *active perception*, where the system deliberately perceives what is deemed important, and *interactive perception*, where the perception process involves interactive engagement with the environment to gather information.

Passive perception. Passive perception involves collecting and interpreting sensory data without adaptive strategies or direct interaction with the environment. It is well-suited for data-driven learning, enabling large-scale dataset collection without manual intervention. However, in *partially observable* settings, passive perception often suffers from ambiguity due to missing or unreliable information, limiting its effectiveness on its own.

Active perception involves establishing a data acquisition and interpretation process that, in turn, leads to the development of a control strategy aimed at maximizing the most relevant information for a given motivation, usually involving an establishment of an internal representation of the environment, *i.e.* a world model (cfr. [Figure 3.1](#)). Active perception has been widely researched in the context of computer vision, where the objective is usually to choose the next best view for object reconstruction [LLA21]. In the context of manipulation, robots that actively engage in physical interactions with their environment often achieve more accurate estimates of environmental states [Zha+23c]¹.

¹: In [Chapter 5 on page 69](#), the robot, guided by LLMs, actively perceives the environment via

Interactive perception describes the type of perception that requires the robot to *interact* with the environment, which entails a modification of environment states, emphasizing the relation between robot and environment. It extends beyond passive observation by engaging with objects or the environment to infer properties. For example, a robot may manipulate an object to determine its weight [Zha+23c] or shake a container to assess its contents [Epp+18]. This approach enhances perception by leveraging physical interaction.

While active and interactive perception often overlap, they have distinct focuses. Active perception involves deliberate sensor adjustments or movement to improve perception without necessarily interacting with objects. For instance, a robot *actively* changing its viewpoint to avoid occlusion [Li+23c] exemplifies active perception but not *interactive* perception. Given this distinction, one could conceptualize *interactive perception* combined with *active learning* [Ren+21; Wan+21] as forming *active perception* in a broader sense.

The choice of perception strategy depends on the task and environment. In complex or dynamic settings, active perception enhances the agent’s ability to gather relevant information, enabling better decision-making and planning through richer world models. However, interactive perception strategies are less commonly explored in the literature, as they require more sophisticated control and planning mechanisms to ensure that the robot’s actions yield informative observations. These strategies also tend to lack generalizability across different tasks and domains. Recognizing the importance of active perception as a key component of autonomous exploration and capability adaptation, this thesis examines it from multiple perspectives, ranging from Unsupervised Reinforcement Learning (URL), *cfr.* Chapter 4, to the integration of language models, *cfr.* Chapter 5 and Chapter 7.

Multimodal fusion

Multimodal fusion combines diverse sensory inputs, *e.g.* vision, audio, and tactile sensing, to enhance perception and decision-making [Atr+10]. Functionally, it can be categorized as *complementary*, where each modality contributes unique, non-overlapping information to mitigate limitations like sensor noise or occlusion, or *redundant*, where overlapping modalities reinforce the same information to improve robustness and reliability. Importantly, learning crossmodal associations from such overlaps can also lead to the development of rich and generalizable representations.

Fusing Levels. Multimodal fusion occurs at different levels/stages of processing:

- ▶ Early fusion (sensor-level fusion) combines raw sensory data before feature extraction, preserving maximum information but requiring precise synchronization and alignment.
- ▶ Mid-level fusion (feature-level fusion) integrates extracted features from different modalities, leveraging shared representations to enhance robustness while balancing efficiency (see also an example in Chapter 4).
- ▶ Late fusion (decision-level fusion) merges modality-specific predictions at

multiple cues to uncover hidden object properties.

the decision stage, often using ensemble methods or probabilistic models to improve reliability (see also an example in [Chapter 5](#)).

Fusing Methods. To implement multimodal fusion strategies, researchers rely on a range of computational paradigms and techniques. Probabilistic models, such as Bayesian networks, provide a principled framework for handling uncertainty in sensory integration. Deep learning-based methods, including transformers [Vas+17] and graph neural networks, support scalable learning of complex cross-modal representations and can flexibly model complementary, redundant, or hierarchical relationships. Within these paradigms, specific techniques such as contrastive learning [vLV18; Rad+21] are employed to align modality-specific embeddings in a shared space, while active learning approaches [Rud+19; Wan+21] enable dynamic adjustment of fusion mechanisms based on task relevance, promoting context-aware integration.

Multimodal fusion is essential in applications such as human-robot interaction, object property discovery, autonomous navigation, *etc.*, where integrating diverse sensory inputs enhances situational awareness. Nevertheless, integrating diverse sensory modalities increases computational demands and requires careful alignment to ensure compatibility across data sources. Fusion strategies must be tailored to task-specific needs, balancing integration depth and abstraction. For example, [Chapter 4](#) employs early fusion of visual and auditory inputs for environmental understanding; [Chapter 5](#) integrates visual, auditory, and tactile cues at the decision stage for complex object interactions; and [Chapter 7](#) focuses on visual semantics, enhancing scene reasoning through fusion with language models.

3.1.2 Modeling and Representations

World models serve as internal representations of the environment, enabling artificial agents to perceive, predict, and interact with their surroundings. The process of constructing these models involves two key components: (1) *world modeling*, representing the world, *i.e.* state transitions, with appropriate structure, and (2) *learning representations*, developing methods for learning effective representations. In this subsection, we first examine the various ways in which the world can be represented, ranging from high-level abstractions to low-level sensory data. We then discuss the methods employed to learn these representations, particularly those based on self-supervised learning.

World Modeling

World modeling involves constructing internal representations of the environment that enable an agent to perceive, predict, and interact with its surroundings effectively [HS18]. These models can vary from low-level details to high-level abstractions, each tailored to support different tasks and objectives.

Low-level Representations of the world are more detailed and quantitative, capturing the fine-grained dynamics of the environment. Probability models such as Markov Decision Processes (MDPs) and occupancy grid maps provide information about

state transitions and spatial layouts. These models are critical for precise navigation and control, where the agent must understand both the locations and the dynamics of obstacles and free space. Representation learning extracts relevant features from the environment, focusing on task-relevant properties while disregarding irrelevant ones. It simplifies skill and model learning, enhancing robustness and generalization to new situations.

High-level Abstraction is beneficial for high-level planning, generalization, and interpretation. At the highest level of abstraction, world models can be represented using symbolic representations, for example, in graph-based structures where nodes correspond to key entities in the environment, and edges encode the relationships or transitions between these entities. Such models are particularly useful for high-level reasoning and planning, where the environment can be reduced to a network of interconnected states or objects. Some tasks semantically resemble each other, which makes it possible for a learned policy to generalize across them with an additional effort of domain adaptation [KNK21].

Mixed-level Representations. For mobile robots, world representation is typically structured into two main types: topological maps and metric maps [LLA21; Azp+23]. Each type has its strengths and is suited for different tasks.

- ▶ *Metric maps* such as occupancy grid maps and geometric maps, provide denser, more detailed information about the environment, including the locations of obstacles and free space. While more informative, metric maps are computationally more expensive to store and process.
- ▶ *Topological maps* represent the environment as a network of discrete locations and their connectivity. This simple representation is efficient and easy to compute, but lacks detailed information about distances or the layout of the environment.

In practice, a combination of topological and metric maps is often used, allowing robots to benefit from the simplicity of topological maps while still incorporating the detailed information provided by metric maps when necessary. This hybrid approach supports both efficient navigation and detailed environment understanding, which is essential for tasks like autonomous navigation and exploration.

In manipulation scenarios, the world is composed of separable objects. A hybrid of high- and low-level representations, object-centric models [KNK21] break the environment down into distinct objects or entities. There are different levels of object-centric representations, such as:

- ▶ *Point-level representations* (e.g. point clouds, pixels, or voxels) that capture basic spatial properties of objects.
- ▶ *Part-level representations* that break down objects into smaller components or features.
- ▶ *Object-level representations*, where entire objects are represented as discrete entities with associated properties and relations.

World models, together with planning mechanisms, form the backbone of advanced decision-making approaches such as model-based reinforcement learning (RL). However, learning accurate models from high-dimensional data (e.g., images or videos) is often challenging, time-consuming, and data-intensive. To effectively

abstract information, these models often integrate prior knowledge about object properties (e.g. geometry, material) and semantics, which is particularly useful in tasks such as robotic manipulation. Object-centric representations enable agents to generalize across tasks involving similar objects by learning the relationships between objects and their properties. Moreover, in real-world settings, agents often interact with multimodal sensory data, such as vision, audio, and touch. Multimodal representations integrate information from these diverse modalities into a unified latent space. By combining sensory inputs, agents can build richer, more informative models that support both high-level semantic understanding and low-level action planning. Incorporating semantic² information into scene representations enhances the agent’s ability to reason about the environment, retrieve object-related information, and plan actions effectively.

An LLM is sometimes a “secret” world model, with textual representations, that can *mental model* agent behaviors [Lu+25; Sch+25] and even forecast/reason about outcomes (in textual space) using built-in knowledge [Hao+23] (cfr. § 8.3 on page 127). This thesis investigates crossmodal predictive modeling (cfr. Chapter 4), and also explores abstracted world representations (cfr. Chapter 5 and Chapter 7) to support more efficient learning and planning at higher semantic levels.

Learning Representations

The acquisition of comprehensive world representations constitutes a fundamental prerequisite for autonomous agents to systematically accumulate environment knowledge. Central to this endeavor are methodological approaches, particularly *predictive modeling* and *cross-modal learning*, which facilitate the extraction of representative features from unprocessed sensory input through self-supervised learning paradigms.

Predictive Modeling. In the context of world models, predictive modeling is essential for agents to anticipate future states based on past executions and observations. A world model, at its core, represents the underlying dynamics of an environment, which can be modeled as either a deterministic function or a stochastic distribution, depending on the degree of uncertainty in the environment.

- *Deterministic models* predict future states s_{t+1} as a direct function of the current state s_t and action a_t , without considering uncertainty or noise in the environment. The world model in this case is a deterministic function f_θ , where θ represents the learned parameters:

$$\hat{s}_{t+1} = f_\theta(s_t, a_t)$$

This model assumes that the environment’s dynamics are fully predictable given the current state and action. Deterministic models are often used in controlled environments where randomness is minimal or can be ignored.

²: This integration will be further discussed soon in § 3.2.2 “Integration: Symbolic Grounding” on page 37

- *Stochastic models* take care of uncertainty [KG17; Mur22] where, in many real-world environments, the future state cannot be predicted correctly. In such cases, the world model is represented as a probabilistic distribution over possible future states. It can be often expressed as a conditional distribution $p(s_{t+1}|s_t, a_t)$, which describes the likelihood of transitioning to state s_{t+1} from state s_t under action a_t :

$$p(s_{t+1}|s_t, a_t) = \mathcal{N}(\mu_\theta(s_t, a_t), \Sigma)$$

Here, the future state is modeled as a Gaussian distribution³ around the predicted state, with variance Σ capturing the inherent uncertainty in the system. Stochastic models are particularly useful when the environment is noisy or unpredictable, as they allow agents to reason about the distribution of possible outcomes rather than a single deterministic prediction.

Crossmodal Learning. Crossmodal learning is a crucial technique for developing comprehensive world models, as it enables an agent to integrate and relate information from multiple sensory modalities, such as vision, touch, and sound. The aim is to learn a joint representation that captures the relationships between these modalities, allowing the agent to perceive and understand its environment in a more holistic way. By aligning and fusing data from different sensory sources into a shared latent space, the agent can predict one modality from another, which is especially useful in fields like multimodal fusion, transfer learning, and scenarios where one modality's data is sparse or difficult to obtain (e.g. predicting visual information from textual descriptions).

As a basis of the research works that will be introduced in later chapters, the following presents a Bayesian perspective on crossmodal learning, outlining the formulation and the mathematical framework for predicting one modality based on another.

- *Bayesian perspective.* In crossmodal learning, we aim to model the conditional probability distribution between different modalities, such that one modality can be predicted from another. Let $\mathcal{X}_1$ and $\mathcal{X}_2$ represent two different modalities. The primary objective is to learn the conditional probability distribution $p(x_2|x_1)$, which describes how modality x_2 can be predicted from modality x_1 . In a Bayesian framework, this can be formulated as:

$$p(x_2|x_1) = \int p(x_2|z)p(z|x_1) dz,$$

where $p(x_2|z)$ is the likelihood of modality x_2 given a latent representation z , $p(z|x_1)$ is the posterior distribution of the latent variable z given modality x_1 , and z represents the shared latent space between modalities $\mathcal{X}_1$ and $\mathcal{X}_2$. This integral can be approximated using variational methods if the true posterior

[KG17]: Kendall et al. (2017), 'What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?'

[Mur22]: Murphy (2022), *Probabilistic Machine Learning: An Introduction*

3: The Gaussian distribution is commonly used in stochastic models due to its mathematical convenience and well-understood properties. It allows for efficient learning and inference, especially when combined with neural networks that output the mean and the covariance.

$p(z|x_1)$ is intractable, leading to the use of a variational distribution $q_\phi(z|x_1)$ modeled by parameters ϕ , resulting in the prediction formula:

$$p(x_2|x_1) = \int p(x_2|z)q_\phi(z|x_1) dz.$$

The joint distribution between the two modalities and their shared latent representations can be expressed as

$$p(x_1, x_2, z) = p(x_1|z)p(x_2|z)p(z),$$

where $p(x_1|z)$ and $p(x_2|z)$ are the likelihood functions describing how the modalities are generated from the shared latent space, and $p(z)$ is the prior distribution over the latent variable, typically assumed to be a simple distribution like a Gaussian. The goal is to optimize the parameters of these functions such that the model accurately captures the crossmodal relationships.

- *Encoding & Decoding.* Given a learned latent space, the task of crossmodal prediction involves encoding one modality and decoding it to predict the other modality. In this framework:

- *Encoding.* Modality x_1 is encoded into the shared latent space using an encoding function $g_\phi^{(1)} : \mathcal{X}_1 \rightarrow \mathcal{Z}$, producing a latent representation $z_1 = g_\phi^{(1)}(x_1)$,
- *Decoding.* The latent representation z_1 is then decoded to predict modality x_2 using a decoding function $f_\psi^{(2)} : \mathcal{Z} \rightarrow \mathcal{X}_2$, resulting in a predicted value $\hat{x}_2 = f_\psi^{(2)}(z_1)$.

The loss function for this crossmodal prediction is typically a reconstruction loss⁴, such as mean squared error:

$$\min_{\psi, \phi} \mathbb{E}_{(x_1, x_2) \sim p(x_1, x_2)} \left[d \left(f_\psi^{(2)}(g_\phi^{(1)}(x_1)), x_2 \right) \right],$$

where $d(\cdot, \cdot)$ represents the $\mathbb{L}_2$ distance between the predicted $\hat{x}_2$ and the ground truth x_2 .

- *Variational Inference.* To approximate the posterior distribution $p(z|x_1)$ in crossmodal learning, variational inference is commonly applied. A variational distribution $q_\phi(z|x_1)$ is introduced to approximate the true posterior. The model is trained to maximize the Evidence Lower Bound (ELBO):

$$\max_{\phi} \mathbb{E}_{q_\phi(z|x_1)} [\log p(x_1|z) + \log p(x_2|z)] - D_{\text{KL}}(q_\phi(z|x_1) \parallel p(z)),$$

where the first term encourages reconstruction of both modalities x_1 and x_2 from the shared latent representation z , and the second term regularizes the encoder $q_\phi(z|x_1)$ to be close to the prior $p(z)$. This setup enables crossmodal

4: For example, later in [Chapter 4 “Sound Guides Representations and Explorations”](#) on page 55, we reconstruct *audio* from *vision*.

representation learning by using one modality to infer latent structure and jointly reconstruct both.

Numerous methods have been proposed for learning effective representations, yet many rely on handcrafted objectives or domain-specific assumptions that limit generalization across tasks and modalities. Self-supervised learning has emerged as a scalable alternative, showing strong potential in both unimodal and crossmodal settings. [Chapter 4](#) explores this direction by leveraging predictive audio-visual modeling to capture meaningful correlations in sensory data. In parallel, [Chapter 5](#) and [Chapter 7](#) investigate an alternative strategy centered on natural language representations, utilizing LLMs to encode environmental semantics and reason about context and potential actions. While differing in modality and abstraction level, these approaches share the goal of enabling more flexible and interpretable robotic behavior, addressing key challenges in bridging perception, representation, and decision-making.

3.1.3 Utilization of World Models

World models are generative models that capture both the static structure and dynamic evolution of an environment. By simulating the consequences of actions, they enable autonomous agents to predict the outcomes of their behavior. This predictive capability allows agents to plan and make decisions without requiring constant interaction with simulators or the real world [NVI+25], which is especially valuable for autonomous exploration in unfamiliar or hazardous environments.

Predictive Capability, Planning, and Exploration

World models enable agents to anticipate the future states of the environment based on their current actions, which is crucial for planning and decision-making during exploration. By simulating the effects of various actions via a world model, agents can optimize their exploration strategies, selecting those that maximize the likelihood of achieving exploration goals. This predictive ability reduces the need for trial-and-error interactions with the environment, ensuring safer and more effective exploration. Furthermore, world models can decouple physics simulation from perception, enabling efficient data synthesis—for example, a single physical interaction can generate numerous records under varying visual conditions [NVI+25].

Adaptation, Generalization, and Long-Term Autonomy

World models are inherently task-agnostic, making them versatile and applicable across various tasks and environments. This allows agents to adapt to uncertainty and continuously update their models as they acquire new information, ensuring they remain flexible and capable of responding to changes in the environment. Additionally, world models facilitate generalization, enabling agents to transfer their knowledge from one environment to another without starting from scratch. This

generalization, particularly from the ones trained with large-scale data, enhances the agent's ability to explore diverse environments efficiently, reducing the time and effort required to learn new tasks. Over time, as agents refine their world models through exploration, they become more autonomous, capable of handling more complex tasks and responding to novel scenarios without human intervention. This self-improvement cycle fosters long-term autonomy, empowering agents to operate independently and adapt to increasingly sophisticated environments.

Despite their advantages, world models face significant challenges. Prediction errors tend to accumulate over long horizons, undermining planning reliability. Their ability to generalize across diverse, dynamic environments is often limited by domain-specific assumptions and insufficient integration of multimodal data. Moreover, learning robust and transferable representations under partial observability and sensor noise remains difficult, requiring complementary strategies for effective long-term autonomy. As environmental complexity grows, world models must capture increasingly diverse dynamics and interactions, which can increase computational demands and risk overfitting.

Future progress is expected to focus on developing multimodal world models that integrate diverse sensory inputs. It is advisable to combine these with Vision-Language-Action Model (VLA) models to enable large-scale, model-based planning enriched with reasoning adaptability.

3.2 SEMANTICS

Semantics, a unifying symbolic medium for intelligence, serves as a bridge that connects perception, reasoning, and action, enabling AI systems to move beyond raw data processing toward meaningful generalization and decision-making. To understand its role, we can structure key works in semantics into foundational concepts and applications.

3.2.1 Abstract Concepts

At the core of semantics lies the ability to represent abstract concepts, *i.e.* ideas that are not tied to specific sensory inputs but rather emerge from structured relationships between entities. AI systems achieve this by embedding meaning into semantic spaces, where similar concepts are positioned closer together based on their contextual or relational similarities.

Semantic Spaces

Semantic spaces provide the foundation for how AI systems represent and process meaning. Early approaches such as Simple Recurrent Network (SRN) [Elm90] focused on learning temporal patterns in sequential data, while later developments like Word2Vec [Mik+13] introduced efficient methods for learning dense vector repre-

sentations of words based on co-occurrence statistics. Transformer-based models like BERT [Dev+19] and GPT [Rad+18] expanded this by incorporating contextual embeddings, allowing words to take on different meanings depending on their surrounding text. Beyond text, knowledge graphs [Hog+21] explicitly encode relationships between entities, enabling structured reasoning. Together, these approaches define how AI systems map raw data into a structured, abstract representation of meaning, forming the basis for higher-level reasoning and decision-making.

Alignment

A key challenge in AI is aligning perceptual data (such as images, video, and audio) with abstract semantic representations. This multimodal alignment allows AI systems to connect sensory inputs with meaningful concepts, enabling them to describe, reason about, and interact with the world in ways that are more akin to human understanding.

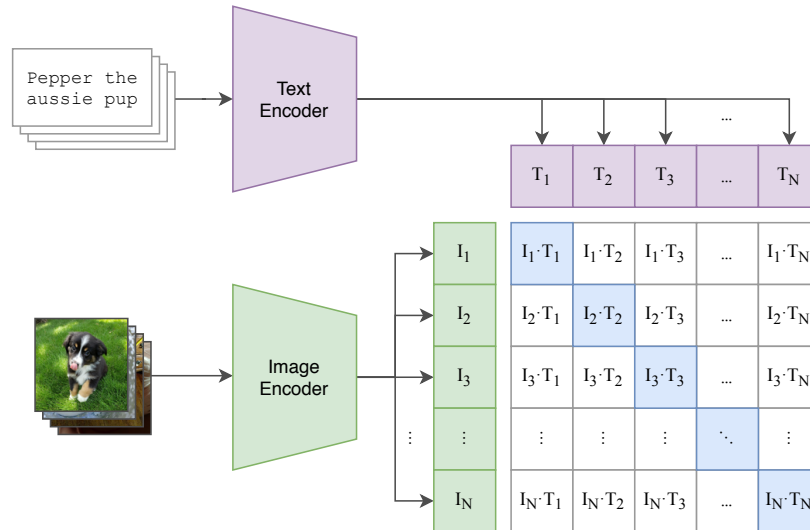


Figure 3.2: Overview of the Contrastive Language-Image Pretraining (CLIP) training objective. The model jointly trains an image encoder and text encoder to predict the correct pairings of images and their text descriptions. During pretraining, CLIP learns to maximize similarity between image-text pairs from the same example while minimizing similarity with other pairings in the batch. (Figure adapted from [Rad+21].)

Contrastive learning is one of the most influential approaches to align multimodalities, where models like CLIP [Rad+21] (see Figure 3.2) and BLIP [Li+22] map images and text into a shared latent space. These models use contrastive loss, an indirect learning loss function for classification, to minimize the distance between paired text and image inputs, enabling them to develop a joint embedding that represents both modalities in the same space. This alignment enables tasks like zero-shot image classification, where the model can recognize novel objects based on textual descriptions without additional training. Contrastive models leverage large-scale web data, making them highly scalable while reducing the dependency on manually labeled datasets.

[Rad+21]: Radford et al. (2021), ‘Learning Transferable Visual Models from Natural Language Supervision’

There also exist multimodal transformers [Sin+22; Che+23b] that go a step further, beyond mapping modalities into a shared space, by enabling the joint processing of different modalities, allowing for more complex fusions.

3.2.2 Integration: Symbolic Grounding

Autonomous agents require a structured understanding of their environment to make informed decisions. While world models enable predictive reasoning by capturing environment dynamics, semantics provide a structured representation of meaning that enhances decision-making. When integrated, these approaches create systems that can both predict *how* the world works and understand *what* things mean (cfr. Figure 3.3). Consider a mobile robot navigating with a semantic map versus a purely geometric one. With semantic understanding, the robot can interpret high-level goals (e.g., “get an apple” → “go to the kitchen”) rather than requiring specified low-level waypoints (e.g. coordinates). This semantic approach offers two key advantages: (1) enhanced interpretability for human operators and collaborators, and (2) improved robustness in scene generalization, as the robot can identify functional spaces across different environments despite visual variations.

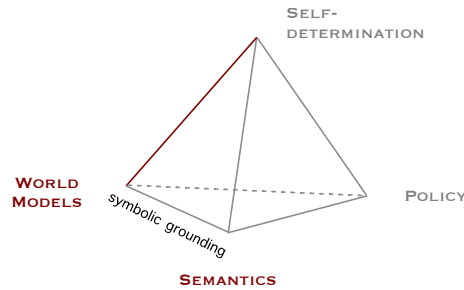


Figure 3.3: Symbolic Grounding = Semantics + World Models. Agents ground symbolic representations by linking abstract semantic concepts to their embodied world models, enabling meaningful interpretation and interaction with the environment.

Scene Graph

Scene graphs [Joh+15; Che+23a; Ran+23; Gu+24; Jia+24; Dai+24] provide a structured representation of visual scenes by organizing perceived elements into a graph-based format where nodes typically represent objects and edges capture relationships between these objects. This representation serves as a critical bridge between raw perceptual data processed by world models and higher-level semantic understanding. Scene graphs formalize visual understanding through several key components:

- ▶ *Objects*, which are discrete entities identified in the scene, often with associated attributes;
- ▶ *Relationships*, spatial (e.g. “inside”, “on”), functional (e.g. “containing”) connections between objects;
- ▶ *Attributes*, properties of objects such as color, size, material, or other state [Sun+24] (see also § 8.2 as an example of object-state sensitive planning).

This hierarchical data structure enables more sophisticated reasoning about visual

scenes than flat representations. Scene representations enriched with integrated semantics enhance informativeness and support a deeper understanding of the environment. They also enable robots to more effectively retrieve object-related information, facilitating subsequent planning or responding to human inquiries.

Semantic Affordance

Affordances are the opportunities for action provided by the environment, according to Gibson’s ecological psychology of development. Semantic affordances extend Gibson’s ecological theory [GP00] by integrating physical action possibilities with objects’ functional meanings, enabling intelligent systems to reason about purposeful environment interactions. This framework transcends physical capabilities by incorporating knowledge about object functions and cultural conventions, allowing systems to make nuanced decisions based on both physical properties and intended purposes. Recent research demonstrates that semantic affordance models facilitate the acquisition of new skills through environment interaction⁵ [ZWW24; HFS23; Rho+25; Li+25c], driven by semantic motivations rather than predetermined task specifications. The strength of semantic affordance lies in their bidirectional mapping between perceptual features and functional possibilities, grounding abstract concepts in physical interactions while elevating these interactions to meaningful, goal-directed behaviors, which is essential for developing adaptive intelligent systems that can continuously expand their behavioral repertoire based on semantic understanding of their environment.

3.3 POLICY

In robotics and RL, a *policy* refers to decision mapping, a function or rule that maps states to actions (what action to take in each situation). Built on top of this control foundation, agents develop versatile capabilities to fulfill various purposes.

3.3.1 Policy Function, Option, and Skill

Policy function is the lowest level of control concepts, among *option* (mid-level temporal abstraction) and *skill* (high-level functional capability).

Policy Function

A policy function is usually denoted as $\pi(a|s)$, mapping a state s to an action a (deterministic) or a distribution of action candidates (stochastic), a strategy under the MDP assumption. Furthermore, $a_t \perp s_{<t} \mid s_t$, i.e. $\pi(a_t \mid s_t) = \pi(a_t \mid s_{\leq t})$. In practice, to incorporate temporal information such as velocity, up to k previous

⁵: An example work of semantic affordances will be introduced in Chapter 7 “Agentic Skill Discovery” on page 99.

observations, $\{o_{t-(k-1)}, \dots, o_{t-1}, o_t\}$, are collected as input for state representation learning. Empirical evidence⁶ suggests that this representation learning with a concatenation of several history frames improves RL performance.

Action space. The action space for a manipulation robot can be either Cartesian space for end-effectors or joint space for responsive joint control. The former is easier to generalize across robot configurations but, however, requires an external motion planning backend. For LLM-based agents with textual environments⁷, the action space is in symbols or natural languages but usually linked with symbolic representations of external utilities such as robot skills⁸.

Option

An *option* is a formal construct used in Hierarchical Reinforcement Learning (HRL). It is defined as a triplet (*precondition*, *policy*, *post-condition*)⁹ [SPS99; SB18; KNK21; GRW17]:

- ▶ *Precondition*: The initiation set, specifying the states in which the option can be executed.
- ▶ *Policy*: A state-action mapping function that governs the agent's behavior while the option is active.
- ▶ *Post-condition*: Typically similar to the termination condition, defining when the option should conclude.

An option addresses the temporal abstraction in the pursuit of control reusability. Preconditions and post-conditions are key characteristics of option execution. Preconditions and postconditions are typically represented abstractly using symbolic propositions or predicates [KNK21; Hel14; Jia+19]. Reasoning over these symbolic representations enables a planner to search for solutions by ensuring that the post-condition of one skill satisfies the precondition of the next.

Skill

In embodied agent systems, a *skill* is a structured, reusable capability that enables an agent to interact with its environment in a goal-directed manner (*cfr.* Figure 3.4). Skills are often hierarchically composed, integrating low-level primitives into more abstract behaviors, and can be autonomously acquired through interaction, exploration, or reinforcement. They are typically parameterized for flexibility and vary in generality,

6: See [Bur+19a] for discussion and also Chapter 4 “Sound Guides Representations and Explorations” on page 55, where three consecutive frames are concatenated as visual input for neural networks.

7: A physical robot may also have a textual environment for decision making, where the surroundings and robot capabilities are abstracted in text.

8: In Chapter 5 “Interactive Multimodal Perception Using Large Language Models” on page 69, multimodal cues from the environment are fed to LLMs as natural language. The actions determined by the LLM decision module are then converted into symbolic representations of robot skill functions for execution mapping.

9: Some prefer the term *effect* over *post-condition*, as used in the Planning Domain Definition Language (PDDL). In this thesis, the two terms are used interchangeably.



Figure 3.4: Conceptual illustration of instructional control and planning with world models. Instructional Control = Policy + Semantics (left). Planning = Policy + World Models (right), *i.e.* planning can be viewed as the process of refining a policy (control) by leveraging a world model (typically in symbolic representation) to simulate and evaluate the possible future outcomes of its actions.

ranging from task-specific actions to broadly transferable competencies, forming the foundation for adaptive and intelligent behavior in embodied agents. Key characteristics of robot skills include:

- ▶ *Task-oriented*: Defined by their ability to achieve specific functional objectives.
- ▶ *Modular*: Self-contained units of functionality that can be developed, tested, and deployed independently.
- ▶ *Reusable*: Applicable across different scenarios, tasks, and potentially different robotic platforms.
- ▶ *Parameterized*: Configurable through adjustable parameters to adapt to varying conditions and requirements.

3.3.2 Planning and Learning

Policies define the decision-making processes that guide agents' actions within their environments. These policies can be developed through various methodologies, each offering unique advantages and suitable applications. Below is an overview of the primary approaches:

Task and Motion Planning

Planning-based methods (*cfr.* Figure 3.4) involve the creation of control policies through deliberate design and reasoning. Task and motion planning (TAMP) [SK16] is a framework in robotics and embodied agents that bridges the gap between high-level task planning and low-level motion control. It integrates two main components: task planning and motion planning, enabling an agent to plan and execute complex tasks in a dynamic environment.

Task Planning. Task planning focuses on determining the sequence of high-level actions that an agent should take to achieve a particular goal. It is usually built upon a controlled set of symbolic representations, where actions are abstracted and reasoned about (often, optimized by efficient searching algorithms). In task planning, a problem is typically defined by a set of states and actions, with the goal being to find a valid path from an initial state to a goal state. This is often done using formal languages, such as Planning Domain Definition Language (PDDL) [Hel14; Jia+19],

which defines actions, preconditions, and effects.

Motion Planning. Motion planning, on the other hand, deals with the low-level details of how an agent physically moves in its environment. It focuses on generating feasible trajectories that avoid collisions with obstacles while respecting the agent's physical constraints (e.g. velocity limits, and joint angles for a robot arm). It is often formulated as a path-finding problem in a continuous configuration space. See also § 2.1.1 “Environment Exploration” on page 9 for the introduction of path planning for map building and exploration.

TAMP systems, which unify both task and motion planning, are becoming more widely used in complex, real-world robotic tasks, such as household robots, warehouse automation, assembly lines, and autonomous vehicles, where both high-level decision-making and low-level movement must work together seamlessly.

Learning for Decision-Making

Traditional task and motion planning methods relied heavily on proper modeling of tasks and environments, being time-consuming for humans to abstract and implement. Learning-based methods derive policies from interactively explored data or human-collected demonstrations.

Reinforcement Learning (RL). RL enables agents to learn optimal behaviors by receiving feedback from their actions in the form of rewards or penalties (see Figure 3.5). Over time, agents develop policies that maximize cumulative rewards, effectively learning from trial and error. The agent interacts with the environment in discrete time steps. At each step t , the agent observes the current state s_t , takes an action a_t according to a policy $\pi(a_t|s_t)$, the environment transits to next state s_{t+1} and the agent receives reward r_{t+1} . The agent's objective is to maximize the *cumulative discounted reward*, or the *reward to go*, expressed as

$$R_t = \sum_{i=0}^{i=\infty} \gamma^i r_{t+1+i},$$

where $\gamma \in [0, 1)$ is the discount factor, trading off immediate and future rewards.

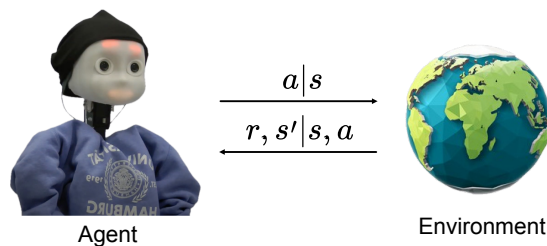


Figure 3.5: Illustration of the Reinforcement Learning (RL) paradigm, where an agent interacts with an environment through actions a , which are decisions based on the current state s . The environment then responds with a new state s' and a reward signal r . This feedback loop continues as the agent learns to maximize cumulative rewards over time by improving its policy for selecting actions in different states. The robot figure is adapted from <https://www.inf.uni-hamburg.de/en/inst/ab/wtm/research/neurobotics/nico.html>.

Typically, RL methods can be categorized as model-based and model-free approaches, with a major distinction of whether to learn and to utilize (in particular, planning with) a transition model.

- *Model-based RL methods* aim to learn a model of the environment's dynamics (cfr. § 3.1), which can be used to simulate future states and rewards, thus being useful for imagined planning. This approach can reduce the amount of interaction required with the environment, as the agent can simulate possible outcomes in an imagined space before really taking actions. Techniques like Monte Carlo Tree Search (MCTS) [Sil+16] and Model Predictive Control (MPC) [Ren+22; KL20] are examples of model-based methods.
- *Model-free RL methods* do not require the agent to explicitly model the environment's dynamics (i.e. it doesn't learn world models to predict how the environment transits between states). Instead, the agent directly learns a policy for optimal decision-making.

- *Value-based methods* aim to learn the value function for actions and derive the optimal policy from it. An example of a value-based method is Q-learning, where the Q-values, $Q(s, a)$, are iteratively updated using the Bellman equation, based on the reward feedback:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right].$$

- *Policy gradient methods* focus directly on learning the policy itself, where the policy is parameterized and updated in the direction of higher expected rewards:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) r_t \right],$$

where the expectation is taken over trajectories τ , sampled from the policy to learn.

- *Actor-Critic methods* combine both value-based and policy-based approaches. The “actor” learns the policy, while the “critic” evaluates the actions taken by the actor using value functions. This hybrid approach can improve learning efficiency. The policy gradient update in Actor-Critic is:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A^{\pi}(s_t, a_t) \right],$$

where $A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$ is the advantage function, which measures how much better action is compared to the expected value of a state, i.e. the state-value function $V^{\pi}(s_t) = \sum_a Q^{\pi}(s_t, a)$.

Besides various aforementioned optimization methods, learning process shaping is also an effective way to accelerate learning.

- *Reward shaping* accelerates the learning process by modifying the reward distribution, typically from sparsely distributed to densely distributed, such

that the agent can easily get positively rewarded while keeping the optimal policy unchanged. Additionally, certain types of reward shaping apply non-linear transformations to the reward signals to reduce the impact of noise and, thus, enhance learning stability [Li+23c]. See also § 8.1 “Reinforcement Learning with Derived Rewards” on page 119 for details.

- *Curriculum learning*, similarly, eases the learning challenge by presenting a sequence of tasks with gradually increasing difficulties, making the policy learning smooth [KNK21].

RL enables agents to learn through trial and error, typically in simulated environments. However, it is data-hungry, sampling-inefficient, and often unstable during training. Additionally, the sim-to-real transfer remains challenging due to gaps in both physics and sensing. While building realistic simulations and applying randomization techniques can improve generalizability and help bridge this gap (particularly for sensing), they cannot eliminate it entirely. As a result, learning from human demonstrations remains the primary method for efficiently training real-world robots.

Imitation Learning (IL). Also known as *learning from demonstrations*, with $\mathcal{D} = \{(s_i, a_i)\}_{i=1}^N$ denoting a dataset consisting of N state-action pairs collected from expert-demonstrated trajectories, IL is a type of machine learning where an agent learns to perform tasks by observing and mimicking the behavior of an expert. The goal is for the agent to replicate the expert’s actions, typically with the intent of solving complex tasks without the need for extensive trial-and-error learning as in RL. IL dramatically reduces exploration time and bridges a way to incorporate human preference.

- *Behavior Cloning (BC)* can be viewed as a supervised learning problem, where the agent learns a policy that maps states to actions by minimizing the difference between its predicted actions and the expert’s actions. The goal is to learn a policy $\pi_\theta(a|s)$ that approximates the expert’s behavior, typically with a loss function:

- for discrete actions

$$L_{BC}(\theta) = -\frac{1}{N} \sum_{i=1}^N \log \pi_\theta(a_i|s_i),$$

- and for continuous actions

$$L_{BC}(\theta) = \frac{1}{N} \sum_{i=1}^N \|\pi_\theta(s_i) - a_i\|_2^2.$$

- *Inverse Reinforcement Learning (IRL)* focuses on recovering the reward function $r(s, a)$ that the expert is implicitly optimizing, with an assumption that the expert trajectories are sampled with maximum rewards, rather than directly mimicking the expert’s actions. The idea is to infer the underlying reward signal from the expert’s demonstrated behavior, following which the objective can be

formulated as

$$\max_{\theta} \mathbb{E}_{\pi^*} \sum_{t=0}^T \gamma^t r_{\theta}(s_t, a_t),$$

where π^* is the expert policy. However, this is an ill-posed problem since many reward functions can explain the expert's behavior. This ambiguity is addressed using various regularization strategies or assumptions, depending on the specific IRL method [Abb08; Zie+08].

- *Adversarial Imitation Learning*. A min-max formulation arises when considering IRL as an adversarial game. For example, GAIL [HE16] expresses the objective function as

$$\mathcal{J}(\theta, \phi) = \min_{\theta} \max_{\phi} [\mathbb{E}_{\pi^*} \log p_{\phi}(e|s, a) + \mathbb{E}_{\pi_{\theta}} \log(1 - p_{\phi}(e|s, a))],$$

where $p_{\phi}(e|s, a)$ is a learned discriminator to distinguish expert trajectories, denoted as e , from generated ones.

- *Diffusion Policy*. Diffusion policies use Denoising Diffusion Probabilistic Models (DDPMs) to generate actions iteratively. The objective function is based on reconstructing expert actions from noisy versions, similar to how diffusion models are trained for image generation. Mathematically, instead of directly modeling $p(a|s)$, diffusion policies approximate it via iterative denoising $p_{\theta}(a|s) = \int p_{\theta}(a|s, z)p(z|s)dz$.

While planning and learning represent two fundamental pillars of autonomous decision-making, their integration remains a core challenge. Planning methods offer structure and foresight but often rely on accurate models and handcrafted representations. Learning-based approaches, particularly those using reinforcement or self-supervised signals, provide adaptability but can suffer from sample inefficiency and instability. In relatively simple hierarchical robotic systems *e.g.* SayCan [Ahn+22] and subsequent works [Zha+23c; Ran+23], high-level and low-level functionalities are often modularized, with the high level handling strategic decision-making and the low level responsible for precise control. In such architectures, symbolic planning and learning-based methods can be selectively applied at different levels to meet specific requirements. Closing the gap between planning and learning, whether by incorporating learned representations into planners or embedding planning structures into learning algorithms, offers a promising direction for developing more robust and generalizable robotic policies.

3.3.3 Integration: Planning and Learning with Foundation Models

Rather than relying on exhaustive human demonstrations, a growing trend is to leverage the knowledge in large foundation models to make high-level decisions or guide learning-based methods. In recent years, foundation models, particularly LLMs, have revolutionized the way autonomous systems approach *planning and learning as skills* (cfr. Figure 3.6). These models, pre-trained on vast amounts of diverse

[HE16]: Ho et al. (2016), 'Generative Adversarial Imitation Learning'

symbolic definition languages, such as PDDL, along with classical planners, support LLMs in formulating problems and assessing the feasibility of long-term tasks. For example, LLM+P [Liu+23] and many similar works [Din+23; Lia+22; Chu+25] convert a language description of a problem into a Planning Domain Definition Language (PDDL) file, and then resort to classical symbolic planners to compute a solution, after which the resulting plan is translated into actions with minimal effort of mapping (see Figure 3.7). This neuro-symbolic integration benefits from the complementary strengths of LLMs (flexibility and world knowledge) and symbolic planners (logical consistency and guarantees).



Figure 3.7: Symbolic planning with PDDL configurations generated by LLMs. This approach uses LLMs to create formal planning specifications, bridging natural language understanding with classical planning frameworks. § 8.2 “LLM-based Embodied Planning” on page 124 will introduce an approach where LLM is used to generate PDDL configurations for symbolic multi-agent planning.

A key advantage of symbolic planning approaches is their interpretability: researchers and users can inspect the generated plans, identify potential issues, and understand the agent’s reasoning process. This transparency becomes particularly valuable in safety-critical applications where explainability is essential. However, these approaches often face challenges in precise abstraction and handling the ambiguity inherent in real-world scenarios, and may struggle with the computational overhead of formal planning in complex domains.

Guided Learning

LLM-guided learning represents a paradigm shift in how, typically, RL agents acquire and refine their capabilities, with LLM serving as knowledge-rich guides throughout the learning process. Learning-based approaches focus on improving agent planning capabilities through various forms of feedback and experience. Reinforcement Learning from Human Feedback (RLHF) has emerged as a cornerstone methodology demonstrating how human annotated trajectories (in the form of paired preference) can deduce a reward model and thus be applied to train robot policies [Chr+17].

Soon RLHF became a core LLM alignment tuning method, leading to various studies, among which, Reinforcement Learning from AI Feedback (RLAIF) [Li+23a] releases the burden of human annotation with auto-generated or LLM-annotated data. In addition to training LLM agents, the training of robotic agents also benefits from AI feedback, such as guidance from LLMs or Vision Language Models (VLMs). The applications may range from motivation to regulation, with outcome or process guidance, which has been discussed in § 3.4 “Self-Determination” on the facing page and § 8.1.2 “Reinforcement Learning with Deductive Rewards” on page 123.

Vision-Language-Action Models

Vision-Language-Action Models (VLAs) [Bro+23; ONe+24; Zit+23; Tea25] represent the frontier of embodied intelligence research by creating end-to-end architectures that process visual inputs, understand language instructions, and generate appropriate actions. Robotic Transformer (RT-1) [Bro+23] and Robotic Transformer 2 (RT-2) [Zit+23] demonstrated how transformer-based architectures can learn mappings from visual observations and language commands directly to robotic actions through large-scale training on demonstration data. Recently, Gemini Robotics [Tea25] is capable of executing smooth and reactive movements to cover a great range of long-horizon, dextrous manipulation tasks. The integration of multiple modalities enables VLAs to ground language understanding in visual perception, addressing a fundamental challenge in robotics and embodied AI. Despite their impressive capabilities, VLAs face significant challenges in sample efficiency, often requiring massive datasets of demonstrations to learn effective policies. Additionally, these systems must contend with the inherent complexity of real-world visual scenes and the physical constraints of embodied action, making them among the most challenging but potentially transformative approaches in the field.

3.4 SELF-DETERMINATION

Self-determination originally refers to the ability of an individual or group to make choices free from external coercion¹⁰. In psychology, Self-Determination Theory (SDT) [RD00] further refines this idea as the capacity to act autonomously, driven by intrinsic motivation and the satisfaction of basic psychological needs such as autonomy, competence, and relatedness, and maintained through processes of self-regulation.

When applied to robotics, self-determination takes on a complementary yet distinct meaning. Here it translates into the design of systems that can independently govern their behavior, self-regulate, and optimize their performance through adaptive control and learning. While *intrinsic motivation* has been widely studied in the context of learning-based exploration [Pat+17; LP23], this section, with a stronger emphasis on robotic *self-determination*, complements the discussion by highlighting the importance of both *intrinsic motivation* and *self-regulation* [RKD97; RD00]. The latter is usually implicitly considered with constructing traditional exploration strategies, but its importance should not be overlooked especially when building *agentic entities* [Qia+24; Seq24; Sha+23] where the criteria for assessing whether agents behave well along the self-motivated goals cannot be always crafted by humans beforehand¹¹.

¹⁰: A highly related but distinguishable term to “self-determination” is “agency”, which emphasizes the power to act as a causal agent and can be seen as a component or prerequisite of self-determination. The self-determination discussed in this thesis emphasizes intrinsic motivation and self-regulation. See also definition in dictionary: <https://www.merriam-webster.com/dictionary/self-determination>

¹¹: As a concrete example, later in Chapter 7 “Agentic Skill Discovery” on page 99, the skills are proposed as intrinsically motivated goals, but they also require a mechanism for verification.

3.4.1 Intrinsic Motivation

Intrinsic motivation refers to the drive to engage in an activity for its inherent satisfaction rather than for separable outcomes, *i.e.* external rewards, suggesting that intrinsic motivation is a natural inclination or tendency that organisms have [RD00]. It is not directly caused by external factors, but rather it is triggered or enhanced when individuals find themselves in environments or conditions that encourage or support this motivation to be expressed¹². In humans, intrinsic motivation is linked to curiosity, mastery, and a sense of autonomy, as described in SDT. It fuels learning and exploration by prompting individuals to seek out novel experiences, solve problems, and improve their skills. In robotics, intrinsic motivation plays a crucial role in enabling autonomous agents to explore and interact with their environment beyond predefined tasks. Instead of relying solely on externally programmed goals, intrinsically motivated robots generate their own objectives based on factors like novelty, surprise, or empowerment. For example, an exploration-driven robot may prioritize areas with high uncertainty or information gain, leading to better world modeling and decision-making. Such mechanisms are essential for lifelong learning, where a robot continuously refines its knowledge and adapts to changing environments.

Intrinsic motivation is the internal drive that encourages an agent to explore and learn for its own sake, independent of any external incentive (*e.g.* task-related rewards). We can capture this concept mathematically by introducing an internal *interest function*¹³, denoted by $\mathcal{J}(s, a, s')$, which, if taking task-agnostic world modeling surprise as an example of incentive, *i.e.* a transition is considered interesting if it leads to a significant prediction error in the agent's internal model of the environment, regardless of any specific external task, quantifies how inherently valuable, novel, or informative a transition (s, a, s') is to the agent. One way to formalize the agent's intrinsic drive is to assume that its action choice for preferred state evolution is governed by the gradient $\nabla \mathcal{J}$. Humans adjust behavior based on experience, aligning more with gradient-like updates. However, defining a smooth yet differentiable function that accurately captures intrinsic motivation is non-trivial.

While psychology defines intrinsic motivation as behavior driven by inherent satisfaction and autonomous self-determination, RL adapts this concept through "intrinsic rewards". They are computational signals that encourage exploration independent of task-specific goals (but are still "extrinsic" in principle, due to the fact that they are assigned by a human). RL does not require differentiating a known loss function, often a discrete reward function $r(s, a)$ is sufficient, formulating intrinsic motivation as an *intrinsic reward* for RL to maximize is a common choice with replacing $r^{\text{intr}}(\cdot)$ with $\mathcal{J}(\cdot)$, resulting in various optimization approximation of $\nabla \mathcal{J}$ according to the RL

¹²: Further, extrinsic motivation can be further classified as: external regulation, introjection, identification, and integration, organized to reflect their differing degrees of autonomy.

¹³: The term "interest" is adopted here, instead of just using "intrinsic reward" or "curiosity" from the RL exploration literature [Bur+19a], to represent a broader perspective of intrinsic motivation. It is a generic measure of how engaging or meaningful a transition (*i.e.* a change of the agent state resulting from an action it takes) is, without being required to specify why something is interesting, while curiosity is usually linked to seeking information or reducing uncertainty and mostly constrained within RL paradigm in a form of scalar rewards.

methods applied. In the context of RL, there are many works defining $r^{\text{intr}}(s, a, s')$:

Modeling uncertainty as motivation defines intrinsic reward as the negative likelihood of the environment dynamics model, *i.e.* $r^{\text{intr}} \propto -\log p_{\theta}(s'|s, a)$ where θ parameterizes a model of transitions (s, a, s') ¹⁴. Intuitively, the high likelihood of new states to come indicates a good quality of modeling the dynamics. Rewarding oppositely, *i.e.* favoring higher prediction errors, motivates the agent to explore states where its model of the world, represented by $(s, a) \rightarrow s'$, is inaccurate, driving curiosity and exploration to reduce the prediction error. However, maximizing uncertainty alone can be misleading. A well-known example is the flickering TV environment, where the screen displays random noise regardless of the agent's actions. An agent driven purely by uncertainty-based rewards may be attracted to such inherently unpredictable dynamics, despite their lack of meaningful structure or learnable value. To address this, some approaches model inverse dynamics $(s, s') \rightarrow a$ [Pat+17], others [Sch+22] go beyond prediction error and explicitly derive the information gain between model parameters and expected transitions. Such formulations encourage exploration that is not only uncertain but also informative, better supporting sample-efficient learning in model-based RL, especially for robotics.

Environment morphology as motivation includes a series of methods that quantify the discovery of structural environment properties as a reward [LP23]. For example, visibility counting as reward introduces measures to count visited states [Mar+17; Li+23c; Lu+22], usually assigning intrinsic rewards as $r^{\text{intr}}(s, a) = 1/\sqrt{N(s)}$, where $N(s)$ accounts to the number of times the agent has visited state s so far¹⁵. Intuitively, this intrinsic reward encourages the exploration of novel states whose $N(s)$ are sufficiently small and produce higher rewards. Another representative method, frontier-based exploration [Yam97] also evaluates explored and unexplored areas to encourage the agent to achieve greater coverage of the environment.

Empowerment as motivation refers to the kind of methods that maximize the influence an agent can have over its future states, *i.e.* skills emerge as structured ways to maximize control over future states¹⁶. Mathematically, empowerment is defined as the mutual information between the agent's action a_t and its future state after k steps, denoted as $I(s_{t+k}; a_t)$. By introducing a latent control variable z , which is uniformly sampled, the mutual information $I(\tau; z)$ between the trajectory τ (often simplified as s_0, s_T for simplicity and robustness) and the control variable z can be used as an intrinsic reward. This formulation encourages the discovery of diverse skills [GRW17; Wan+21; Eys+19; Las+21a; BI20; Las+21b], thereby promoting exploration. See Figure 3.8 (left) and *cfr.* § 2.1.2 “Autonomous Adaptation” on page 15 for discussion on adaptive empowerment.

14: In practice, the *forward* prediction error of the learned dynamics, $r^{\text{intr}/f} = \|s' - \hat{s}'\|$ (or the *backward* prediction error, $r^{\text{intr}/b} = \|a - \hat{a}\|$ when modeling $(s, s) \rightarrow a$), usually serve as a measure of uncertainty [Pat+17; Bur+19a], which can be interpreted under Gaussian distribution assumption $p_{\theta}(s'|s, a) = \mathcal{N}(s'|\mu_{\theta}(s, a); \Sigma)$. Derivations can be found in Appendix A “Prediction Error and Gaussian Modeling” on page 138.

15: For continuous state setting, state abstraction or pseudo-count, *e.g.* $N(s) \propto \frac{1}{\rho(s)}$ where $\rho(\cdot)$ is a density function, is often used to estimate the number of visits to a state.

16: Skills are learned control strategies that reliably lead to predictable and desirable outcomes. In this sense, skill learning can be framed as the process of increasing empowerment over time.

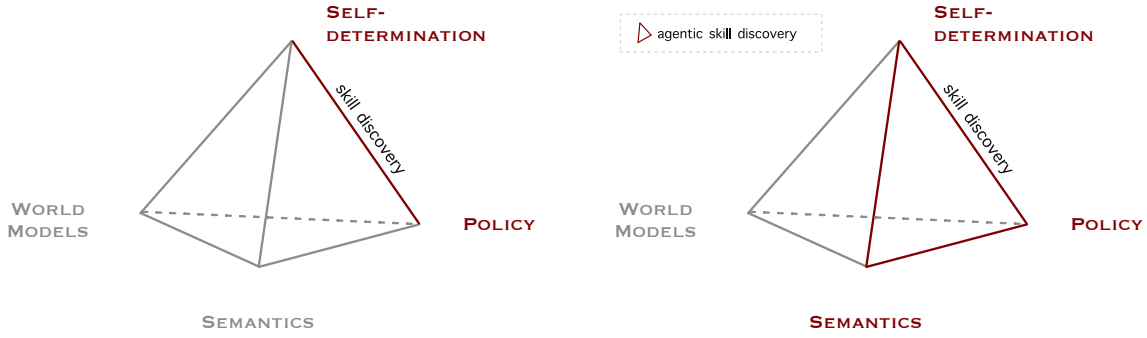


Figure 3.8: (Left) Skill Discovery = Self-determination + Policy, *i.e.* traditional skill discovery focuses on combining intrinsic motivation (first phase of self-determination) to shape policy learning. (Right) Agentic Skill Discovery = Self-determination + Semantics + Policy. Agentic Skill Discovery (ASD) extends this by incorporating semantics, enabling agents to ground learned skills in meaningful representations and goal structures. It integrates internal motivation, semantic understanding, and policy learning to support autonomous skill acquisition in novel environments, guided by LLMs that interpret and shape the learning process.

Semantic motivation represents a recent advancement in the LLM era, driven by the powerful reasoning and in-context learning capabilities of contemporary LLMs. When integrated with environment morphology, a *scene graph* can be established and motivate an agent to explore the environment assets with semantic guidance [Jia+24]. Given the basic information of environment and agent setting as context, the employed LLM is supposed to propose *interesting* [Cel+23] *yet meaningful* [Wan+24a; ZWW24; Ma+24a] *semantic goals* for the agent to achieve, or, in other words, new abilities to pursue via determining *semantic affordances*. As discussed earlier in § 3.2.2 “Integration: Symbolic Grounding” on page 37 and further elaborated in Chapter 7 “Agentic Skill Discovery” on page 99, where robots are semantically motivated to acquire new abilities grounded in their environments, the emerging trend of leveraging LLMs to guide agent learning shows great promise due to the large-scale knowledge embedded in LLMs. Agents empowered with LLM-driven self-determination can better understand and reason about the world, and formulate efficient, interpretable motivations to support autonomous adaptation.

3.4.2 Self-Regulation

Self-regulation, often overlooked in earlier, less agentic systems where learning was entirely controlled by humans through predefined criteria, is the agent’s ability to monitor, evaluate, and adjust its learning process to achieve desired outcomes, particularly in alignment with intrinsic motivations. For humans, it allows individuals to stay focused, resist distractions, and modify their actions based on feedback. For robots, self-regulation translates into adaptive determination mechanisms that ensure learning proceeds correctly. Autonomous agents continuously assess their own performance through built-in sensors and error metrics. This self-monitoring allows the system to detect failures, gauge success, and identify areas for improvement without relying solely on pre-programmed responses.

Being complementary to setting goals to motivate agents, self-regulation focuses on assessing whether agents are on the right track of learning. A self-regulating

robotic system must evaluate its internal states, predict future conditions, and adjust its learning accordingly. In this sense, a *skill* can be viewed as the minimal unit of learning that necessitates self-regulation, *i.e.* the awareness and ability to evaluate whether the outcomes of an agent’s actions align with predefined goals. Traditional works, *e.g.* RL with intrinsic motivation, usually regulate agents with minimal effort by manually examining the designed reward or loss function, neglecting a deep discussion into self-regulation with open purposes. For more complex systems with hierarchical execution structures, goals can exist at various levels, ranging from high-level semantic objectives to low-level vision-conditioned control tasks. With the development of, especially LLM-based, agentic systems, it is not even possible for humans to exhaustively supervise such complicated, sometimes multi-agent, systems. Therefore, an explicit higher level of autonomous yet robust regulation design is essential for the success of the agency. See [Chapter 7](#) for a detailed discussion on the self-regulation of agentic systems (*cfr.* [Figure 3.8](#)) (right), where LLMs are employed to supervise the learning process and outcomes of RL agents.

Process Supervision vs Outcome Supervision

Given a goal, the health of the learning status can be evaluated either densely, during the ongoing *process*, or sparsely, based on the final *outcome*, depending on the availability of supervision signals. For learning-based exploration, this reward signal is usually predefined as reward functions, *e.g.* prediction error as a reward in curiosity-driven exploration. For the former dense assessment, a per-step supervised signal is usually accessible to guide the learning process, while the sparse one indicates a post-assessment of the resultant behaviors. In RL for a specified task learning, this exhibits as dense rewards, where the agent is likely to pursue the reward frequently, or as sparse rewards, wherein the agent can only receive rewards upon occasional task completion. In the context of LLM training, this distinction usually leads to the application of the Process-supervised Reward Model (PRM) or Outcome-supervised Reward Model (ORM) [Lig+24]. If reliable, the reward model can be applied to scaling LLM inference. Usually, assessing an outcome is much easier than assessing the whole process¹⁷. Complex agentic systems must be evaluated across diverse subtasks, rendering process supervision of each learning component intractable. Even outcome-level assessment (*i.e.* task verification) poses a significant bottleneck, particularly for LLM-based agents (see [Chu+24b] for an example of process supervision, and [Cem+25] for observed failure cases in LLM agents).

¹⁷: Imagine verifying an LLM-generated solution, which can be considered a subtask within a broader agentic workflow for a given math problem. *Process supervision* involves evaluating each reasoning step, while *outcome supervision* only requires checking the final extracted answer. In [Chapter 6 “Enhancing Reasoning via Logic-Guided Inference Scaling”](#) on page 83, a method proposed to carry out per-step verification for LLM inference is discussed in detail. A detailed discussion on process (fast) and outcome (slow) self-regulation can be found later in [Chapter 7 “Agentic Skill Discovery”](#) on page 99, where LLMs and VLMs are tasked to regulate RL learning process and outcome respectively.

Termination

Appropriate termination matters much in learning-based approaches, by which the agent passively (*e.g.* because of encountering irreversible bad conditions) or actively (*e.g.* with explored information being sufficient enough for task requirement, active termination saves unnecessary further effort¹⁸) terminates the ongoing learning process in time and assigns a success or failure status for the happened learning. *Terminating in time.* In RL, a terminating signal can be regarded as a special sparse reward that helps learning with implicit reward shaping [Bur+19a]. As an example, the strategies of terminating ASLAM exploration are when reaching the following conditions [LLA21]:

- ▶ Map Completeness, sufficient map construction with no obvious unexplored areas (*e.g.* measured by frontiers [Yam97] or certain information gain/uncertainty) remain.
- ▶ Resource Constraints, upon reaching limits in resources such as battery life, traveled distance, elapsed time, or computational capacity.

Assigning outcome. At the end of each trajectory, a success or a failure status is assigned, with which the success rate can be computed. Since success rate has been a very important measure for robot performance, it also serves as a fitness function for, for example, evolutionary search approaches to find the best-shaped reward functions [Ma+24b]. In semantic environments, *e.g.* LLM reasoning¹⁹, the outcome is usually more context-rich and can also be fed back to the agent to further refine its decisions [LBS23; Zha+24c; ZWW24; Ma+24b]. For example, a code-generation LLM can adjust its outputs based on real-time feedback from an Integrated Development Environment (IDE).

With the development of LLMs and VLMs, these large-scale models can be leveraged to regulate an agent's timely termination and assign appropriate outcomes [Ma+24b; ZWW24]. By integrating multimodal reasoning capabilities, VLMs enable the agent to interpret environment cues, assess task completion, and determine when to terminate execution. Furthermore, VLMs facilitate outcome assignment by associating observed states with predefined criteria, ensuring coherent decision-making. This approach enhances the adaptability and efficiency of autonomous systems operating in dynamic environments.

3.4.3 Integrations

Self-determination, when combined with different foundational concepts such as world models (§ 3.1), semantics (§ 3.2), and policy (§ 3.3), can lead to distinct research directions. For example, interactive scene graphs [Jia+24] (see Figure 3.9 left) emerge at the intersection of self-determination, world models, and semantics, enabling agents to structure and interpret dynamic environments. Similarly, model-based exploration

¹⁸: This is common in the active perception field where “terminate” is a special action of an agent, resulting in non-fixed steps for each learning episode (see an example of this design in [Li+23c]).

¹⁹: Formulating the LLM generation process as an MDP results in prompts being the initial state, LLM output tokens as actions, and the context so far as the current state.

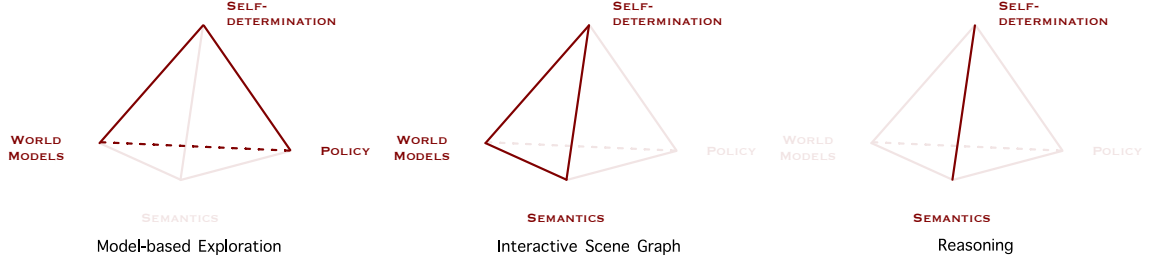


Figure 3.9: From left to right: (*Left*) Model-based Exploration = Self-determination + World Models + Policy; Agents explore their environment by leveraging internal motivations (self-determination), learned world models, and policies that guide action selection to reduce uncertainty and discover novel states. (*Middle*) Interactive Scene Graph = Self-determination + World Models + Semantics; Agents build and update structured, interpretable representations of their environment by grounding semantic concepts in perceptual models and selectively interacting with relevant entities. (*Right*) Reasoning = Self-determination + Semantics; Advanced reasoning emerges as agents use semantic knowledge to internally simulate, compare, and select among possible actions or explanations, driven by their own goals or queries.

[Moe+23; LP23] (see [Figure 3.9 middle](#)) integrates self-determination, world models, and policy to guide strategic decision-making in uncertain environments.

In this context, self-determination and semantics together define *reasoning* [Wei+22a; YZW23; Zha+24c; Heb+24; Dee+25] (see [Figure 3.9 right](#)), which can be regarded as an exploration occurring solely within the realm of natural language space. This semantic exploration enables agents to traverse conceptual spaces, infer logical relationships, and refine their understanding through structured problem-solving and counterfactual analysis. Discussions on semantic reasoning will be further elaborated later in [Chapter 6 on page 83](#).

Furthermore, Agentic Skill Discovery (ASD) [ZWW24; Rho+25; Li+25c], which will also be detailed later in [Chapter 7 on page 99](#), emerges from the combination of self-determination, semantics, and policy learning, allowing agents to autonomously explore semantic affordances and develop their capabilities.

An overview of how contributions introduced in later chapters align with the conceptual foundations is provided in [Table 3.1](#).

Table 3.1: Systematic overview of core contributions and their integration into conceptual foundations.

Ch.	World Models	Semantics	Policy	Self-determination
Chapter 4	●	○	◐	●
Chapter 5	◐	●	○	●
Chapter 6	○	●	○	●
Chapter 7	◐	●	●	●
Chapter 8	◐	●	◐	◐

Part II

CORE CONTRIBUTIONS

SOUND GUIDES REPRESENTATIONS AND EXPLORATIONS

4

Self-determination + World Models + Policy

To fulfill **Objective I**, i.e. “to construct self-deterministic agents that can leverage non-verbal multimodal cues to autonomously explore the environment and develop possible abilities beyond immediate task requirements”, this chapter proposes a model-based exploration strategy grounded in visual-auditory associations.

Multimodal cues offer complementary information that helps disambiguate goals and dynamics in complex settings. In particular, impact sounds, as salient indicators of physical interaction, serve as an internal signal for meaningful change. These auditory events are treated as sources of intrinsic motivation, the first aspect of *self-determination*, enabling the agent to assign value to exploratory behavior even in the absence of external rewards. By learning to predict and seek out these informative cues, the agent constructs richer *world models* that capture both visual and auditory consequences of its actions. These models, in turn, guide the *development of policies* that are not only goal-directed but also proactive in exploring known knowledge boundaries, resulting in generalizable representations and policies.

Cross-modal learning is essential for developing representations that are both meaningful and invariant to variations across modalities, particularly when handling noisy or incomplete data. While vision is the most commonly used modality for perception in autonomous agents, it can be ambiguous or insufficient in certain contexts, e.g. inferring object interactions or detecting events outside the field of view. In such cases, sound provides a valuable complementary signal, offering information about physical events like collisions, drops, or movements. Moreover, sound is naturally abundant in the real world and can be captured without direct contact, using small, low-cost microphones that are easily integrated into mobile platforms. These properties make it a practical and informative modality to support environment understanding and guide exploration beyond what is visible.

Although deep learning has shown strong capabilities in extracting information from multiple sensory modalities, sound remains underutilized in robotic manipulation learning. This chapter explores this potential to enhance autonomous exploration and representation learning within the framework of Unsupervised Reinforcement Learning (URL), where agents are expected to actively collect experiences and jointly learn representations and policies in a self-supervised manner. Specifically, this chapter introduces a framework for constructing realistic robotic manipulation scenarios that incorporate physics-based sound simulation¹, alongside a multimodal Reinforcement Learning (RL) exploration approach termed **Intrinsic Sound Curiosity Module** (ISCM). Experiments, with sound enabled during pre-training and disabled

1: See [Appendix B “Impact Sound Simulation”](#) on page 139 for details on sound simulation.

during adaptation, show that representations learned by ISCM outperform the ones by vision-only baselines, and pre-trained policies can accelerate the learning process when applied to downstream tasks.

4.1 INTRODUCTION

Research in the field of neuroscience shows that with multiple cues from a diverse range of sensory modalities comes enhanced behavioral performance towards faster response, more accurate movement, and a better sense of stimulus [Lau+04]. When presented with multiple modalities, *e.g.* a combination of auditory, haptic, and visual perception, an observer will make the *assumption of unity* that decides whether the multimodal information originates from a common source or from some separated objects and events [WW80]. The perception of unity arises when the perceiver assumes that a physical event is redundantly expressed and sensed across diverse modalities, and decisions are commonly made based on the temporal and spatial consistency of information [VS07], or on semantic congruence factors [Lau+04].

Vision is an exceptionally information-rich modality and one of the most critical senses through which humans perceive the world. However, it remains challenging for robots to directly extract structured knowledge from visual input. Although deep neural networks have significantly improved the quality of visual representations, such representations often remain difficult to interpret. When agents rely solely on these learned features, the limited scope of the information they capture can constrain generalization and restrict the range of tasks the agent can perform. For many vision-based tasks, a common approach begins by constructing neural networks using pre-trained models or training them in a self-supervised manner. This is often done through intra-modal objectives, such as designing simple but diverse sub-tasks within the visual domain [DZ17]. In contrast, crossmodal learning approaches, *e.g.* predicting the consistency between visual and auditory signals [Zha+18; AZ18], go beyond pure vision and are better suited for preserving the assumption of sensory unity, where different modalities provide coherent information about shared underlying causes.

These two components are tightly coupled: stable and informative representations are crucial for effective policy learning [Bur+19a], while a sufficiently exploratory policy is necessary to collect diverse, non-trivial observations. Humans can benefit from multiple sensing cues in terms of both perception and behavior. Intuitively, an active agent who is allowed to explore freely can benefit from multimodal cues in two aspects: 1) learning meaningful representations by crossmodal self-supervision [Eis+21; Hig+20; Par+18], and 2) being intrinsically motivated to explore the environment under the unity assumption reflected by the uncertainty of crossmodal predictions.

Sounds are generally much more distinctive compared with visual events. For some specific tasks related to physical properties estimation, the sound alone is reliable to guide a robot and measure its performance [Cla+18]. For others, it may be informative but not sufficient, *e.g.* a classification of objects that share common auditory properties [Mir+21], or precise control of a water-pouring robot [Lia+20]. In this case, sounds are supposed to fuse with other sensory inputs to present a much more robust

description of states, or to scaffold the agent’s exploration.

Sound is abundantly while hardly considered for general manipulations due to the facts that 1) vision is content-rich and is thus sufficient for traditional planning-based robots so the sound is often ignored; 2) the correlation of sound events with a task goal could be difficult to program or to discover automatically by traditional methods, which further limits its exploitation. However, things go the other way when a deep reinforcement learner is deployed to control. 1) Relying exclusively on vision may lead to exhaustive sampling requirements. Though deep neural networks are capable of extracting features from high-dimensional inputs, there is no guarantee of information sufficiency as samples are collected gradually. Representations can overfit to the trajectories of a non-optimal agent, especially when transferred to new scenes, where a biased policy could lead to a worse learning process. Moreover, exploration time for robots is often desired to be minimal for natural wear and safety concerns, which calls for efficient and robust pixel interpretation. 2) Fortunately, latent associations among modalities [Jae+21; Kum+19] and behavior consequences [Sil+21] can be discovered automatically by deep learning, which shows the potential of crossmodal control.

Therefore, our approach contains two phases: first, to train the image encoder of a RL agent with visual-auditory correlations, and second, to use the crossmodal error as an intrinsic reward to encourage meaningful exploration. Contributions in this chapter include:

- ▶ The ManipulateSound² environment built upon the ThreeDWorld simulator [Gan+21], detailed in § 2.3.1 “ThreeDWorld” on page 20, that comprises robotic control with physically generated sound (see Figure 4.1).
- ▶ A general architecture to utilize sound feedback for unsupervised RL exploration, resulting in more robust representation and active exploration.

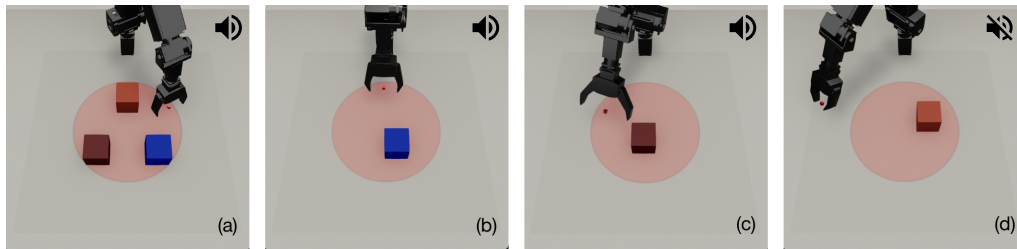


Figure 4.1: *ManipulateSound* environments with different objects that have different physical properties: (a) a task with *three different colorful cubes* to push out; (b) a task with a *single blue ceramic cube* to push out; (c) a task with a *single brown wooden cube* to push out; (d) a task with a *single red metal cube* to push out; sound intentionally turned off during evaluation.

4.2 RELATED WORK

We introduce sound as a means to enhance both self-supervised representation learning and the active exploration of URL agents. While the use of impact sound to guide representation and exploration is a novel integration, the individual components

2: See code link: <https://github.com/xf-zhao/ManipulateSound>

have been studied previously. In the following, we briefly review related work in both areas.

4.2.1 Self-supervised Representation Learning

Self-supervised learning covers methods to learn representations from data that contains automatically created pseudo-labels according to certain objectives. Based on the sensory inputs, self-supervised learning can be roughly classified into two categories: intra-modal and crossmodal self-supervised learning.

A common intra-modal way to create pseudo-labels of images is to perform multiple parameterized augmentations. Then, neural networks are trained to predict which transformation has been carried out on each sample [DZ17; ACM15]. Generally, representations learned with transformations that align with realistic physics make more sense to a robot than random ones. For instance, to obtain representations with ego-motion equivariance addressed, images are collected with a camera on a moving car and grouped into neighbor pairs by driving commands [JG15]. The forward model in the Intrinsic Curiosity Module (ICM) [Pat+17] predicts the next state s' with the input of a tuple of current state and action (s, a) so that the agent can learn to represent the environment dynamics.

Self-supervised representation learning is naturally applicable to scenarios where multiple modalities are involved. Representations emerge concurrently with different focuses and biases, but often have strong relations with one another. To jointly model multiple modalities, such as audio and visual components of videos [Gao+20], a binary classification model to discriminate whether the visual and auditory input are aligned [AZ17; DTG20], or a regression model to predict corresponding audio statistics given vision [Owe+16] can be established. Although these settings are simple enough, they make use of the unity assumption of events, such that extraordinary abilities can be acquired, *e.g.* sound localization, audio-visual retrieval [AZ18], and speech separation [OE18]. In our case, we train a discriminating model that is easy to implement and applicable for general usage.

The available sensory perception for robots can be even diverse [Cal+18; Mur+18; Gan+20b; Che+20]. A work by [Lee+19] shows that fused representations of visual input, force-torque sensing, and proprioception by self-supervision are beneficial for sample efficiency. Synchronizing multimodal cues and handcrafting modularized tasks to align them properly. We keep the complexity low by focusing on the impact of sound.

4.2.2 Active Exploration

A RL agent can gain remarkable abilities by optimizing the objective of maximizing the accumulated reward of experiences [Sil+21]. However, for a task with sparse rewards [Nai+18; Sek+20], which is a common case, the learning process can be quite slow due to the inefficiency of sampling. Reward-shaping [Hu+20] is a commonly used method to alleviate this problem, but it requires expert knowledge and human

effort to tune and is vulnerable to environmental disturbance. Many active exploration strategies have been investigated to encourage the agent to seek novel states [PGG19; Eys+19; Pat+17; Bur+19b] among which ICM proves to be robust on many tasks [Bur+19a; Las+21b]. So we construct our auditory-curiosity module on top of ICM, building on an existing visual processing pathway.

As an alternative to sound, using haptic sense as feedback and reward [Raj+21] achieves good performance and active exploration in terms of frequent contacts, supporting sample-efficient learning. Similar to our work, [Gan+20a] uses vision and action to predict the next clustered auditory events, and the classification error will thus be used as the overall intrinsic reward. However, the transferability of learned representations is not as well studied as in our work. A discriminator is trained in work [DTG20] to exploit information consistency of aligned image sequences and audio, and an intrinsic reward is computed according to the uncertainty of the classifier. Despite the extra efforts required to construct offline data sets, this work is restricted to Atari games or audio-dense scenarios. When applied to robotic control, an object will only produce sound when there is contact. Silence or background noise dominates most of the time. It is even harder to construct misaligned pairs because a random shuffle strategy fails in cases where silence is capable of being aligned with most of the visual scenes. Moreover, a cold-starting problem will arise, particularly when the policy is not sufficiently rewarded to produce collisions. Therefore, we use intrinsic motivations extracted from both visual and auditory cues.

4.3 METHOD: ISCM

Preliminaries. Typical RL problems are formulated as Markov Decision Processes (MDPs)³, comprised by states $\mathcal{S} = \{s\}_N$, actions $\mathcal{A} = \{a\}_N$, transition probability $\mathcal{P}_{ss'}^a$, and rewards $\mathcal{R} = \{r\}$. The goal of the agent is to optimize the policy $\pi_\theta(a|s)$ that maximizes the expected discounted sum of rewards $\mathbb{E}_{\pi_\theta} \sum_{n=0}^{\infty} \gamma^n r_{t+n}$, where γ is the discount factor. Usually, out of realistic constraints and generality considerations, we do not have full access to internal states $\mathcal{S}$ but a series of sensors attached to the workspace, resulting in partial observations $\mathcal{O} = \{o\}$. Before being fed into the policy module, high-dimensional sensory inputs must be compressed to latent states that can efficiently represent the environment [Mni+15; Bur+19a].

The following subsections describe the proposed ISCM framework, which consists of visual representation learning with self-supervised crossmodal dynamics modeling (§ 4.3.1), intrinsic visual-auditory rewarding (§ 4.3.2) with dynamics modeling errors as the intrinsic motivation, and the joint learning process for both representation and policy (§ 4.3.3).

3: $s_t, a_t, r_t, s_{t+1}, o_t$ are the state, action, reward, next state, and observation at time step t , respectively. Without specification, we use s, a, r, s', o to simplify the notation.

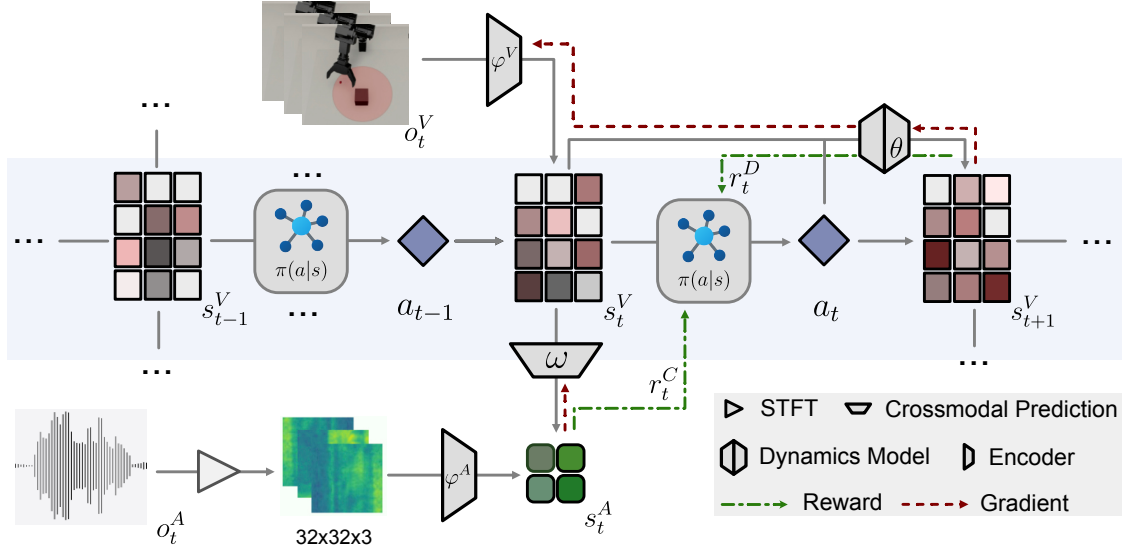


Figure 4.2: The Intrinsic Sound Curiosity Module (ISCM) framework comprises dynamics modeling (as in the vanilla ICM), representation learning, and intrinsic reward computation that leverages both dynamics-based signals and visual-auditory crossmodal cues.

4.3.1 Visual Representation Learning

Visual exploration is a fundamental task for embodied AI agents, where the agent is allowed to actively gather visual information about the environment and then distill knowledge into models such as a topological map or a dynamics model [Dua+22]. Generally, the agent is supposed to explore as many novel states as possible with an internal encouragement aligned to certain targets, *e.g.* a measure of the *coverage* such as the amount of visited unique states in a navigation scenario [DTG20], a *prediction error* of a learned dynamics model [Pat+17; DGI21] or of a reconstruction model when an agent tries to generate other views of an object than the observed ones [Dua+22].

With a combination of multiple sensory inputs for internal states, the agent is allowed to have a more comprehensive view of the environment. However, it will require either a lot of domain-specific assumptions of crossmodal associations or an increase in model complexity [Jae+21; Jae+22] to derive efficient representations from fused inputs. To ensure a fair comparison with vision-only baselines, sound is used solely as a supplementary modality. The agent has access to sound only during the pre-training stage. The baseline encoder, which we compare our model with, is trained by modeling the environment dynamics with visual states, while the one of ISCM (Intrinsic Sound Curiosity Module) additionally fits a visual-auditory sub-task (see Figure 4.2). Before adaptation to downstream tasks, visual encoders of the agents are initialized with weights from the ISCM and ICM baseline. The RL agent policy is trained with Deep Deterministic Policy Gradient (DDPG).

Let the visual and auditory observation at time step t be denoted as o_t^V and o_t^A , respectively. A visual encoding function $\varphi^V(\cdot)$ comprised of convolutional neural networks is thus applied on o_t^V to compute the state $s_t^V = \varphi^V(o_t^V)$, which is later used for both policy learning and dynamic environment modeling. Evidence shows that a well-pretrained encoder is essential for the generalization of supervised learning models [PY10; DZ17] and RL agents [Las+21b; Bur+19a]. Hence, the sound-free visual

encoder and the sound-guided counterpart are trained separately for comparison.

There are two jointly-trained models to model environment dynamics in ICM: a *forward dynamics model* $M_\theta^{\text{Fwd}}(\cdot)$ and an *inverse dynamics model* $M_\theta^{\text{Inv}}(\cdot)$. The forward model tries to predict the forward n -step transition s_{t+n} (usually, $n = 1$) given the current state s_t and action a_t , *i.e.* $\hat{s}_{t+n} = M_\theta^{\text{Fwd}}(s_t, a_t)$, while the inverse one tries to predict the action taken between aligned states $\hat{a}_t = M_\theta^{\text{Inv}}(s_t, s_{t+n})$, which encourages noise-robust representations [Pat+17]. These two dynamics models are optimized concurrently with respect to L_2 constraints, defined as

$$L_t^{\text{Fwd}} = \|\hat{s}_{t+n} - s_{t+n}\|_2^2$$

and

$$L_t^{\text{Inv}} = \|\hat{a}_t - a_t\|_2^2.$$

Note that here we use L_2 loss also for action predictions since we control the continuous actions of the robot arm; otherwise, a cross-entropy loss can be considered for discrete actions.

To benefit from sound, a crossmodal prediction model with parameters ω , which can be either a discriminator (for discrete s^A) or a regression model (for continuous values), is then trained to learn the associations of concurrent vision and sound, *i.e.* $p_\omega(\hat{s}_t^A | s_t^V)$. The crossmodal loss is optimized in latent space, $s^A = \varphi^A(o^A)$, where $\varphi^A(\cdot)$ is a fixed auditory encoder with output suitable for either discrimination or regression. Typically, to construct auditory features for regression, $\varphi^A(\cdot)$ consists of randomly initialized neural networks, with no requirements for any further training. These representations are compact, stable, and generally reliable [Bur+19a; Bur+19b], especially when dealing with impact sound whose information density could be low compared to information in speech. Alternatively, $\varphi(\cdot)$ can be chosen as a threshold to distinguish valid event sound from background noise, considering the simplicity and the aforementioned knowledge that even with a simple discriminating task, surprisingly good abilities can be acquired through cross-modal learning [AZ18; DTG20; Zha+18]. Much of the time in a manipulation scenario, there is just silence before any valid collision or friction happens. To avoid the model eagerly collapsing to zero prediction and causing dying neurons [Lu+20], we use weighted cross entropy loss by w^+ to amplify the importance of positive samples, *i.e.* *crossmodal prediction loss*:

$$L_t^C = -w^+ \cdot p(s_t^A | s_t^V) \log p_\omega(s_t^A | s_t^V) - [1 - p(s_t^A | s_t^V)] \log [1 - p_\omega(s_t^A | s_t^V)] . \quad (4.1)$$

For regression, the optimization is similar except for an unweighted L_2 loss

$$L_t^{C'} = \|\hat{s}_t^A - s_t^A\|_2^2. \quad (4.2)$$

To summarize, the objectives for visual representation learning in vanilla ICM and the proposed ISCM are separately written as

$$\arg \min_{\varphi, \theta} \mathbb{E} [L_t^D] \quad (4.3)$$

and

$$\arg \min_{\varphi, \theta} \mathbb{E} \left[(1 - \alpha) L_t^D + \alpha L_t^C \right], \quad (4.4)$$

respectively, where $L_t^D = \beta L_t^{\text{Fwd}} + (1 - \beta) L_t^{\text{Inv}}$ is the overall *dynamics loss* and α, β are hyper-parameters to mediate the relative importance between modules. Note that the objective is expected to be minimized over samples with time stamp t . Therefore, it is reasonable to encourage the agent to collect informative samples by injecting the model's prediction error, as a form of intrinsic reward, into the agent's exploration objective.

4.3.2 Intrinsic Visual-Auditory Reward

Unlike typical supervised learning in which the data is drawn from a stationary distribution, RL agents actively seek samples according to the policy that updates towards reward-weighted maximum likelihood estimation [PMA10]. So when dealing with the sparse-reward case, the intrinsic reward mechanism helps prevent representations from focusing too much on non-interesting areas.

The visual-auditory reward in our case is defined as $r_t^C = \log(L_t^C + \epsilon)$, *i.e.* if the agent's (unity) assumption violates its perception, it will be encouraged to experience more, and vice versa. ϵ is a constant added to maintain numerical stability, particularly for values near zero. With $r_t^D = \log(L_t^D + \epsilon)$ as the ICM reward when modeling the environment dynamics, the overall intrinsic reward of ISCM is computed as

$$r_t = \lambda r_t^C + (1 - \lambda) r_t^D, \quad (4.5)$$

where λ controls the relative importance of crossmodal prediction and dynamics modeling for exploration.

4.3.3 Representation and Policy Learning

The learning process is separated into 1) fully unsupervised pre-training and 2) task-specific fine-tuning stages with the curiosity mechanism omitted. It begins with an agent freely exploring an environment, trajectories of $\{o_t^V\}$ and $\{o_t^A\}$ are accumulated for representation learning; intrinsic rewards are computed for policy learning. When an exploration budget is reached or when the agent is believed to have enough knowledge, the pre-trained visual encoder will be fixed, and the actor-critic networks will be fine-tuned on downstream tasks with only vision and extrinsic sparse rewards accessible.

4.4 EXPERIMENTS AND RESULTS

We aim to answer the following research questions regarding **Objective I**:

- **R.Q. 4.1** Does intrinsic sound curiosity help the agent to explore more actively and learn effective representations?

- **R.Q. 4.2** Does unsupervised policy pre-training help the agent to adapt to new tasks?
- **R.Q. 4.3** How does the choice of crossmodal prediction affect the performance?

4.4.1 Experimental Setup

The experiments are carried out in simulation because unsupervised exploration in the real world is costly, which we leave for future work. One way to manipulate objects that produce authentic sound is to use a fixed data set with a physics computation interface [Gao+21]. For generality, we build our multimodal manipulation scenarios, shown in Figure 4.1, based on ThreeDWorld [Gan+21], *cfr.* § 2.3.1 “ThreeDWorld” on page 20, a novel embodied AI simulator [Dua+22] which is built upon the Unity game engine with multimodal capacities. To the best of our knowledge, it is the only one so far that supports physically simulated impact and scrape sounds [TCM19; Aga+21] at run time. The tabletop robot is composed of a 6-DoF OpenManipulator-Pro robotic arm and a 2-DoF gripper⁴. It is allowed to manipulate cubes with diverse physical properties that are essential for both dynamics and sound characteristics, *e.g.* masses, materials, and bounciness.

Observations. A camera and a single-channel microphone are placed above the table to capture observations. We focus more on vision and sound, so the robot’s proprioception is not included, and the robot has no knowledge of the object’s coordinates.

Task Setting. One or several cubes are randomly placed inside a red circular area, and the goal is to push them out of the circle within a limited number of steps (*cfr.* Figure 4.1). Specifically, each step will have a penalty of $-1/50$, and an immediate reward of 1 will be delivered once the task is completed; otherwise, the episode ends at 50 steps.

4.4.2 Implementations

We use the ICM implementation of Unsupervised Reinforcement Learning Benchmark (URLB) [Las+21b] as the baseline, and further extend it to our ISCM architecture. Refer to Algorithm 1 for pseudo code⁵.

Visual observations are processed as follows: a) Raw RGB image observations ($o_{t-2}^V, o_{t-1}^V, o_t^V$) are stacked to the size of $84 \times 84 \times 9$ pixels. b) Four layers of Convolutional Neural Network (CNN) with ReLU activation are applied subsequently to encode vision to a latent state s_t . c) A model using two fully connected layers with ReLU activation is constructed for sound prediction. d) Visual inputs are available in both pre-training and fine-tuning.

Auditory observation processing: a) An auditory observation o_t^A is generated at run-time by a physical engine; it is then converted to the spectrogram using Short-Time

⁴: https://github.com/ROBOTIS-GIT/open_manipulator_p

⁵: See code link: <https://github.com/xf-zhao/ISCM>

Algorithm 1: Pseudo code for Intrinsic Sound Curiosity Module (ISCM)

```

1 Initialize: Replay buffer  $\mathcal{D} \leftarrow \emptyset$ , policy neural networks  $\pi$ , visual encoder  $\varphi^V$ ,
  auditory encoder  $\varphi^A$ ;
2 for  $n = 1$  to  $N_{pre-train}$  do                                     /* Exploration */
3   Observe  $o_t = \{o_t^V, o_t^A\}$ ;
4    $s_t \leftarrow \varphi^V(o_t^V)$ ,  $s_t^A \leftarrow \varphi^A[\text{STFT}(o_t^A)]$ ;
5   Compute  $L_t^D$  and  $L_t^C$ ;
6    $a_t \leftarrow \pi(s_t)$ ;
7   Observe  $o_{t+1} \sim \mathcal{P}_{ss'}^a$ ;
8   Compute intrinsic rewards  $r_t$ ;
9    $\mathcal{D} \leftarrow \mathcal{D} \cup (o_t, a_t, o_{t+1})$ ;
10  Sample  $\mathcal{D}_{batch}$  from  $\mathcal{D}$ ;
11  Update  $\varphi^V$ ,  $\pi$  using samples in  $\mathcal{D}_{batch}$  with Equation 4.4 and Equation 4.5;
12 Fix visual encoder  $\varphi^{V*} \leftarrow \varphi^V$  for evaluations;
13 Chose task  $T$ ;
14  $\mathcal{D} \leftarrow \emptyset$ ;
15 for  $n = 1$  to  $N_{fine-tune}$  do                                     /* Adaptation */
16   Observe  $o_t^V$ ;
17    $s_t \leftarrow \varphi^{V*}(o_t^V)$ ;
18    $a_t \leftarrow \pi(s_t)$ ;
19   Observe  $o_{t+1}, r \sim \mathcal{P}_{ss'}^a$ ;
20    $\mathcal{D} \leftarrow \mathcal{D} \cup (o_t, a_t, r, o_{t+1})$ ;
21   Sample  $\mathcal{D}_{batch}$  from  $\mathcal{D}$ ;
22   Update  $\pi$  using samples in  $\mathcal{D}_{batch}$  with extrinsic rewards;
23 Evaluate  $\pi$  with the accumulated rewards on task  $T$  for performance;

```

Fourier Transform (STFT), *i.e.* $o_t^S = \text{STFT}(o_t^A)$. This is a consideration that complex sounds that come from objects with distinct materials are more distinguishable in the frequency domain with the help of the Fourier transform. Since the agent is updated with samples from a replay buffer and actions are chosen solely based on the visual input, there is no wait for the computation of STFT in real-time control. b) Spectrograms ($o_{t-2}^S, o_{t-1}^S, o_t^S$) are then stacked as the auditory input of $32 \times 32 \times 3$ size. c) Finally, s_t^A is obtained by applying a certain threshold for silence discrimination and by passing through a fixed auditory encoder with 36-dimensional output for regression. Auditory inputs are available only in pre-training.

ICM Modeling (baseline). ICM modeling steps are as follows: a) Trajectories of (s_t, a_t, s_{t+n}) are fed into the ICM dynamics models for both encoder training (Equation 4.3 with $\beta = 0.5$) and intrinsic reward r_t^D computation with $\epsilon = 1$. b) The sample with r_t^D is thus used to train a DDPG base learner. c) After enough explorations, the DDPG model will have to adapt to tasks with supervised rewards.

ISCM Modeling (ours). ISCM modeling steps are as follows: a) Paired multimodal observations (o_t^V, o_t^A) are used to train the visual encoder (Equation 4.1 and Equation 4.4 with $\omega, \alpha, \beta = 100, 0.2, 0.5$) and to compute intrinsic crossmodal rewards r_t^C . b) Overall intrinsic reward (Equation 4.5 with $\lambda = 0.8, \epsilon = 1$) is thus computed to

train a DDPG-based learner.

All the mentioned neural networks are optimized by RAdam [Liu+20] with a learning rate equal to 0.001. For many unsupervised RL approaches, the performance decays with an excessive number of environment interactions [Las+21b]. There is so far no general strategy to determine when to early-stop explorations for better generalization. We empirically choose 200K environment steps to pre-train and 30K steps to fine-tune, considering the convergence of learning curves. The result is averaged over 4 runs with different seeds.

4.4.3 Evaluation

The performance of unsupervised agents can be evaluated by means of measuring the adaptation process on downstream tasks or by statistically analyzing data diversity, *e.g.* counting of collisions [Gan+20a], variance in the introduced sensory vector [Raj+21], or transformations (distance of movement, orientation changes) of objects. However, the latter method varies from task to task and is not always applicable.

Whereas the main focus of this work is to demonstrate the effectiveness of learned representations, the tasks are chosen to be simple to master for an agent. In this case, accumulated reward rather than success rate is more appropriate to compare the learning efficiency because the former can reflect the consumed steps, under the setting that the agent is punished for every unfinished step. Following previous works in URL [Las+21b], task-related (extrinsic) rewards are solely evaluated as performance metrics rather than being used for training.

- ▶ During pre-training, the extrinsic reward act as a measurement of agent activeness, *i.e.* how often an agent occasionally achieves meaningful events.
- ▶ In the adaptation stage, the extrinsic reward is used to evaluate the performance of the agent in a task-specific manner.

4.4.4 Results and Discussion

RESEARCH QUESTION 4.1 Does intrinsic sound curiosity help the agent to explore more actively and learn effective representations?

The activeness of exploration can be heuristically measured by the diversity of collected states, object interactions, and the incidental accumulation of extrinsic rewards. Note that these extrinsic rewards are not provided during training but serve as indicators of accidentally achieving meaningful events. We observe that when sound is involved, the agent shows increased interest in object interactions, leading to more frequent accidental completions of such events (see [Figure 4.3](#)).

Observations from URLB [Las+21b] suggest that the learned representations are generally universal and transferable, whereas the behavior policies may not be—particularly those trained with access to perfect state information (*i.e.* fully observable MDPs). We compare episode rewards during fine-tuning for DDPG learners with identical hyper-parameters but different model initializations: (1) full ICM pre-training; (2)

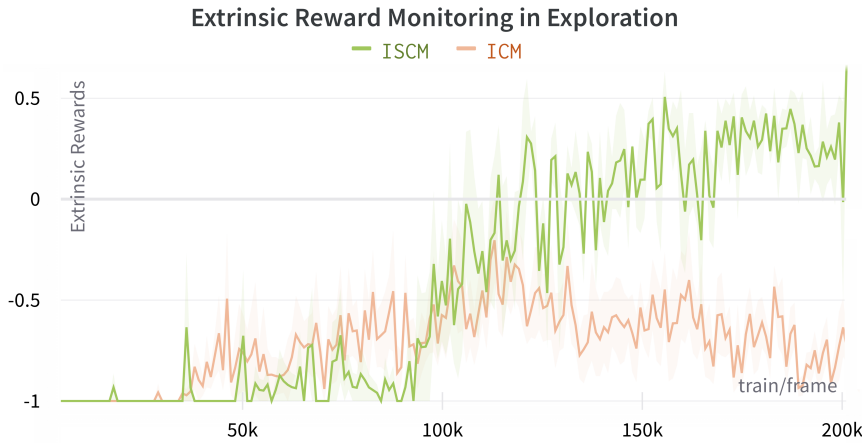


Figure 4.3: Monitoring of extrinsic rewards (recorded but never used) in exploration. The ISCM agent has more chances of accidentally accumulating extrinsic rewards as a result of sound contributing to additional rewards.

ICM representations with a re-initialized policy (ICM-PR); (3) full ISCM pre-training; (4) ISCM representations with a re-initialized policy (ISCM-PR); and (5) no pre-training (just policy learning with DDPG). As is shown in Figure 4.4, we reiterate that representations learned in unsupervised exploration are essential, and add further findings:

- ▶ There is a big performance gap between the DDPG learned from scratch (DDPG, dashed gray curve) and the other four with pre-trained weights (colored curves), which suggests that unsupervised exploration is helpful for faster adaptation to new tasks.
- ▶ The full pre-trained module (representations and behavior policy) with sound (ISCM, solid green curve) outperforms the baseline that solely depends on vision (ICM, solid orange curve).
- ▶ Without considering pre-trained policies, representations learned with a visual-auditory prediction (ISCM-PR, dashed green curve) outperform the ones learned with only vision (ICM-PR, dashed orange curve).

RESEARCH QUESTION 4.2 Does unsupervised policy pre-training help the agent to adapt to new tasks?

By comparing all solid with dashed curves, we find pre-trained policies to have positive effects on task adaptation, which reveals that skills acquired in unsupervised exploration are also reusable. However, more studies on policy analysis, *e.g.* decomposition of the learned policy for abstract behaviors, are required for a clear view.

RESEARCH QUESTION 4.3 How does the choice of crossmodal prediction affect the performance?

A vision-to-sound regression model using L_2 loss (cf. Equation 4.2) is trained with the same hyperparameters, replacing the crossmodal prediction module in Figure 4.2 with a regression head. See Figure 4.5 for comparison results. Though a vector (for

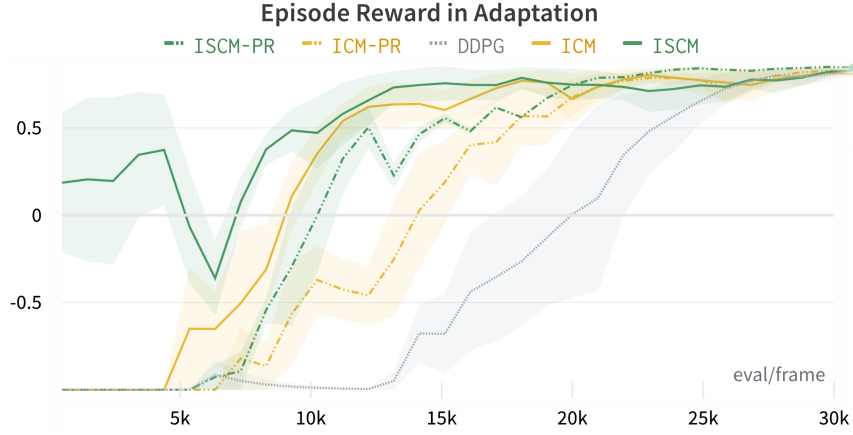


Figure 4.4: Episode rewards in fine-tuning stage accumulated by DDPG learners with all hyper-parameters configured the same except for the initialization of models: 1) ICM: models with representations and policy pre-trained by ICM. 2) ICM-PR: models with ICM pre-trained representations but a re-initialized policy. 3) ISCM: models with representations and policy pre-trained by ISCM. 4) ISCM-PR: models with ISCM pre-trained representations but a re-initialized policy. 5) DDPG: models without pre-training.

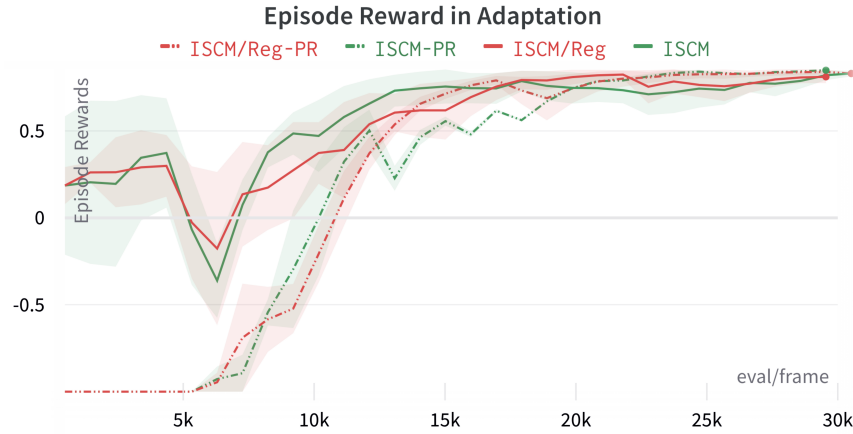


Figure 4.5: Episode rewards in fine-tuning stage are accumulated by base DDPG learners that are initialized differently. 1) ISCM: fully pre-trained module with a discrimination auditory encoder. 2) ISCM/PR: pre-trained representations (but policy re-initialized) with a discrimination auditory encoder. 3) ISCM/Reg: fully pre-trained module with a regression auditory encoder. 4) ISCM/Reg-PR: pre-trained representations (but policy re-initialized) with a regression auditory encoder.

regression) rather than a scalar (for discrimination) is believed to have a higher capacity of information, we find the discriminator setup (green curves) achieves a comparative performance with a regressor (red curves), while being simple to implement. Similar findings can also be found in recent works [Gan+20a] where clustered auditory events are being predicted instead of regressing sound features. It may result from the following reasons: 1) impact sound presents not much more information than a deduction of event occurrence; 2) simulated sound is still far away from perfect, such that vision, sound, and dynamics are not matched well as in reality. Future work will include construction of more complex environments and sim-to-real adaptations to investigate more on these research questions.

4.5 CONCLUSION

Sound is one of the most common and efficient modalities, but is yet less considered for learning either simulated or real-world robotic manipulations. Unlike many of the curiosity-driven RL variants, especially the ones combined with audio that pay attention to non-robotics applications such as playing Atari games, we are focusing on investigating how robots can benefit from exploring multimodal environments. In this chapter, the importance of unsupervised representation learning and of active exploration is addressed. We further propose the ISCM architecture to use physics-based sound as guidance regarding both aspects. Our experiments demonstrate that a sound-guided reinforcement learner is more active and excels in forming sufficient as well as stable representations over vision-only baselines.

4.6 LIMITATIONS AND FUTURE WORK

Although this work employs advanced physics-based sound simulation to evaluate the proposed ISCM architecture, its applicability and generalizability to real-world scenarios remain to be explored, as real-world sounds are often complex, noisy, and uncertain. Due to the simplicity of the current environment, the agent’s capacity for self-determined exploration and learning is limited. Future work in more complex settings may foster greater self-determination, allowing agents to move beyond event-based cues to interpret semantically meaningful signals and discover emergent behaviors. This research direction will be further discussed in [Chapter 7 “Agentic Skill Discovery”](#) on page 99.

INTERACTIVE MULTIMODAL PERCEPTION USING LARGE LANGUAGE MODELS

5

World Models + Semantics + Self-determination

To address **Objective II**, i.e. “to develop an interactive multimodal perception framework in which the agent actively gathers, integrates, and semantically interprets diverse sensory inputs, enabling grounded semantic understanding and context-aware decision-making in complex environments.”, this chapter introduces an *interactive exploration strategy built on top of LLMs* to leverage multimodal cues, including vision, audio, tactile, and weight.

As discussed in **Chapter 4**, integrating multimodal cues has been highly effective for both representation learning and decision-making, enabling agents to interact more effectively with their environments. However, Reinforcement Learning (RL) methods applied previously are often data-hungry and require a large amount of data to learn relevant knowledge of the environment. In contrast, planning methods based on LLMs leverage off-the-shelf knowledge, e.g. *world modeling in semantic space* (i.e. natural language representations), and reasoning abilities acquired through large-scale training. These models can integrate multiple modalities and reason about interactions in a *self-determined manner*, enabling both intrinsic motivation and self-regulation through closed-loop reasoning based on environmental feedback. This offers a compelling alternative to conventional approaches that depend on costly data collection or manually designed structures.

Programming robot behavior in a complex world faces challenges on multiple levels, from dextrous low-level skills to high-level planning and reasoning. Recent pre-trained Large Language Models (LLMs) have shown remarkable reasoning ability in few-shot robotic planning. Advanced knowledge and reasoning abilities inside large foundation models enable embodied agents to a dramatic degree of generalization, promising the extendibility to planning in unforeseen environments and tasks. However, it remains challenging to ground LLMs in multimodal sensory input and continuous action output, while enabling a robot to interact with its environment and acquire novel information as its policies unfold. For example, in a robotic manipulation task, an agent might first use vision to identify an object and then rely on tactile feedback to assess its texture or stability. By integrating these sensory signals, e.g. learning a joint representation of them, the agent can make more accurate predictions and decisions, thereby enhancing its ability to interact with the environment. Understanding and modeling the physics of the world is crucial for task completion. Large foundation models, e.g. LLMs and Vision Language Models (VLMs), trained on vast amounts of knowledge and equipped with reasoning abilities, exhibit in-context modeling of the environment at a semantic level.

To address these challenges, **Multimodal environment chatting** (Matcha) **agent**, an interactive perception framework, is therefore proposed with an LLM as its backbone, whose ability is exploited to instruct epistemic actions and to reason over the resulting

multimodal sensations (vision, sound, haptics, proprioception), as well as to plan an entire task execution based on the interactively acquired information. [Figure 5.1 on the next page](#) shows the conversational interaction process for perception and decision-making. The framework is verified in a robot interaction scenario built with multimodal cues being accessible, whereas observations from each modality can only provide partial information to solve a given task, necessitating a robot to decide on a range of epistemic actions in order to sample sensory information among multiple modalities, before being able to execute the task correctly. Our study demonstrates that LLMs can provide high-level planning and reasoning skills and control interactive robot behavior in a multimodal environment, while multimodal modules with the context of the environment state help ground the LLMs and extend their processing ability. The project website can be found at <https://matcha-agent.github.io>.

5.1 INTRODUCTION

How do humans perceive the surroundings to uncover latent properties?

Suppose you are presented with an uncommon object in a strange shape and of unknown material, you may explore its properties in both passive and active ways, if possible, *e.g.* by observing the geometry, touching, and even knocking on the surface in order to deduce its exact functionalities from the feedback. Unnecessary explorations, which could be essential for other scenarios such as smelling, will not be performed in this context unless something counterintuitive happens. We humans naturally perform these *multimodal observations and examinations* in daily life through *common sense and established knowledge*, and over time we adapt with the accumulation of experience [Bar+06].

In this work, we show that this is also possible for a robot well-equipped with multiple sensors and LLMs. An environment may be filled with rich content, and the robot can be overwhelmed with diversified sensory stimuli. An intelligent robot should (i) selectively attend to relevant stimuli, avoiding unnecessary distraction by irrelevant details; and (ii) generate appropriate responses to the perceived information within the context of the task at hand.

Interactive Multimodal Perceptions

Like humans, robots can perceive the environment in either a passive or an interactive way [KNK21]. *Passive perception* refers to methods such as visual or auditory monitoring, enabling robots to efficiently gather information without actively interacting with the environment. However, passive perception has inherent limitations, the most notable being its inability to effectively address *epistemic uncertainty*, the type of uncertainty arising from a lack of knowledge [CK23].

Epistemic uncertainty inevitably arises from diverse sources, *e.g.* from the ambiguity in human instructions, from low-resolution sensing (*e.g.* reduced image size for convolution), or from insufficient modalities. Many of them can only be reduced with

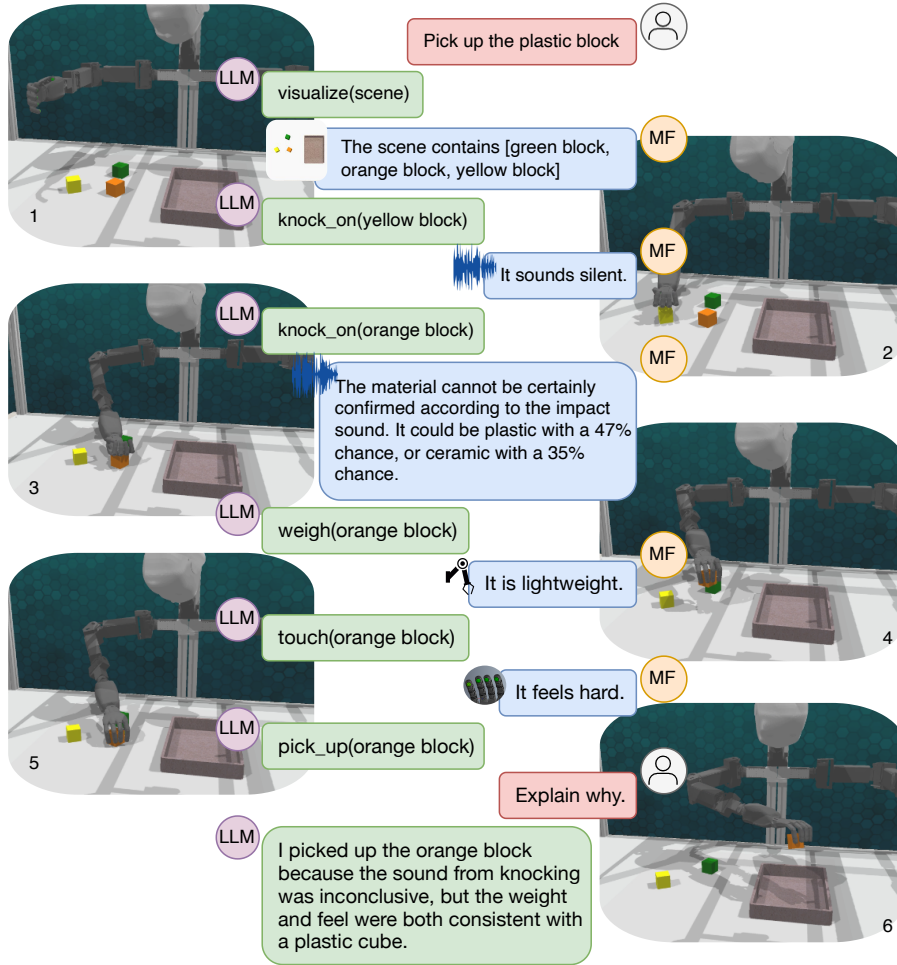


Figure 5.1: Upon receiving a human instruction, the robot recurrently “chats” with the environment to obtain sufficient information for task completion. An LLM generates action commands to interactively perceive the environment and, in response, the environment provides multimodal feedback (MF) through dedicated multimodal perception modules.

interactive perception, in which a robot actively interrogates the environment to increase accuracy and even uncover latent information. For example, when being asked to deliver a *steel* screw instead of one with a similar color & shape but made of *aluminum*, an assistant robot may need to locate possible candidates with *passive* vision and further, *interactively*, resort to a weighing or a magnetic module for confirmation.

Despite the promising advantages, interactive perception is less common than the passive manner because it entails increased complexity [Li+23c]. Efforts are needed to design a mediating system to handle various sensory data and to adapt to changes in the conditions of both the robot and the environment, such as a new robotic module being available or the involvement of novel objects.

Chatting with the Environment

LLMs have been showing incredible potential in areas besides robotics [Ahn+22; Cui+23; Lyn+22; Mia+23]. Human knowledge that resides in LLMs can help a robot abstract and select only suitable features, *e.g.* relevant to the region of interest

or informative modalities, to simplify the learning process. Moreover, in terms of generalizability, the knowledge of LLMs allows a behavioral agent to adapt efficiently to novel concepts and environment structures. For instance, when being asked to *use one adjective for each to describe how a sponge and a brick feel*, ChatGPT¹ will respond with “soft” and “hard” respectively. This is helpful for a robot with a haptic sensing module to distinguish between these two novel, never-seen objects.

LLMs are usually generative models that predict tokens to come, but with certain designs, *e.g.* conversational prompts, LLMs are capable of generating chat-like texts. This allows their integration with a robot to not only plan with respect to a robot’s built-in ability [Zen+23a; Ahn+22] but also respond according to environment feedback. However, they cannot directly process application-specified raw multimodal data. We resort to modular perceptions for each modality that are separately trained before being plugged into the LLM backbone. Each module semantically translates the resulting multimodal sensations into natural language that can be understood by LLMs and processed in a unified manner in a semantic space.

Our contributions are threefold. Firstly, we establish a manipulation scenario with multimodal sensory data and language descriptions. Secondly, we propose Matcha² agent, where an LLM is prompted to work in a chatting fashion, thus having continuous access to environment feedback for contextual reasoning and planning. Finally, we show that LLMs can be utilized to perform interactive multimodal perception and behavior explanation. Accordingly, an interactive robot can make reasonable and robust decisions by resorting to LLMs to examine objects and clarify their properties that are essential to completing the task (see Figure 5.1 on the preceding page).

5.2 RELATED WORK

5.2.1 Multimodal Learning and Robotic Information Gathering

Research in multimodality in robotics nowadays attracts growing attention [Akk+23] because of its success in, for example, audio-visual learning [Zha+22; Wei+22b; Zhu+21] and language-visual learning [SMF22a; SMF22b]. It is beneficial and sometimes essential for a robot to learn from multimodality because one modality could carry some distinct information, *e.g.* tones in speech, that cannot be deduced from another [Lee+22].

Capable robots require managing one or several sensors to maximize the information needed for disambiguation [Bar+06] regarding a specific goal. This problem is known as *active information acquisition* [Ata15; WKS21] or, particularly in robotics, *robotic information gathering* [RMH21], where robots have to properly select perceiving actions to reduce ambiguity or uncertainty. Besides handcrafted rules, some information

1: <https://openai.com/blog/chatgpt/>

2: By the name of a type of East Asian green tea. To fully appreciate matcha, one must engage multiple senses to perceive its appearance, aroma, taste, texture, and other sensory nuances.

advantage measures, *e.g.* entropy or information gain, are usually employed to maximize [Ata15]. However, the combination of multimodal data is usually challenging. There are studies on fusing multimodal data according to their uncertainties, but this may face numerical instability and is difficult to transfer from one application to another [Wan+22]. Instead of directly fusing the multisensory data in a numerical space, we propose to use multimodal modules to translate them into natural language expressions that an LLM can easily digest.

5.2.2 Large Language Models in Robotic Planning

Recent works use LLMs to decompose high-level instructions into actionable low-level commands for zero-shot planning. They use LLMs as a planner to autoregressively select actions that are appropriate with respect to the instruction according to application-based prompts [Zen+23a], the semantic similarity between mapped pairs [Hua+22], or the contextual language score grounded on realistic robot affordances [Ahn+22]. Other approaches ground LLM knowledge in human interaction [Cui+23] or various other fields where domain knowledge is distinct and modular frameworks can be composed via language as the intermediate representation [Pat+19; Mia+23; Zen+23a]. However, these works design a robot to form a planning strategy with *built-in knowledge*, rather than *interact* with the surroundings and make decisions based on *actively collected information* from the environment. There is no feedback loop for their LLMs to perceive the environmental cues, such that only “blind” decisions are made in the robotic unrolling process. In contrast, our interactive architecture allows LLMs to access the environment state from multiple modalities for adaptive planning.

5.3 METHOD: MATCHA AGENT

5.3.1 Architecture

We propose Multimodal environment chatting (Matcha) **agent** which is able to interactively perceive (*i.e.* “chat” with) the environment through multimodal perception when the information from passive visual perception is insufficient for completing an instructed task. The epistemic actions are executed autoregressively until the agent is confident enough about the information sufficiency in that situation. [Figure 5.2](#) provides an overview of the architecture of Matcha agent. It is a modular framework of three parts: an LLM backbone, multimodal perception modules, and a low-level command execution policy. They connect to each other with language as the intermediate representation for information exchange.

To be specific, given a high-level instruction, especially one that Matcha cannot directly perform with the command policy alone, the LLM backbone will reason the situations and select the most contextually admissible perceiving command to gather information. After the execution of the policy module, the resulting environment response is processed by a correspondingly evoked multimodal perception module

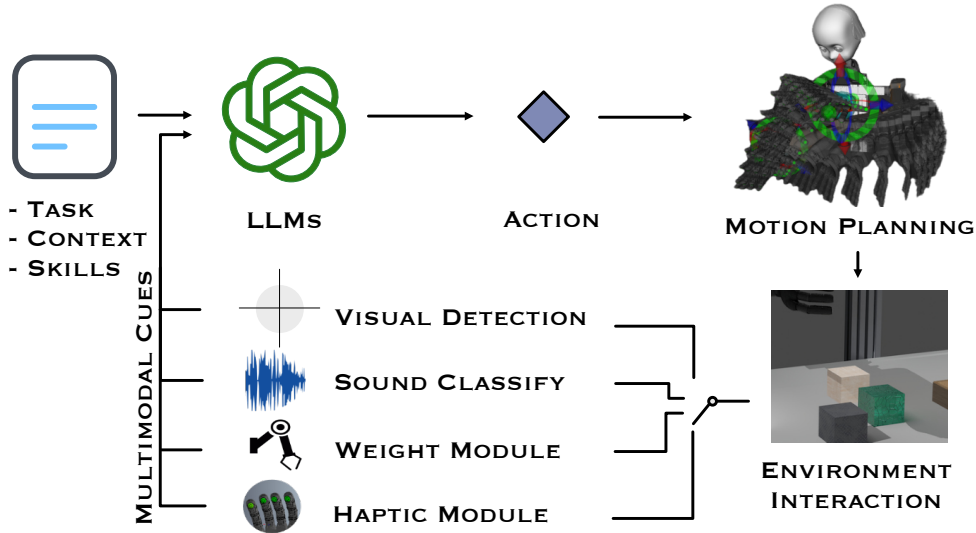


Figure 5.2: Overview of Matcha. The framework contains LLMs as backend, multimodal perception modules, and a language-conditioned control policy implemented with motion planning. These components communicate with each other with natural language as the intermediate representation. Three types of language information are involved in composing the prompt: 1) language instruction, environment context, and robot capabilities, and 2) LLMs’ decisions and resultant feedback from multimodal perceptions in textual representation. The switch indicates possibly evoking paths of the interactive perception decided by LLMs.

into semantic descriptions, *e.g.* “clinking sound” by an auditory module after the “knock on” action. Finally, the executed command itself as well as the environment state description are fed back to the LLM for future planning. The LLM is employed in a few-shot prompting manner without any need for fine-tuning, being independent of other components. Policy and perception modules can be separately designed and plugged into the framework whenever needed. Intrinsically linked by natural language, this framework is flexible and can scale and adapt easily to possible robotic upgrades or diverse robotic scenarios.

5.3.2 Multimodal Perception and Execution Policy

To demonstrate our framework, we implement a language-conditioned policy using a set of widely accessible and practical modalities. Other varieties for specific scenarios can also be easily integrated due to the flexibility of modularity of the framework. Detailed experimental implementations will be introduced in § 5.4.

Vision. Usually, a monitoring camera is the cheapest option for a robot to passively perceive such rich information. We employ pre-trained ViLD [Gu+22], an open-vocabulary visual detection model, as the vision perception module to detect objects with their categories and positions in the scene. Then, the results will be delivered to a policy module for identification and execution. Meanwhile, a prompt template “The scene contains [OBJ1, OBJ2, ...]” is applied to construct a scene description, which enables the LLM to have an initial impression of the environment. Typically, pre-trained vision models are not designed to discern attributes that extend beyond those easily extractable from topology or textures, such as material composition. The use of low-resolution images for expedited processing exacerbates the loss of information

concerning such attributes. In our experiments, we prioritize demonstrating the integration of diverse modalities instead of extensively fine-tuning ViLD to encompass all aspects.

Impact Sound. Impact sound can provide valuable information for robotic multimodal learning [Zha+22]. However, passive sound collection, *e.g.* through an end-effector-mounted microphone, predominantly captures background noise unless the robot actively generates informative sounds through intentional actions, such as the "knock on" behavior in our implementation. This auditory perception module classifies the consequent impact sound into a description and then wraps it in a natural language form. Actually, a clip of audio may contain sufficient information for some of the usage, *e.g.* to distinguish metal from glass [Dim22]. However, it may not be the case for other scenarios, for example, to select the only targeted one among a set of similar "dull" sounds that could indicate either plastic, wood, or hard paper. Therefore, we showcase both of the designs, *i.e.* one with a specific material classification (*e.g.* "glass") and another with solely low-level and non-distinct descriptions (*e.g.* "tinkling"). The modular output is also wrapped with templates to a full sentence, such as "It sounds tinkling", to guarantee processing consistency with LLMs.

Weight. Weight measurements can usually be obtained via the torque exerted on the robotic arm subsequent to the execution of a "weighing" action. It can be simplified by its weight in simulation. In simulation, this measurement is simplified by directly using the object's mass value. The weight information is directly translated into natural language like "It is lightweight" or "It weighs 30g". Note that with implicit clarification of the scenario and the type of objects that a robot is manipulating, LLMs can interpret numerical values into contextual meanings.

Haptics. Haptic perception is essential for human interaction with the physical world and offers valuable potential for robots to infer properties such as hardness, texture, and compliance. However, high-resolution tactile sensors are often expensive and impractical for many applications. Therefore, in this work, only highly abstract descriptions of force-torque feedback are used following a touch action on an object, *e.g.* "It feels soft" or "It feels hard and smooth".

Execution Policy. The execution policy is conditioned on the generated command by an LLM and the visual information provided by the vision perception module. Once an actionable command together with an identified target is suggested by the LLM, the policy module locates the targeted object and executes a certain action. Meanwhile, the environment feedback will be concurrently collected for multimodal perception modules for further post-processing as demonstrated above.

5.3.3 Prompt Engineering

An issue of grounding LLMs in robotic scenarios is that some of the suggestions generated by LLMs are not executable for a specific robot [Ahn+22; Hua+22], which stems from the fact that LLMs are pre-trained with extremely large open-domain corpora, while the robot is constrained by its physical capability and application scenarios, *e.g.* a tabletop robot is not able to perform a "walk" action.

In this work, the LLM is applied for few-shot planning [Mia+23; Zen+23a], in which all the executable commands are defined together with several task examples as the initial “chat” history. See Table 5.1 for the leading prompt which enables the LLM to become grounded in the specific scenario and follow the contextual patterns for commanding the execution policy. We found that only language models that are large enough can follow the patterns in the prompt strictly, *i.e.* only generate commands that have been defined in strictly case-sensitive letters and with the same amount of allowed parameters for each, while small ones can hardly obey this constraint and generate unexpected commands, which brings extra demands for tuning. As the action planning is performed by LLMs constrained by a given prompt, the proposed Matcha agent demonstrates high flexibility and generalizability upon the possible incorporation of novel actions or perception modules into the system.

Table 5.1: The snippet of the 5-shot prompt setting. (The other four exemplars are omitted here due to the high similarity).

The following are conversations with an AI to complete tasks that require active information gathering from multimodalities. Otherwise, the materials of objects are unknown, and it will be ambiguous for an AI to choose the right object. AI has the following skills to help complete a task:

1. “robot.knock_on()”: to knock on any object and hear the sound to determine the material it consists of. Most of the materials can be determined by this skill.
2. “robot.touch()”: to touch with haptics sensors. It is useful for some of the materials.
3. “robot.weigh()”: to weigh objects if the knocking method is not proper.
4. “robot.pick_up()”: to pick up the targeted object. After this skill is performed, the episode will terminate with the result.

Note that the tasks are always set to be accomplishable, and the selected skill should start with a “>” symbol.

...

Human: “pick up the glass block” in the scene contains [yellow block, blue block, green block]

AI: *robot.weigh(yellow block)*

Feedback: It weighs light.

AI: *robot.weigh(blue block)*

Feedback: It weighs a little bit heavy.

AI: *robot.knock_on(blue block)*

Feedback: It sounds tinkling.

AI: *robot.pick_up(blue block)*

done()

...

5.4 EXPERIMENTS AND RESULTS

We conduct experiments in simulated multimodal manipulation scenarios to evaluate the proposed Matcha framework to study the following research questions regarding **Objective II**:

- ▶ **R.Q. 5.1** Can Matcha integrate multimodal perceptions at the decision level?
- ▶ **R.Q. 5.2** How does the level of abstraction in submodule outputs influence the performance?
- ▶ **R.Q. 5.3** How do different scale LLMs affect the performance?

5.4.1 Experimental Setup

Task and Multimodal Scenario. We evaluate Matcha in an object-picking task: a robot is instructed to pick up an object that is referred to by a latent property, *i.e.* *material*, which is, however, not visually distinguishable under our settings. Tasks are intentionally designed such that information from a single modality could be insufficient to determine object properties, while other perception sources can provide compensations to reduce or eliminate this ambiguity. For example, glass and metal surfaces could exhibit similar hard and smooth properties upon contact, in which case differences in impact sound can aid in further differentiation. Table 5.2 lists variational multimodal descriptions of the materials. These properties are wrapped as natural language sentences before being fed back to the LLM.

Robot Setting. Experiments are conducted in CoppeliaSim³ simulations with the NICOL robot⁴ [Ker+23], where several blocks in various colors, materials, weights, and surface textures are randomly selected and placed on the table next to a brown container (see Figure 5.1). The ViLD [Gu+22] model is meant to be easily generalized to describe complex scenes despite the simplicity of the object setting here. After detection, the objects are represented universally by their name, which serves as a parameter for the action function to identify. Objects with the same color will be distinguished as “.. on the left/right” given the simplicity of avoiding more than two duplicated colors for the same shape. The desktop robot is equipped with two *Open-Manipulator-Pro* arms⁵, but only its right arm is activated to operate. It is capable of executing actions in [“knock on” “touch” “weigh” “pick up”] with a parameter to indicate the targeted object. The first three actions correspond to the interactive perception of impact sound, haptics, and weight, respectively, and the last action finalizes the task by picking and transporting an object into the box. Each instruction is guaranteed to be achievable with the capability of the robot.

Multimodal Simulation. Due to the lack of support for physics-driven sound and deformable object simulation in Coppeliasim, we have implemented reasonable alternatives. For the haptics of objects, we simplify haptic perception by assigning variational descriptions regarding their material, *e.g.* fibrous objects are usually perceived as “soft” and a plastic object can be either “soft” or “hard”. Note that advanced implementations can also be achieved using a neural network, as is used in the sound perception module when haptic data for deformable objects is available. For the impact sound, we split the YCB-impact-sound dataset [Dim22] into training and testing sets and augment them with tricks such as shifting, random cropping, and adding noise. The training set is used to train our auditory classification neural networks, while the audios in the testing part are randomly loaded as an alternative

3: For further details, refer to § 2.3.2 “CoppeliaSim” on page 21 and visit <https://www.coppeliarobotics.com/>. In Chapter 4, impact sound simulation is carried out using ThreeDWorld (*cfr.* § 2.3.1) in an online setting where the subtle physics-driven differences matter as the focus is on joint representation and policy learning. In contrast, this chapter emphasizes decision-level fusion and multimodal reasoning, employing offline sound simulation with audio recordings from a real-world robot dataset is sufficient and reliable.

4: See also § 2.2.1 “NICOL” on page 18 for details.

5: https://emanual.robotis.com/docs/en/platform/openmanipulator_p/overview/

to run-time impact sound simulation for the materials mentioned,

Sound can be informative, though not perfect, for determining materials [Dim22]. Besides showing the mediating ability of multiple modalities by the LLM, we further investigate its reasoning ability by employing indistinct descriptions instead of exact material labels.

- ▶ *Distinct description*: the sound module describes sound feedback by the corresponding material name and its certainty (in percentage) from the classification model, e.g. “It is probably glass” or “It could be plastic with a 47% chance, or ceramic with a 35% chance” The distinct description setting is more task-oriented, and it examines the robot’s ability to mediate multiple sensory data for disambiguation.
- ▶ *Indistinct description*: we listed some commonly used indistinct sound descriptions in human communications in Table 5.2, e.g. using “dull” to describe the sound from a plastic block and “tinkling” to describe the sound for both ceramic and glass objects. This setting is more task-agnostic and thus has the potential for generalization. Moreover, it compels the LLM to infer “professional” material terminology from ambiguous yet multimodal descriptions.

Language Models. The online OpenAI text-davinci-003 API⁶ is applied as the LLM backend because it demonstrates robust instruction-following ability and outstanding reasoning performance⁷. We also evaluate with a weaker but far less expensive LLM text-ada-001, a GPT-3 model which is usually fast and capable of simple tasks, under the same setting as comparison.

5.4.2 Results

We test the proposed Matcha agent in 50 randomly generated scenarios for each setting and report the success rate.

RESEARCH QUESTION 5.1 Can Matcha integrate multimodal perceptions at the decision level?

We found that a pre-trained impact sound classification model achieved an average accuracy of 93.33%. When considering the scenario where the robot randomly knocks on one of three objects and uses the sound module to identify the target material, the theoretical success rate is calculated as 89.18% (computed from $\frac{1}{3}p + \frac{2}{3}p^2|_{p=93.33\%}$, where p is the accuracy of the sound module). Other modalities are generally less distinctive than sound, making it impractical for humans to manually craft fusion rules that significantly improve this baseline. Therefore, this theoretical success rate with only the sound module serves as both a practical upper bound and a baseline

⁶: <https://platform.openai.com/docs/models/gpt-3>

⁷: The code-davinci-002 is not chosen because it is the common sense instead of the ability of code completion that matters more to the active perception. At the time this experiment was carried out, the text-davinci-003 model was the state-of-the-art GPT-3.5 model available; while the later released ChatGPT or GPT-4 model showcases the impressive improved abilities of reasoning, future works will explore the potential of these models.

Table 5.2: Descriptive properties of different materials used by Matcha.

Materials	Impact Sound	Haptics	Weight
Metal	• resonant and echoing • metallic • ringing	• hard and cold • rigid, cold, and smooth	• heavy • 300g
Glass	• tinkling • tinkling and brittle	• hard • hard and smooth • cold and smooth	• a little bit heavy • 150g
Ceramic	• clinking and rattling • rattling • tinkling and brittle	• hard • tough	• average weight • not too light / heavy • 100g
Plastic	• dull • muffled	• hard • soft	• light • 30g
Fibre	• muted • silent	• soft • flexible	• lightweight • underweight • 10g

for evaluating Matcha’s performance if it were to rely solely on impact sound. From [Table 5.3](#), Matcha achieves a relatively higher success rate of 90.57% compared to the ideal theory baseline, indicating its effective utilization of compensatory information from other modalities in addition to sound.

RESEARCH QUESTION 5.2 How does the level of abstraction in submodule outputs influence the performance?

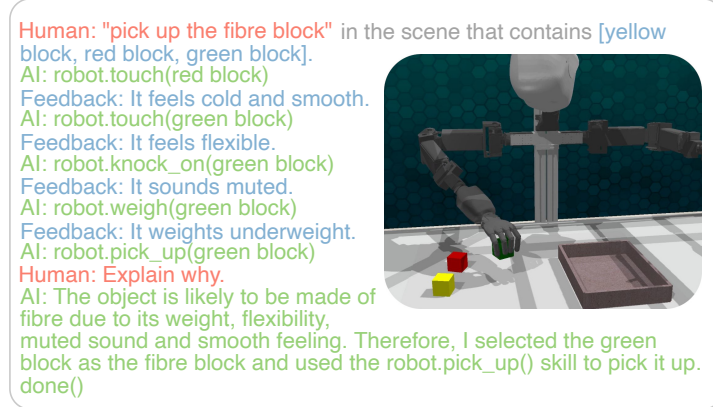
Even when provided only with indistinct descriptions of impact sounds, Matcha achieves a success rate of 56.67%, significantly above the chance level of 33.33% (random selection among three options). This demonstrates the system’s capacity for few-shot deduction using vague or imprecise language. Failure case analysis reveals that materials like glass and ceramic are often confused due to their similar descriptions across modalities such as sound, haptics, and weight. These challenges are mitigated when using distinct descriptions, where the sound classification network can directly predict the material, preserving more information in the translation from perception to language. While Matcha shows strong reasoning capabilities, these results underscore the importance of carefully designed, sufficiently specific modular outputs. Excessive abstraction can lead to ambiguity, reducing the effectiveness of multimodal integration in interactive perception tasks.

RESEARCH QUESTION 5.3 How do different scale LLMs affect the performance?

We observe that only strong, large LLMs (GPT-3 text-davinci-003 in our experi-

Table 5.3: Matcha task success rate with various LLMs.

LLM	Type of Description	Success Rate
text-ada-001	Indistinct	19.05%
	Distinct	28.57%
text-davinci-003	Indistinct	56.67%
	Distinct	90.57%

**Figure 5.3:** A successful example in which the robot deduces “fiber” material with indistinct descriptions of impact sound.

ments)⁸ can perform the task following the primary prompt as we defined, while weak ones possibly generate commands that are not executable and show degraded performance (see Table 5.3). For example, GPT-3 text-ada-001, a smaller model in the GPT-3 model series, may generate commands like “robot.knock_on(*metal block*)” or “robot.weigh(*yellow block, blue block*)”, which are not compatible with the design interface of robot skill functions due to mismatches in parameter types.

5.4.3 Case Studies

We provide case studies to showcase the interactive perception ability of Matcha in multimodal environments. Following the convention of this chapter, texts containing information from humans, LLMs, and the environment are indicated in red, green, and blue, respectively. The case in Figure 5.3 demonstrates that the agent can infer latent material properties from multimodal yet ambiguous descriptions. During the planning process, the agent decidedly terminates exploration of the “red block” and instead engages in interacting with the “green block” exhaustively for a comprehensive examination. Common sense and, moreover, established knowledge in the LLM enables Matcha to efficiently interact with the environment. Matcha’s proficient behaviors provide evidence of effective reasoning, as it aligns with the subsequent explanation provided by the LLM, namely, that fiber can often be considered “flexible” rather than “cold and smooth”. The example depicted in Figure 5.4 presents a fascinating observation: the impact sound of the “orange block” suggests it is more likely to be plastic than metal, but Matcha accurately distinguishes it from plastics

⁸: At the time of our experiments, text-davinci-003 was the most advanced GPT model publicly available.

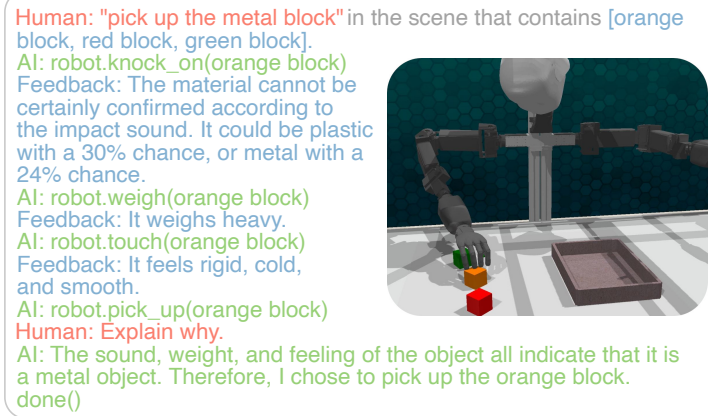


Figure 5.4: A successful example with a distinct description of impact sound. This example shows that by leveraging multimodal perception, LLM rectifies the misclassification that may occur when relying solely on sound modules.

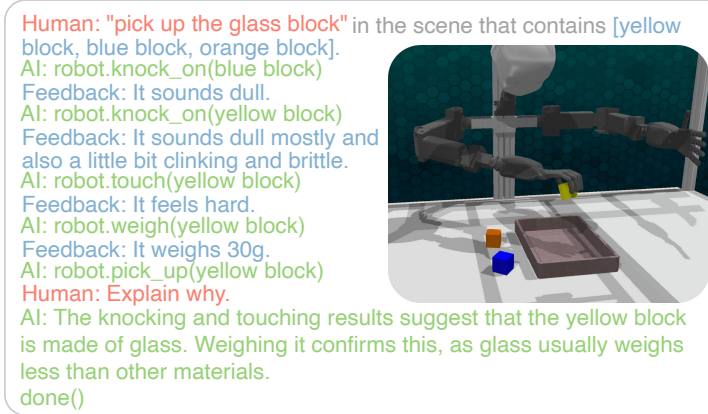


Figure 5.5: An example in which the agent fails to distinguish glass and ceramic in the setup of using indistinct descriptions of impact sound.

after engaging in the interactive perception of weight and haptics. This showcases the potential of multimodal perception to improve classification accuracy. Figure 5.5 provides a failure case with indistinct descriptions where the robot picks up a ceramic block when asked to pick up the one made of glass. The underlying cause of this failure is the sensing similarity between glass and ceramic, which creates difficulty in resolving epistemic uncertainty.

5.4.4 Discussion

Weak LLMs, *e.g.* ones without fine-tuning on instruction alignment [Ouy+22], may not have sufficient capability for precise planning, and thus may require carefully engineered prompts or other grounding techniques. On the other hand, strong LLMs exhibit impressive in-context learning [Zha+23b] abilities. These observations highlight the potential of leveraging knowledge within strong LLMs, as it enables the successful execution of tasks that were previously deemed infeasible. LLMs can derive significant advantages from utilizing common knowledge, being robust to various instructions regardless of their changes in synonym, linguistic structure or

even semantic meanings out of the scope that the robot is initially designed within, e.g. an instruction variation from “the metal block” to “a block that may be suitable for cracking a nut”, in which the robot has to establish a meaningful connection between the object’s multimodal perceptions and the required utility. Nevertheless, the reasoning trace may not always align with human expectations. There are cases in which LLMs may prematurely draw conclusions due to their limited logical reasoning ability, particularly when faced with a task that requires reasoning from a long list of facts.

5.5 CONCLUSION

The Matcha framework achieves strong generalizability by leveraging the commonsense knowledge in LLMs, whereas control algorithms like those trained with Reinforcement Learning (RL) [Li+23c; Sin+20] require extensive data to learn cross-modal commonsense [Sin+20] and remain less efficient and generalizable. Their potential for integration and enhancement with other fields has attracted growing attention from different research areas. In this work, we demonstrate the feasibility of using an LLM to realize interactive multimodal perception. We propose Matcha, a multimodal interactive agent augmented with LLMs, and evaluate it on the task of uncovering object-latent properties. Experimental results suggest that our agent can perform interactive multimodal perception reasonably by taking advantage of the commonsense knowledge residing in the LLM, being generalizable due to its modularity and flexibility.

5.6 LIMITATIONS AND FUTURE WORK

Abstraction. Representing the real world with natural language can be limited by environment dynamics and the feasibility of abstracting desired behavior with natural language. The vision module applied in this work is a separable visual detection module, which is unable to describe the scene with fine-grained details. This brings the requirement of vision-enabled LLMs [Zhu+23; Ton+22; Zit+23], built on which the reasoning can be malleable.

Fine-tuning and Reasoning. While large LLMs excel at complex tasks, their high computational and memory demands make local deployment costly. Future work will focus on distilling domain-specific knowledge into smaller, efficient models for greater flexibility and control. Reasoning, crucial for long-term planning and complex decision-making, will be further explored in [Chapter 6](#).

ENHANCING REASONING VIA LOGIC-GUIDED INFERENCE SCALING

6

Semantics + Self-determination

To address **Objective III**, *i.e.* “to enhance agent reasoning abilities to interpret complex instructions and make informed decisions”, this chapter introduces the application of logic principles to guide LLM reasoning, trading inference-time compute for better reasoning performance.

Reasoning is one of the emergent abilities of advanced LLMs when scaled large enough. It can be regarded as exploration in *semantic space*, where an agent actively explores possible solutions with *self-determination*, since complex reasoning process often involves diverse sampling and self-verification (the second phase of self-determination following intrinsic motivation), until reaching a semantic consistency. Since LLMs become the core of nowadays intelligent agents, the enhancement of their reasoning ability directly amplifies the usability and robustness of AI agents, in no matter virtual or embodied environments. In **Chapter 5**, LLMs are utilized to reason and make decisions with the contextual information of the environment. However, the reasoning ability of LLMs is still limited, especially when it comes to complex tasks that require multi-step reasoning.

With the impressive performance of reasoning models such as OpenAI-o1 [Ope24] and DeepSeek-R1 [Dee+25; Dee+24], as well as other emerging counterparts, the development of advanced large *reasoning* language models, whether through fine-tuning or inference-time scaling [Sne+24; Mue+25; Gei+25; Liu+25b], has become a prominent research focus. The method proposed in this chapter follows the latter approach.

Advancements in LLMs have demonstrated remarkable generalizability across a wide range of domains. However, their reasoning capabilities, particularly in complex tasks requiring multi-step reasoning, remain a significant challenge. Recent work on enhancing reasoning in LLMs, especially in domains such as mathematical problem solving, typically follows one of two major approaches:

- ▶ Large-scale Reinforcement Learning (RL), represented by DeepSeek-R1-Zero [Dee+25], which deploys solely RL algorithm with rule-based reward functions¹ to explore the reasoning space, resulting in strong reasoning models with an emergent “aha moment”, where the model presents rethink patterns in an anthropomorphic tone.
- ▶ Inference-time scaling (or test-time scaling), which refers to ways of encouraging LLMs to explore solution space with additional computational resources during the inference phase (*i.e.* when the model is used to generate outputs) to improve

¹: Instead of using a pre-trained reward model in, for example, Internally Rewarded Reinforcement Learning (IRRL) [Li+23c], Reinforcement Learning from Human Feedback (RLHF) [Chr+17; Cas+23; Ouy+22], or Reinforcement Learning from AI Feedback (RLAIF) [Lee+24] paradigms, as a guarantee of reward signal reliability.

the quality of responses, without tuning the model itself. A simple yet efficient example is Chain-of-Thought (CoT) prompting [Wei+22a], which leverages the in-context learning of LLMs to ability invoke a series of intermediate reasoning steps before reaching a final answer.

While the RL-based approach is powerful, it demands substantial data and computational resources. In contrast, inference-time scaling is more efficient and broadly applicable to existing models. However, a key limitation of current inference-time methods is their limited integration of symbolic reasoning, such as logic-based principles, into the reasoning process. Despite the vast amount of internalized knowledge in LLMs, they often fail to utilize this knowledge effectively to construct coherent and logically consistent reasoning chains. Prior research in context distillation has focused on extracting internal knowledge to enhance preference alignment, typically by eliciting contrastive responses using specially designed prompts [Yan+24; Li+25b]. Yet, efforts to distill knowledge specifically for improved reasoning remain underdeveloped. To address this gap, this chapter introduces **Logical Thoughts (LoT)**, a method designed to elicit logically contrastive reasoning traces and fuse them into a coherent, unified reasoning chain.

LoT is an inference-time scaling method built upon CoT, which generates a sequence of intermediate reasoning steps leading to a final answer. However, longer reasoning chains are often more susceptible to error propagation [Wu+25]; that is, a single mistake in an intermediate step can compromise the entire reasoning process. This vulnerability motivates the incorporation of a self-determined verification mechanism to assess and refine each step of the chain. Particularly, LoT introduces *Reductio ad Absurdum* to systematically verify and correct reasoning steps in a step-by-step manner. Experimental evaluations conducted on language tasks in diverse domains, including arithmetic, commonsense, symbolic, causal inference, and social problems, demonstrate the efficacy of enhanced reasoning by logic. The implementation of LoT is publicly available at: <https://github.com/xf-zhao/LoT>.

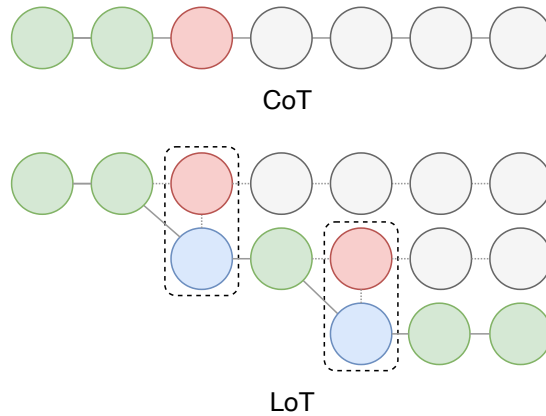


Figure 6.1: An overview of CoT (chain-of-thought prompting, [Wei+22a]) and LoT (ours). In CoT, the failure of entailment (●) makes the rest of the deduction untrustworthy (●), impeding the overall success of the deduction. In contrast, LoT is designed to think-verify-revise: it adopts those who pass the verification (●) and revises (●) those who do not, thereby effectively improving the overall reasoning capability.

6.1 INTRODUCTION

LLMs exhibit remarkable capabilities in handling tasks that require common sense reasoning or specialized domain knowledge, often giving the impression of near-omniscience. Their effectiveness has been demonstrated across a wide range of domains beyond traditional language processing [Bub+23; Yao+23b; Ahn+22; Zha+23c]. However, one major problem residing in generative LLMs yet to be solved is their tendency to hallucinate wrong statements in a confident style [Ban+23]. A quick example can be found by asking a non-internet-based LLM about very recent news, *i.e.* it will too easily make up facts without hesitation.

An educated human with expertise in logical reasoning can systematically examine words before coming to a conclusion. Unlike logical reasoning by humans, the logical incompetence of deductions by LLMs makes their decisions untrustworthy. LLMs may have a large number of logical concepts and tricks available, but fail to actively utilize them in an organized manner. However, principles in logic well-adapted by humans can leverage the reasoning ability of language models. Take a simple logic reasoning task as an example: *“If Tom plays football outside, then John will also join to play; if John plays football, then Mary won’t go outside. Knowing that Mary is outside, is Tom playing football?”* Nine out of ten answers from ChatGPT² will conclude that *“we cannot conclude whether Tom is playing football or not”*. However, with the help of the knowledge that *the contrapositive holds the exact same truth value as the original proposition*, we may prompt ChatGPT to *“use contrapositive”* to reason in another way. Then it deduces correctly: *“... Using the contrapositive of the first statement, if John does not join to play (which we have deduced), then it implies that Tom does not play football outside. Therefore, based on the given information and the contrapositives, it can be deduced that Tom is not playing football.”* Even though logical concepts are not new to an LLM, the model initially struggles to incorporate them. See Figure C.1 in Appendix C.1 for the full conversation.

By prompting an LLM to utilize *logical equivalence*, *i.e.* expressing premises that are logically equivalent but phrased differently in natural language, the original statements can be reformulated in diverse ways, effectively fostering the exploration of potential solutions. Motivated by the reasoning process in logic, we propose the **Logical Thoughts** (LoT) prompting framework, a fully automatic reasoning paradigm, to further self-improve the zero-shot reasoning³ ability of LLMs, which not only lets an LLM *think step by step* but also verify, step by step, according to the guidance via the principle of *Reductio ad Absurdum*, and revise the reasoning chain if necessary to guarantee a sound inference (see Figure 6.1 for an overview).

6.2 RELATED WORK

In order to unleash the power of a pre-trained generative language model, the quality of the interaction prompts plays an important role.

2: <https://openai.com/blog/chatgpt>

3: Under the setting where no exemplars are provided in the prompts for in-context learning.

6.2.1 Chain-of-Thought Prompting

Prior works show that LLMs have the ability to solve complex tasks but require a proper strategy to unleash this ability, *e.g.* human-in-the-loop alignment tuning [Ouy+22] and Chain-of-Thought (CoT) prompting [Wei+22a]. To generate a chain of thought that decomposes the original problem into several small parts that a language model can easily handle, CoT creates few-shot exemplars of a detailed reasoning path that it lets the model follow. Least-to-most [Zho+23] explicitly prompts the LLM to divide complex questions into sub-problems and solve them one by one. Moreover, zero-shot-CoT [Koj+22] showcases the impressive effectiveness of simply attaching the sentence “Let’s think step by step.” before any reasoning trace starts.

We build our approach under a zero-shot setting and integrate zero-shot-CoT as a baseline to compare against. While existing CoT-based methods focus on encouraging the reasoning step to be concrete, but lack supervision of their faithfulness, we propose a step-by-step verification mechanism.

6.2.2 Variational Prompting

As an auto-regressive model, the output of an LLM can be influenced by its input. Therefore, there are many research endeavors on prompt variations. Summarizing existing works, the reasoning procedure benefits from prompts that (1) are *relevant* to the reasoning task, (2) are *diverse* in expression, (3) lead to *decomposition* of complex tasks, (4) suggest *grounding* with known facts, and (5) result in *progressive revision* of reasoning steps. In the design of LoT prompting, we selectively adopt these effective prompt properties.

Relevance

An LLM can be easily distracted by irrelevant words in the prompt. A pre-selection of context enhances the correctness of reasoning [CSH23; CS22; Lin+23]. Previous works typically resort to an LLM (which can be either the LLM to train or an independent one) to evaluate the relevance of facts and infer with the ones that contribute to a reasoning step [CSH23; Lin+23]. Our verification of each reasoning step is conducted by prompting LLMs to select relevant premises to deduct from.

Diversity

The collective intelligence from a set of reasoning paths (typically, sampling N times) helps produce a reliable answer that is nearly consistent among these variants. Despite the N -times increased cost, this ensemble approach has been widely combined with other techniques for higher accuracy [Li+23b; Lin+23; Yao+23a; Zhe+23]. A single reasoning trace may be biased. In order to produce a set of reasoning candidates, previous works resort to generating samples several times with the same prompt [Wan+23b], or creating diverse prompts in the beginning for variants

[Li+23b]. However, the ensemble-based approaches are both costly and inefficient. The performance of their majority voting strategy can be limited since it is not a guided, in-depth thinking strategy.

Decomposition

Automatically decomposing complex questions into simpler sub-questions enhances the reliability and interpretability of reasoning processes, reducing reasoning errors and increasing consistency. This strategy has been shown to significantly improve performance in LLMs, as evidenced by the success of techniques such as Least-to-Most Prompting [Zho+23], Zero-shot CoT prompting [Koj+22], and other structured prompting methods [Yao+23a; Wei+22a]. Decomposition also aligns well with human problem-solving strategies, where breaking down a problem often leads to a clearer understanding.

Grounding

External functions, *e.g.* a third-party calculator for mathematical problems [Sch+23], information acquisition from Wikipedia [Yao+23b], or an affordance evaluation in robotics [Ahn+22], can ground the generation to be meaningful. This verification can be triggered under a specified condition or always be applied to the reasoning process [Lig+24; Lin+23; Li+23b]. LoT is primarily inspired by a logical standpoint to ground LLM generations with logical principles, empowering an LLM to argue different possibilities. It suggests verification and also introduces revisions of the suspected reasoning steps.

Revision

Revision (or refinement) can be regarded as a special kind of *diversity*, but it is conditioned on the previous generation that provides hints. It re-examines the words with a focus on their quality in terms of, for example, validity and conciseness [Mad+23; Zhe+23; Wel+23]. It is an iterative generating process conditioned on previous content. Many previous works actually benefit from this manner, though not explicitly mentioned. For example, Progressive-Hint Prompting [Zhe+23] generates consistent answers by progressively guiding the LLM with hints of accumulated possible answers. It repeats the generation until the answer is deemed consistent with the previous. Other works generate content conditioned not only on the previous content but also on extra feedback [Mad+23]. To obtain a revision with high quality, this guiding feedback should be specific and actionable. LoT avoids unnecessary duplicating on non-controversial reasoning steps and only revises steps deemed implausible, resulting in a chain that grows only when required (Figure 6.1 blue circle). Besides, we employ a post hoc explanation [Jun+22] to provide constructive suggestions for purposeful revisions.

6.2.3 Neurosymbolic Models

Neurosymbolic models combine neural networks with symbolic representations and reasoning techniques [WS00b; WS00a; GL20; Sar+22]. Their success stems from their ability to leverage symbolic (structured) knowledge to enhance learning or reasoning [Sar+22; GL20; Nye+21]. Unlike end-to-end black-box frameworks, these neurosymbolic models are more interpretable and explainable because of their transparency.

There exist works that adopt concepts from symbolic logic [Agl12] to establish a reliable reasoning path [CSH23; Jun+22]. To solve binary question-answering problems, it has been proposed to generate a post hoc explanation graph for a statement and compute the relative relations to formulate a symbolic logic expression [Jun+22]. The truth of the statement is thereby assigned by solving the satisfiability problem of this symbolic expression. The LoT framework employs a controlled prompting strategy that leverages logic rules and post hoc arguments to enhance error detection.

6.3 METHOD: LOT

As demonstrated in the contraposition example presented in § 6.1, when known logical rules are utilized to achieve a *logical equivalence*, the resultant distinct natural language expression affords LLMs a chance to engage in reasoning from an alternative perspective.

A challenge is that the language model has to identify the inherent logical structures first to know whether certain prior knowledge can be effectively applied. Moreover, transforming everything from the real world into a symbolic expression is unrealistic. The applicable scenario is limited because questions in many reasoning fields beyond logic, *e.g.* mathematics problem solving, can hardly be expressed in symbolic logic. Nevertheless, there is promise in incorporating concepts from logic that contribute to the process of argument proof in order to construct a neurosymbolic framework [GL20; CSH23] that facilitates a *causal* reasoning trace, *i.e.* the premises and leading thoughts entail the thoughts that follow. Continuing with the success of “let the model talk”, *e.g.* “let’s think step by step” in zero-shot-CoT [Koj+22], we further propose to guide the conversation with logic for exploration of solutions. See Figure 6.2 for the guiding diagram.

6.3.1 Reductio ad Absurdum

Self-checking is a challenging task for LLMs [Lin+23; Hua+24], and humans may also struggle with it. In logic, an effective technique to establish a claim is known as *reductio ad absurdum*, which involves an initial assumption and consequent derivation of absurdity or contradiction.

Let P and Q denote two propositions. The relation between a premise P and its conclusion Q can be expressed as $P \vdash Q$. Here “ $\vdash$ ” is a syntactic turnstile which

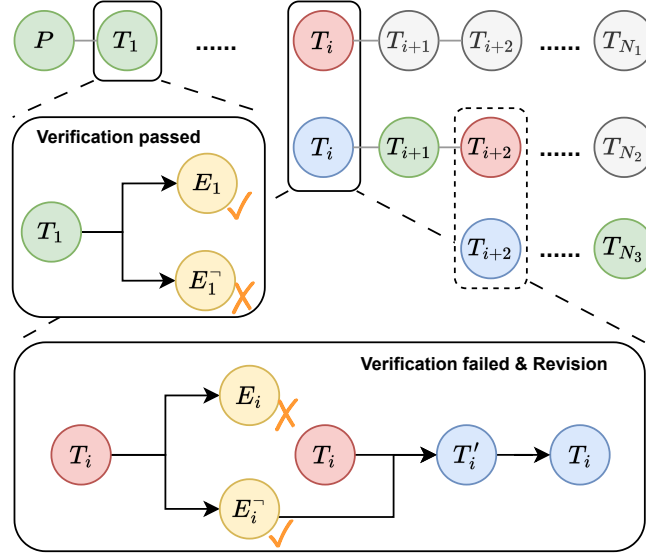


Figure 6.2: A diagram demonstrating the *think-verify-revision* loop of LoT. The two zoomed-in boxes display the processes when a thought passes (top-left) and fails (bottom) the verification, respectively. A thought passing the verification is kept in the reasoning trace, while a thought failing the verification is revised, and a new chain of thought is generated based on the revision. The symbols in this figure are introduced in § 6.3.2 and § 6.3.3. See also Figure C.2 in Appendix C.5 with extended details.

means Q is a syntactic consequence of P [Agl12], *i.e.* there exists a proof that claims the conclusion Q given the premise P . In order to prove Q using *reductio ad absurdum*, let us assume its negation $\neg Q$ is valid and then check the contradiction⁴ of the *conjunctive proposition*

$$C = P \wedge \neg Q, \quad (6.1)$$

where “ $\wedge$ ” is a binary *conjunction operator*, meaning the truth of the conjunction requires the truth of both sides. Upon the contradiction of the co-existence of the P and $\neg Q$, $P \vdash Q$ is thus proved true, and then we can claim the validation of the conclusion Q given the premise P .

Many logic principles, *e.g.* the contraposition mentioned in § 6.1 (see Appendix C.2 for a proof), can be derived by deductions following this rule. This thinking paradigm helps humans check arguments carefully before composing a conclusion. As we will demonstrate later, the reasoning ability of LLMs can also be improved by benefiting from this paradigm. The next subsection, § 6.3.2, presents the prompting and verification process for individual reasoning steps. Based on the verification result, the chain either proceeds (ignoring intermediate verification) or resets (discarding both the remaining original steps and intermediate verification thoughts).

6.3.2 LoT Prompting

There is evidence that a series of coherent explanations helps an LLM to unleash its reasoning ability [Wei+22a; Koj+22; Zho+23], while discouragement on its utterance, *e.g.* prompts like “just tell me the result without any explanation”, negatively impact

⁴: A proposition is considered contradictory if and only if it is false under every valuation.

on an LM's reasoning ability. So we elaborate on the success of an explicit reasoning process.

A typical N -step reasoning trace can be expressed as $\{P, T_1, \dots, T_N\}$, where P is the known premise and T_i is the i -th step of thoughts that originates from the output of a vanilla CoT.⁵ Usually, T_N concludes the thoughts and answers the specified question. Unfortunately, LLMs hallucinate. LLMs usually generate content autoregressively, which means the generation of T_i is based on the former content $\{P, \dots, T_{i-1}\}$. Errors in T_i will propagate and gradually influence $T_{i'}$ for increasing $i' > i$, making the successive deductions and ultimately the final conclusion untrustworthy (cfr. Figure 6.1). Therefore, we propose a verification loop to double-check each reasoning step. Following Equation 6.1, this double-check procedure unrolls by checking the validity of $P, \dots, T_{i-1} \vdash T_i$, i.e. the contradiction of

$$C_i = P \wedge T_1 \wedge \dots \wedge T_{i-1} \wedge \neg T_i, \quad (6.2)$$

once $T_{<i}$ passed the verification. If any step T_i fails the verification, this implies that the premises and previously verified thoughts $T_{<i}$ do not entail T_i . In this case, $T_{\geq i}$ needs to be revised.

To negate T_i by an LLM, a straightforward way is to format $\neg T_i$ as “It is false to say T_i ” or to give to the LLM an instruction of “Negate T_i ”. Then, the LLM has to further identify possible contradictions in C_i (Equation 6.2).

We have the following two prompting implementations for the proposal of logic-based candidates to assist self-verification:

- *Cmps-LoT*. Given that T_i is articulated in natural language and can span multiple sentences, we aim to derive a more insightful negation by adopting the chain-of-thought methodology. Here, we task the model with *composing* a post hoc explanation⁶, E_i^- of $\neg T_i$ [Jun+22], and then prompt the LLM to check the validity of $C'_i = C_i \wedge E_i^-$ instead of just C_i . We call this simple approach *Cmps-LoT*.
- *Adpt-LoT*. Considering that a logical error in a text generated by an LLM is hard to spot by the LLM itself [Hua+24], we additionally propose to alleviate the difficulty in verifying T_i by generating a pair of post hoc explanations E_i and E_i^- of T_i and $\neg T_i$ respectively, and let the LLM decide between $T_i \wedge E_i$ and $\neg T_i \wedge E_i^-$ and *adopt* one of the two. We call this full approach *Adpt-LoT* or *just LoT*⁷.

An LLM is then often biased by the prompt and, as a result, generates an explanation consistent with the prompt. Because of this “compulsory” behavior, once a statement is deemed *false* in the leading prompt, the LLM tries hard to discover errors even if they are less obvious. LoT gains advantages from the mandatory error-detection

⁵: In practice, we employ the prompt “Let’s think step by step.\n # 1.” to guide an LLM to generate reasoning steps in a consistent format, each leading with a *number* “# 1.”, “# 2.”, etc.. Subsequently, *regular expression* rules can be easily applied to split these steps into atomic units T_{number} .

⁶: A post hoc explanation is an explanation completed by the LLM with a prompt like “ T_i is true because” or “ T_i is false because”.

⁷: This strategy works on the hypothesis that the discrimination choice (from two opposite, post hoc arguments that are already listed there) is more trustworthy than the one that LLMs compose from scratch.

behavior. Additionally, when transitioning from the Cmps- to the Adpt- variant, the problem transforms into a preference discrimination task [Sau+22], featuring more balanced reviews for both T and $\neg T$ and being more manageable.

Algorithm 2: Adpt-LoT Reasoning⁸

input : *Problem/Premise* P , LLM model

output: *Verified thoughts collection* $\mathcal{T}$

```

1 Initialize  $\mathcal{T} \leftarrow \{P\}$  ;
2  $T_1, T_2, \dots, T_N \leftarrow \text{RegEx}[\text{LLM}(\mathcal{T})], i \leftarrow 1$  ;
3 while  $i \leq N$  do
4    $E_i^- \leftarrow \text{PostHocLLM}(E|\neg T_i; \mathcal{T})$  ;
5    $E_i \leftarrow \text{PostHocLLM}(E|T_i; \mathcal{T})$  ;
6    $\hat{E} \leftarrow \text{LLM}(E_i; E_i^- | \mathcal{T})$  ;
7   if  $\hat{E}$  is  $E_i^-$  then
8      $T'_i \leftarrow \text{LLM}(T|\mathcal{T}; T_i; E_i^-), T_i \leftarrow T'_i$  ;
9      $\{T_{>i}\}_{N'} \leftarrow \text{LLM}(\mathcal{T} \cup T_i), N \leftarrow N'$  ;
10   $\mathcal{T} \leftarrow \mathcal{T} \cup T_i, i \leftarrow i + 1$  ;
```

6.3.3 Chain Growth

In order to investigate a step T_i , LoT masks out all of the trailing thoughts $T_{>i}$ and branches out for revision T'_i conditioned on $\{T_{\leq i}, E_i^-\}$. Since precise feedback is important to the success of revision [Mad+23], we also encourage the LLM to revise any inappropriate thought with the advice to reason “why it is wrong”, *i.e.* E_i^- . Then, an adapted chain with a new conclusion can be re-generated based on the concatenation of the verified thoughts so far, *i.e.* $\{T_{<i}, T'_i\}$. This loop continues until the final conclusion passes the verification, which results in a chain with all the nodes being verified (see Figure 6.2 for an abstract depiction and Figure C.2 for a detailed example). Note that this chain grows only when required. Algorithm 2 shows the pseudo-code of the function to compute the reasoning trace of (Adpt-)LoT. The pseudo-code for Cmps-LoT can be found in Algorithm 5, in Appendix C.3 on page 145, where the distinct procedure for identifying contradictions is emphasized.

6.4 EXPERIMENTS AND RESULTS

For the following reasons, we carry out the experiments in a zero-shot setting: 1) Zero-shot-CoT [Koj+22] has a wide task-agnostic application potential, while few-shot requires domain knowledge; 2) The few-shot prompts heavily influence the performance even on the same dataset, so it is hard to evaluate fairly, as the prompt varies. Drawing direct comparisons with other prompting works in the literature is

challenging due to variations in task settings and backend language models. Many of these works are specifically under a few-shot setting, which would necessitate additional modifications to adapt them for zero-shot reasoning. We consider this an area for future investigation.

Since our work is an enhancement on the chain produced by zero-shot-CoT [Koj+22], we compare LoT with it as the baseline to demonstrate the benefit of step-wise verification and revision for zero-shot reasoning. We evaluate the accuracy of tasks in various domains as the overall performance measure and also report the impact of the logical revision on the original reasoning chain.

We aim to answer the following research questions regarding **Objective III** by conducting experiments:

- ▶ **R.Q. 6.1** Does LoT outperform the original zero-shot CoT, *i.e.* logic-guided inference enhances reasoning ability, in various domains as well as with LLMs of varying model scales?
- ▶ **R.Q. 6.2** What is the impact of LoT on individual reasoning chains (*e.g.* revision frequency, length)?
- ▶ **R.Q. 6.3** Do post-hoc explanations help LLM self-check?

6.4.1 Experimental Setup

Dataset

We demonstrate the effectiveness of LoT on diverse language topics:

- ▶ Math reasoning tasks GSM8K [Cob+21] and AQuA [Lin+17]. The GSM8K dataset contains grade school mathematics questions that should be responded to by numerical answers; AQuA has more advanced questions, but has several optional answers to choose from.
- ▶ Commonsense reasoning tasks DateUnderstanding and OddOneOut [Sri+23]. The DateUnderstanding task necessitates the utilization of both common sense and fundamental arithmetic calculations to find out the correct date, making it sufficiently challenging to prevent it from being solvable through simple one-step reasoning. The OddOneOut requires common sense to deduce the unusual object in the context.
- ▶ Causal inference tasks CauseEffect and ShuffledObjects [Sri+23], where both of the tasks require reasoning from the context for a correct deduction.
- ▶ Symbolic reasoning task LastLetter [Sri+23]. In this task, the language model has to extract the last letter of given candidates and concatenate them in order, which is simple for humans but challenging for language models because of tokenization [Mie+21].
- ▶ Social interaction reasoning task, SocialQA [Sri+23], that measures the model's emotional and social intelligence in human daily activities. Completing the task requires an understanding of human behavior.

To get a formatted answer that can be directly compared with the ground truth in the aforementioned dataset, a final prompt asking for the final answer is attached

after the reasoning trace, *e.g.* for the GSM8K dataset, we simply attach “Therefore, the final numerical answer is:” at the end. For robustness, this answer is matched with a regular expression to extract only numerical digits before comparing it with the ground truth.

Backend LLMs

Previous works show that the performance improvement of the CoT technique varies when applied to language models of different capabilities [Wei+22a; Koj+22]. Therefore, we conducted an evaluation of the LoT method using a range of models, including Vicuna-7b, Vicuna-13b, and Vicuna-33b models [Chi+23], as well as GPT-3.5-turbo and GPT-4. The Vicuna model is an open-sourced language model trained by fine-tuning LLaMA [Tou+23] on user-shared conversations. It demonstrates strong performance across various scenarios and offers flexibility in terms of model size selection. On the other hand, GPT-3.5-turbo and GPT-4 are larger models known for their state-of-the-art performance in numerous tasks.

To ensure stable results and promote self-error detection within the models, we set the temperature parameter to 0.1. Additionally, the `max_token` parameter was established at 2048, a sufficient limit to accommodate all the datasets employed in our evaluation.

6.4.2 Analysis

RESEARCH QUESTION 6.1 Does LoT outperform the original zero-shot CoT, *i.e.* logic-guided inference enhances reasoning ability, in various domains as well as with LLMs of varying model scales?

To answer the first question, we conduct zero-shot experiments with datasets covering more diverse topics and with language models of different sizes. The LoT-enhanced performance is compared with the zero-shot baseline in Table 6.1. The experiment shows that LoT can enhance the performance of the base CoT in various domains. The performance benefits are more consistent when the model size gets considerable (>7B parameters). Moreover, the performance gain becomes more prominent as the model’s ability increases (*e.g.* GPT-4).

RESEARCH QUESTION 6.2 What is the impact of LoT on individual reasoning chains (*e.g.* revision frequency, length)?

We report more insightful case-wise statistics and discussions in this section, including

- ▶ average revision frequency in Table 6.2;
- ▶ the resultant number of reasoning steps in Table 6.3;
- ▶ and a case study to illustrate the logical reasoning procedure. More detailed statistics including the worsening rate (*i.e.* the ones being originally correct by

CoT but “corrected” to be wrong by LoT) and improvement rate (*i.e.* the ones that are originally wrong and being corrected by LoT) can be found in Table C.1, in Appendix C.4 on page 146.

Revision Frequency. In order to measure the complexity of revisions, we list the average revisions *per chain* in Table 6.2 and typical reasoning steps required by CoT and LoT in Table 6.3. Note that the number of steps is not human-defined or prompted since our setting is in zero-shot, so the language models decide by themselves the length of a reasoning chain. The percentage of revisions indicates the frequency of LoT to revise the candidate’s reasoning chain. We observe that language models with powerful capabilities (*e.g.* GPT-4 and GPT-3.5-turbo) are more active in revision than smaller models, and challenging tasks such as the math reasoning task lead to more revisions. However, revision does not necessarily alter the final deduction answer. For example, LoT with GPT-3.5-turbo backend revises several times on SocialQA yet achieves the same accuracy as CoT. Intuitively, solving one problem may lead to multiple pathways, and some revisions might aim to enrich the sentence by incorporating additional known conditions and rhetorical supplements.

Resultant Steps. The average step count is the number of valid reasoning steps in the final CoT and LoT reasoning paths (*i.e.* the intermediate verification, refinement, *etc.* are not shown). From Table 6.3, we can conclude that 1) larger language models generally generate longer chains and are also more active in revision; 2) the LoT *refined*⁹ reasoning chain is typically slightly shorter than the original zero-shot CoT

Table 6.1: We evaluate the accuracy of our methods and compare them against baseline approaches using various models and datasets, with computation based on ground truth annotations. The percentage difference of CoT [Koj+22] without (✗) and with (✓) LoT enhancement using different LLMs is shown below each cell group (green if positive, red if negative). CoT generally gains better performance when being enhanced by LoT. Larger models, such as GPT-4, exhibit strong and robust self-correction ability.

	LoT	GSM8K	AQuA	Date	SocialQA	Cau.Eff.	Objects	Letter	OddOut
Vicuna-7b	✗	17.52	21.65	7.24	37.00	52.94	34.00	0.00	25.58
	✓	17.68	20.47	7.24	36.50	52.94	35.00	0.00	25.58
		(+0.16)	(-1.18)	(0.00)	(-0.50)	(0.00)	(+1.00)	(0.00)	(0.00)
Vicuna-13b	✗	33.79	22.05	32.31	41.00	68.75	31.00	2.00	29.07
	✓	37.56	23.62	33.15	48.50	68.75	31.50	4.00	45.35
		(+3.77)	(+1.57)	(+0.84)	(+7.50)	(0.00)	(+0.50)	(+2.00)	(+16.28)
Vicuna-33b	✗	40.33	26.38	15.70	37.50	52.94	32.00	14.67	40.70
	✓	40.49	29.53	20.35	47.50	68.75	34.50	14.00	43.02
		(+0.16)	(+3.15)	(+4.65)	(+10.00)	(+15.81)	(+2.50)	(-0.67)	(+2.32)
GPT-3.5-turbo	✗	78.75	57.09	51.26	72.00	92.16	60.75	67.33	81.40
	✓	80.15	60.63	52.37	72.00	92.16	58.25	67.33	81.40
		(+1.40)	(+3.54)	(+1.11)	(0.00)	(0.00)	(-2.50)	(0.00)	(0.00)
GPT-4	✗	94.29	71.56	83.09	77.50	100.00	100.00	92.61	95.35
	✓	95.71	74.31	85.16	77.50	100.00	100.00	93.14	96.51
		(+1.42)	(+2.75)	(+2.07)	(0.00)	(0.00)	(0.00)	(+0.53)	(+1.16)

⁹: Note that LoT ultimately produces a clean reasoning chain containing only *valid* steps, excluding

Table 6.2: The average step-wise revision frequency is presented as a percentage, reflecting how often a reasoning step is revised by LoT.

Revision $\cup$	GSM8K	AQuA	Date	SocialQA	Cau.Eff.	Objects	Letter	OddOut
Vicuna-7b	2%	4%	2%	1%	2%	0%	3%	0%
Vicuna-13b	7%	10%	5%	5%	0%	7%	2%	0%
Vicuna-33b	2%	9%	8%	7%	6%	9%	1%	7%
GPT-3.5-turbo	16%	28%	32%	5%	20%	9%	4%	16%
GPT-4	3%	20%	7%	2%	0%	1%	0%	8%

Table 6.3: The average number of resultant reasoning steps without (✗) and with (✓) LoT applied.

	LoT	GSM8K	AQuA	Date	SocialQA	Cau.Eff.	Objects	Letter	OddOut
Vicuna-7b	✗	1.22	1.16	1.34	1.09	1.00	2.54	3.46	1.00
	✓	1.27	1.21	1.35	1.10	1.02	2.54	3.49	1.00
Vicuna-13b	✗	2.81	2.89	5.06	2.69	1.00	2.93	1.66	1.00
	✓	2.74	2.87	5.05	2.71	1.00	2.96	1.69	1.00
Vicuna-33b	✗	1.94	1.99	2.31	3.26	1.00	3.26	1.20	1.70
	✓	1.94	1.91	2.33	3.13	1.06	3.23	1.21	1.64
GPT-3.5-turbo	✗	4.17	6.83	3.66	2.50	1.73	3.02	4.84	1.57
	✓	4.08	6.24	3.56	2.51	1.92	3.05	4.81	1.70
GPT-4	✗	3.42	4.22	2.71	2.33	1.00	3.00	4.05	1.00
	✓	3.41	4.39	2.72	2.33	1.00	3.01	4.05	1.08

output. Our conjecture is that this phenomenon might arise because, during the refinement process, the language model strives to incorporate additional information, consequently yielding concise chains of reasoning.

RESEARCH QUESTION 6.3 Do post-hoc explanations help LLMs self-check?

In order to examine the error-finding ability, the results of using LoT and its ablated variants on three challenging tasks are shown in Table 6.4, where “Self-Check” is an ablation of *reductio ad absurdum* such that the LLM is only prompted to “double check the step” without logical hints.

With LoT, the complex examination problem is broken down into two procedures: 1) the post hoc explanation mechanism for rigorous error detection, and 2) a single-choice preference for selecting the more reasonable opinion from the two candidates. The better performance of (Adpt-)LoT suggests that when it comes to error detection in LLM’s reasoning, it is more effective for an LLM to embrace one of two opposing viewpoints (T, E or $\neg T, E^-$) rather than *composing* the verification directly (which is the case for both “Self-Check” and Cmps-LoT), especially when coping with tasks that are difficult such as math reasoning.

Table 6.4: Zero-shot accuracy results (in %) in comparison of LoT (*i.e.* Adpt-) and its ablated variants. The underlying LLM is GPT-3.5-turbo.

Method	GSM8K	AQuA	Date
CoT	78.75	57.09	51.26
Self-Check	76.15	56.19	51.57
Cmps-LoT	77.67	57.48	52.37
LoT	80.15	60.63	52.37

Case Study

We show a successful case on the Date Understanding task to demonstrate the verification and revision procedure applied to the chain of thoughts initialized by zero-shot-CoT. (See [Appendix C.6 on page 147](#) for detailed prompts and further case studies.)

Snippet A

Question: Today is Apr 10, 1985. Jane’s appointment will be 3 days later. What is the date 10 days ago in MM/DD/YYYY?
 Let’s think step by step.
 Answer:
 #1. Today is Apr 10, 1985.
 #2. Jane’s appointment will be 3 days later.
 #3. So, Jane’s appointment will be on Apr 13, 1985.
 #4. To find the date 10 days ago, we subtract 10 days from Apr 13, 1985.
 #5. Apr 13, 1985 - 10 days = Apr 3, 1985.
 #6. Therefore, the date 10 days ago is Apr 3, 1985.

In Snippet A, there are initialized zero-shot-CoT reasoning steps where step #6 is actually incorrectly inferred (colored in red). Here, *black* color indicates given context or fixed prompts; *non-black* color indicates generated content by the LLM. The error occurs because zero-shot-CoT is distracted by the irrelevant premise of “Jane’s appointment will be 3 days later” and concludes with a wrong answer.

Snippet B

Verification of the next step:
 #6. Therefore, the date 10 days ago is Apr 3, 1985.
 Review X: <review>step #6 is TRUE because we are subtracting 10 days from the given date of Apr 13, 1985. By subtracting 10 days, we go back in time and arrive at Apr 3, 1985. Therefore, the date 10 days ago is Apr 3, 1985. </review>
 Review Y: <review>step #6 is FALSE because the reasoning in step #4 is incorrect. To find the date 10 days ago, we need to subtract 10 days from Apr 10, 1985, not from Apr 13, 1985.
 The correct calculation would be:
 Apr 10, 1985 - 10 days = Mar 31, 1985.
 Therefore, the date 10 days ago is Mar 31, 1985. </review>

To verify the 6th step, as shown in Snippet B, the LLM is prompted to generate two opposite reviews¹⁰ given the question context and the previous five verified steps.

¹⁰: In practice, as is shown in the example case, the special language mark “<review></review>” is applied for the sake of robust separations of different reviews.

Snippet C

Let's check two different reviews (X and Y).
 Support the more plausible one and criticize the other one.
 Review X: ...
 Review Y: ...
 Let's start by analyzing one by one.
 I. What are the premises and previous steps to support the verification of step #6? (Your answer should quote the exact quote as support.)
 II. Criticise the incorrect review.
 (Note the examined step doesn't have to tackle the whole problem at once.)
 Finally, identify whether step #6 is true or false.
 Analysis and conclusion:
 I. The premises and previous steps to support the verification of step #6 are as follows:
 - Today is Apr 10, 1985.
 - Jane's appointment will be 3 days later.
 - Jane's appointment will be on Apr 13, 1985.
 - To find the date 10 days ago, we subtract 10 days from Apr 13, 1985.
 II. Review Y is correct in criticizing step #4. The reasoning in step #4 is indeed incorrect. To find the date 10 days ago, we need to subtract 10 days from Apr 10, 1985, not from Apr 13, 1985.
 The correct calculation would be:
 Apr 10, 1985 - 10 days = Mar 31, 1985.
 Therefore, the date 10 days ago is Mar 31, 1985.
 Conclusion: Step #6 is false. The correct date 10 days ago is Mar 31, 1985, not Apr 3, 1985.

Due to the content limitation, we omit the long prompt and focus on the resultant generation. Note that *Review Y* successfully detects the unobvious error and provides a correction that can be adopted later for revision.

In Snippet C, the LLM is encouraged to extract and deduct based on relevant premises. Finally, it ends with adopting *Review Y* and reaching a correct conclusion.

6.5 CONCLUSION

LLMs have impressive reasoning ability in domains that require commonsense knowledge, specialized expertise, comprehensive understanding, *etc.* However, there is still room to improve their multi-step reasoning capability. Building upon zero-shot-CoT, we derive the LoT prompting framework from a symbolic logic perspective, utilizing the widely applicable principle of *reductio ad absurdum*, resulting in a robust *think-verify-revise* framework with plausible prompting properties. Experiments conducted on a variety of reasoning tasks spanning different domains demonstrate that enhancing zero-shot Chain-of-Thought (CoT) with LoT leads to improved reasoning ability, particularly when applied to LLMs of large scale.

6.6 LIMITATIONS AND FUTURE WORK

Prompting, Fine-tuning and Generalizability. LoT establishes a controlled prompting strategy for self-correction. Nonetheless, it is worthwhile to explore future endeavors of prompting engineering as well as fine-tuning LLMs for the purpose of spontaneous

logical reasoning¹¹. Moreover, although our approach does not require a complex or abstract symbolization process, as we use contrastive prompting to elicit contradictions, assigning a binary label (true *vs* false) to a reasoning step is not always feasible and may constrain the potential for exploration.

Generation Probability. Rather than letting the LLM choose from different reviews, another possible method is to access and compare the probability of the generations. Unfortunately, there is no public access to the generation probability of GPT-3.5-turbo yet¹² as it is possible for completion models (such as text-davinci-003). Considering a cheaper price and better performance, we conducted our experiments with the chatting model and left this possibility for future work.

Zero-shot, Few-shot, and Beyond. Significant potential exists for enhancing the reliability of the verification-revision procedure, and devoting efforts to the advancement of prompt engineering may prove to be valuable and worthwhile. Since this work is done with an aim to be as generalizable as possible, the experiments are all conducted in the zero-shot setting. Nonetheless, incorporating domain knowledge into the exemplar prompt proves advantageous for enhancing performance [Koj+22; Wei+22a], it is still worthwhile to explore the potential when LoT is applied in the few-shot setting in future work. Furthermore, conducting extensive experiments across diverse domains would be instrumental in assessing the generalization capability of the proposed method, especially for those areas that require reliable deduction besides mathematics, such as legal reasoning, scientific research, ethics moral reasoning, and so on.

While our research primarily focuses on integrating human knowledge into CoT prompting, further exploration of additional logical deduction principles could enhance the reasoning process. Moreover, we demonstrate the efficacy of bolstering the robustness of complex reasoning by discerning between conflicting outputs, suggesting the potential extension of this approach to prompt and refine LLMs for self-improvement. This may entail utilizing self-checked outputs for Reinforcement Learning from AI Feedback (RLAIF) [Li+23a; Lee+24]. Such endeavors show promise, particularly in situations where a positive “Generation-Discrimination gap” (GD gap) exists [Sau+22], *i.e.* language models with promises to be further improved by discerning the quality of its generation, but we defer the investigation of this avenue to future research endeavors.

¹¹: Various prompting templates may influence the outcome a lot [Liu+25a]. In our case, for example, in mathematics problems, a prompt that leads to an active examination of numerical computation can assist the verification process [Mad+23]. A study of few-shot and of the domain-specific design of prompts for the verification-revision structure is worthwhile to explore, which we leave for future work due to the scope of this work.

¹²: <https://platform.openai.com/docs/api-reference>

AGENTIC SKILL DISCOVERY

Semantics + Policy + Self-determination

To fulfill the final **Objective IV**, *i.e.* “to construct autonomous agents with advanced self-determination that can verbally sense environmental context and discover meaningful skills from scratch in pursuit of long-term embodied autonomy”, this chapter presents a highly autonomous robot deployed in a novel environment. It actively explores and acquires new capabilities, driven by *semantic* intrinsic motivation and self-regulated learning, guided by LLMs and Vision Language Models (VLMs), resulting in *self-determined learning* of a *semantics-grounded policy*.

In **Chapter 5** and **Chapter 6**, LLMs are employed to reason about the environment in pursuit of task completion. This reflects a common paradigm in current robotic systems that largely depend on predefined objectives and human-provided knowledge. However, the capacity to reason and adapt to novel, unforeseen conditions holds significant promise for achieving greater autonomy. Such adaptation, or self-development, entails not only adjusting existing skills but also acquiring entirely new ones to handle unfamiliar situations, a capability that is essential for long-term robot autonomy with minimal human supervision.

Language-conditioned robotic skills bridge the high-level reasoning capabilities of LLMs with low-level robotic control. A remaining challenge is to acquire a diverse set of fundamental skills. Existing approaches either manually decompose a complex task into primary robotic actions in a top-down fashion or bootstrap as many combinations as possible in a bottom-up fashion to cover a wider range of task possibilities. These decompositions or combinations, however, require an initial skill library. For example, a “grasping” capability can never emerge from a skill library containing only diverse “pushing” skills. Existing skill discovery techniques with Unsupervised Reinforcement Learning (URL) acquire skills by exhaustive exploration but often yield non-meaningful behaviors. Previous evidence [Ma+24b; Kwo+23] shows that LLMs are able to program reward functions or optimization objectives for a learning-based method to optimize over, bridging high-level semantic reasoning with low-level control. However, challenges inevitably arise: existing methods cannot be directly applied to robot skill learning because they lack a success determination mechanism, *i.e.* the ability to assess whether a task has been successfully completed, or more broadly, to enable self-regulation in learning novel skills. This chapter introduces **Agentic Skill Discovery (ASD)**, which addresses this gap by enabling agents to not only explore their environment but also develop skills that support their long-term autonomy.

Specifically, a novel learning framework, which is entirely driven by LLMs, for autonomous robot skill discovery is introduced. The framework begins with an LLM generating task proposals based on the provided scene description and the robot’s configurations, aiming to incrementally acquire new skills upon task completion.

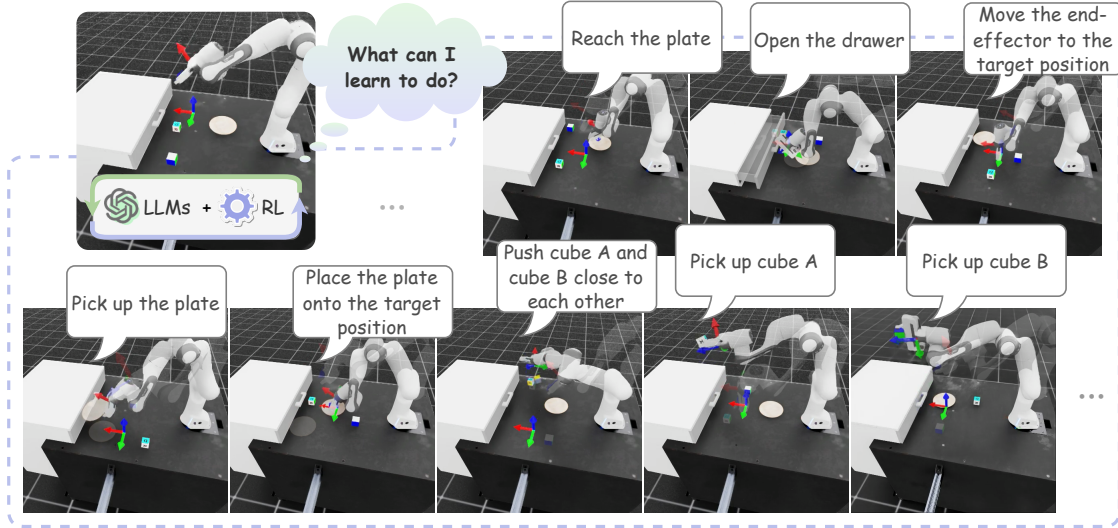


Figure 7.1: Guided by LLMs, Agentic Skill Discovery (ASD) enables robots to explore the environment and incrementally acquire contextual skills. Tasks proposed by LLMs are learned as skills via parallel RL. A high-quality VLM verifies success and ensures reliability.

For each proposed task, a series of RL processes is initiated, utilizing reward and success determination functions sampled by the LLM to develop the corresponding policy. The reliability and trustworthiness of learned behaviors are ensured by an independent VLM. This work shows that starting with zero skill, the skill library emerges and expands to more and more meaningful and reliable skills, enabling the robot to further propose and complete advanced tasks efficiently.

7.1 INTRODUCTION

“In the beginning was the Word” [CS15]. Can LLMs unleash the generative potential of words, as implied in this famous quote? In a more practical setting, can an LLM teach the skills and goals of human actions to a robot that is initialized without any skills or motivation?

Large Language Models show potential in many fields that require common sense and reasoning. Large-scale models excel because of their training on human datasets, and textual or even multimodal reasoning capabilities. LLM-based agents, especially robots, extend the potential to embodiments, but they still show limitations when applied to direct robotic control. The reasons are insufficient real-world robot data for training, as well as the diversity of topologies and physical properties. As a workaround, abstracting robot control to a certain level and referring to each abstraction as a specific “skill” helps LLMs to control robots generically [Ahn+22; Zha+23c; Zha+23a; MJS24; Wu+23; Din+23]. For example, SayCan [Ahn+22] builds a control framework that lets a robot follow a set of basic language instructions. When commanded with a complex task, an LLM decomposes the task into actionable low-level actions.

Acquiring diverse robotic skills with minimal human supervision has garnered considerable attention. However, previous methods have either attempted to chain

existing skills, relying heavily on a collection of basic skills [Du+23; Cel+23; Zha+23a], or explored from scratch but often yielded non-interpretable robot behaviors, especially those using unsupervised RL [Li+23c; Par+23; Eys+19; Sha+20]. We ask whether an LLM can encourage a robot to learn novel tasks that consist of entirely novel yet relevant skills. Imagine a robot being placed in a new environment. The robot must be motivated to explore the environment in a way to learn applicable skills before becoming ready to perform further tasks. Given the knowledge about human actions that resides in LLMs, we expect that an LLM can, by itself, suggest a variety of contextually meaningful skills for the robot to learn.

We refer to this autonomous exploration as Agentic Skill Discovery (ASD), which enables robots to interact with their environment through semantically-driven motivation and autonomously initiate requisite learning procedures. In this research, we address the challenge of open-skill learning in robotic systems guided by LLMs, wherein either LLMs or a complementary Vision Language Model (VLM) must facilitate both the learning process and success/failure evaluation without human intervention or predetermined assessment criteria (Figure 7.1 on the facing page). Our primary contributions encompass a learning approach that incorporates:

- ▶ Task proposal: an LLM iteratively proposes novel and open tasks that are suitable for the given environment (§ 7.3.1, Figure 7.2), and collects the resulting skills.
- ▶ Task completion for skill learning: existing works such as Eureka [Ma+24b] focus on solving *determined tasks* where the success/failure is defined by the human in advance, and cannot be generalized to automatic learning of open robot skills. In contrast, we introduce using LLM to propose a success determination function, which provides a criterion for optimizing the reward functions. To make open skill learning tractable, we showcase the importance of having a post-validation VLM for success determination, as a guarantee for reliability and trustworthiness (§ 7.3.2, Figure 7.4). It is essential to minimize false positives and false negatives to ensure the skill library remains reliable, avoiding the inclusion of overtrusted skills and the exclusion of useful ones (Figure 7.7). Furthermore, we apply the Retrieval Augmented Generation (RAG) technique to retrieve validated function references for the efficacy of LLM reasoning. We note that the application of RAG to open-skill learning is novel and progressive beyond the state of the art.
- ▶ Reuse of skills: to complete certain sophisticated tasks requiring long-horizon manipulations, we prompt LLMs to decompose them into a chain of sub-tasks that are either already explored or are novel to learn on demand, forming a chain of conditional skills.

7.2 RELATED WORK

Humans and animals engage in active exploration, continuously forming new representations of the world and discovering new skills. Autonomous agents must be capable of assessing not only what exists in the world but also what is learnable

and useful for future tasks. The challenge of robotic self-development is particularly relevant in open-ended, unstructured environments, where pre-programmed behaviors are insufficient. This raises an important question: *Can robots develop applicable capabilities to explore their environments in an active, self-determined manner?* If so, how can they learn what kinds of skills can be acquired within their surroundings? Inspired by human developmental processes, many autonomous systems are equipped with intrinsic motivations that encourage exploration and learning. By generating their own goals and learning tasks from a curiosity-driven perspective, robots can expand their skill repertoire beyond what was originally programmed. Despite advances in RL and self-supervised learning, current methods still struggle with open-ended skill discovery and long-term adaptation, due to the fact that many exploration techniques focus on maximizing novelty or information gain but lack mechanisms for deciding *which discoveries are meaningful for future tasks*.

This section overviews previous research works in discovering novel skills (or, maximizing empowerment) and LLM-based advancements relating to this field.

7.2.1 Skill Discovery

Acquiring diverse robotic skills with minimal or no human supervision is a key challenge in reducing human effort. Unsupervised Reinforcement Learning (URL) has emerged as a prominent research area for generating diverse behavioral trajectories that are distinguishable from one another [Li+23c; Par+23; Pen+22; Eys+19; Sha+20]. Typically, URL methods employ information-theoretic objectives during unsupervised training to maximize behavioral entropy. However, this “bottom-up” scheme (*i.e.* clustering various trajectories as skills) often results in non-interpretable and semantically meaningless skills from a human perspective, making the learned skills hard to collect and reuse. The trade-off between low costs and meaningful resulting behavior depends on the amount of human knowledge introduced, *i.e.* more supervision (high cost) generally indicates more natural robotic behaviors, and vice versa. While some works introduce further constraints relating to human demonstrations to acquire natural behaviors [Pen+22], the scalability is still limited by the need for domain-specific data collection. Recent works [Rho+25] have sought to constrain the exploration space to a meaningful subspace, where robot behaviors align with human-defined language instructions, as evaluated by LLMs. Our work discovers skills at a higher level of abstraction, *i.e.* in a “top-down” scheme where the learning objectives are ensured to be semantically meaningful in advance, producing a growing repertoire of semantically distinct skills that demonstrate greater diversity and interpretability.

Chaining known skills into new ones can substantially extend robots’ abilities and efficiencies. In order to efficiently combine basic skills in a meaningful way and avoid a combinatorial explosion, LLMs can be utilized to reason the logical ways of stacking skills to complete new, long-horizon tasks [Zha+23a; Cel+23]. When LLMs are prompted with environment contexts, such as available robot joints and object types, they can propose meaningful motivations for the next movement [Du+23; Zha+23a]. Prompting an LLM to get an “interesting outcome” can yield generally

meaningful skill combinations [Cel+23]. However, prior works primarily focus on combining pre-acquired skills to create more complex behaviors. In such approaches, the so-called “new skills” are assembled from an existing library of basic skills rather than developed from scratch. Consequently, the potential skill space is constrained by the foundation of the initial skill set. For example, a robot proficient only in “pushing” would be unable to acquire the skill of “grasping” within the confines of skill assembly. In contrast, the concept of agentic environment skill discovery, where both the expansion of high-level skill libraries and the initiation of necessary low-level skill training occur autonomously, remains underexplored. To address this gap, our work employs RL agents, supervised by LLMs and VLMs, to acquire novel low-level policies that were previously beyond the robot’s capabilities.

7.2.2 Code LLM Control

Most LLM-based robot behavior relies on pre-defined primitives (skills). This inflexible design makes it difficult to generalize to unseen objects and instructions [Ahn+22; Zha+23c]. Recent approaches let code-LLMs write programming code to complete open instructions [Lia+22; Hua+23]. In particular, VoxPoser [Hua+23] uses a VLM and LLM to construct a 3D cost map to guide a robot engaging with its surroundings. VoxPoser relies heavily on the quality of the initially composed cost map, limiting its ability to perform exploratory behaviors. In addition, it depends on a trajectory solver during inference, which can further constrain its flexibility and scalability. In contrast, ASD launches RL, letting the agent explore the environment and exploit the learned language-conditioned policies. As for the automatic learning of low-level control policies, previous approaches show that LLMs are capable of programming reward functions to optimize over, achieving remarkable performance even for complex tasks [Ma+24b; Yu+23]. However, these methods have not yet been applied to robot skill learning, where tasks are newly proposed. This presents a significant challenge, as only the language instruction is available, without predefined success or failure conditions to guide the learning process.

7.3 METHOD: ASD

The Agentic Skill Discovery (ASD) framework introduces a fundamentally different approach to robotic skill acquisition, where skills emerge purely from the interaction between a language model, a robot, and its environment, without any pre-structured human guidance such as demonstrations, reward designs, preferences, or handcrafted supervision signals. At a high level, ASD enables autonomous exploration through self-generated task instructions (§ 7.3.1), while at a lower level, it learns to master these discovered tasks as skills through RL with self-determined success criteria and reward strategies (§ 7.3.2). The framework then leverages these acquired skills to tackle long-horizon tasks and further expand its skill repertoire through task

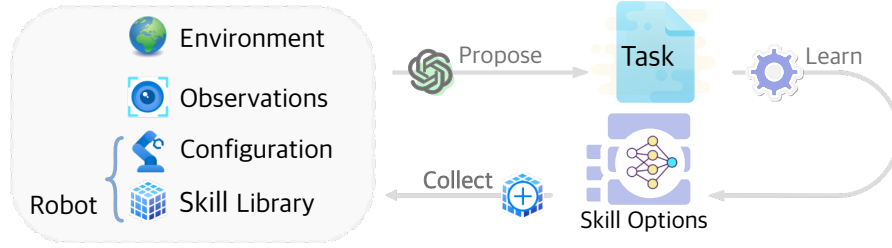


Figure 7.2: Contextual skill acquisition loop of ASD. Given the environment setup and the robot’s current abilities, an LLM continually proposes tasks for the robot to learn (see Figure 7.4 for the learning scheme). Successfully completed skills are collected as acquired skills in the skill library. The learning process for each target skill may yield multiple viable execution strategies, which we preserve as distinct skill options. For each option, we store both the learning specifications (including reward and success functions) and the corresponding trained policy networks, enabling efficient retrieval and deployment of these skills in future tasks.

decomposition (§ 7.3.3) and *on-demand skill learning*¹. Unlike traditional approaches where skill complexity and granularity are carefully engineered, ASD faces the unique challenge of operating in an unconstrained space where the language model must discover appropriate skills, determine their complexity, and establish success criteria without predefined constraints or reward structures. This represents a significant departure from existing methods, as it removes human scaffolding from the skill acquisition process while enabling truly autonomous skill discovery and composition. A pseudo-code of the ASD framework is shown in Algorithm 3 on page 108².

7.3.1 Iterative Task Proposal and Skill Collection

Instead of relying on exhaustive human efforts, ASD utilizes LLMs to propose meaningful tasks given the description of a certain scene. Those tasks will be assigned to RL agents to learn corresponding language-conditioned policies (see § 7.3.2). Figure 7.2 overviews skill acquisition by the propose-learn-collect loop.

To provide the LLM with sufficient information about the environment, we provide it with the source code of the observation space [Ma+24b]. Also, the robot configuration, such as robotic arm type and DoF, is prompted as the initial background description. Due to environment complexity and unpredictable learning challenges, we implement iterative task proposals and learning rather than allowing the LLM to propose all tasks at once. In particular, the LLM will be informed about tasks that could not be completed so that it will have a sense of the limits of the learning agent, influencing the successive task proposals. For the sake of efficiency and reusability, we encourage the LLM to propose tasks that are meaningful, atomic, independent, and incremental (see Figure D.4 in Appendix D.3 for detailed prompts).

1: For example, a new “placing” skill should be learned *on demand* when instructed to stack two cubes together, if the skill library contains only primary skills like “pushing” and “picking”.

2: For more implementation details, refer to open-sourced code at: <https://github.com/xf-zhao/Agentic-Skill-Discovery>.

Skill Options

Generally, control with various *options* for a given task has the potential to be more robust and generalizable [GRW17; Eys+19]. As will be introduced in § 7.3.2, in the process of the evolutionary search of diverse reward functions, a given task will be successfully learned by several options, forming a set of various control policies. A task will be considered completely learned, or in short complete, if the resultant agent behavior aligns with expectations; otherwise, it is deemed unsuccessful after an extended period of learning.

Completed tasks will be considered as “skills”, along with their “options”, to be stored in the skill library, and the names of attempted but uncompleted tasks will be added to a failure pool. A summary of the completion status will be generated to guide LLMs in proposing subsequent tasks, taking into account the learning curve and potential difficulties. Failed tasks are usually too sophisticated for LLMs to write reward functions to master at once. Hence, we will decompose and complete them by combining acquired and on-demand-learned skills (§ 7.3.3). We collect all options for the same skill, *i.e.* various policies paired with successful reward functions, as future execution candidates, and we leave the study of mixing various skill options for one robust skill control for future research.

7.3.2 Evolutionary Skill Learning with Fast and Slow Success Determination

LLMs are capable of composing reward functions for RL agents to accomplish specified tasks [Yu+23; Ma+24b]. We extend this strategy of Eureka [Ma+24b], which prompts LLMs to program reward functions and evolve them with deterministic selection where only the best reward function, as assessed by the success rate as a *fitness function*, will survive and mutate. See Figure 7.3 for an illustration.

Challenges in Open-Skill Learning

In open-skill learning, where the focus is on optimizing newly proposed tasks rather than predefined ones, naive Eureka-like methods [Ma+24b] cannot be directly applied due to the fact that the success criteria cannot be predetermined. This is because the fitness function, which serves as the ground truth for determining success, is unknown

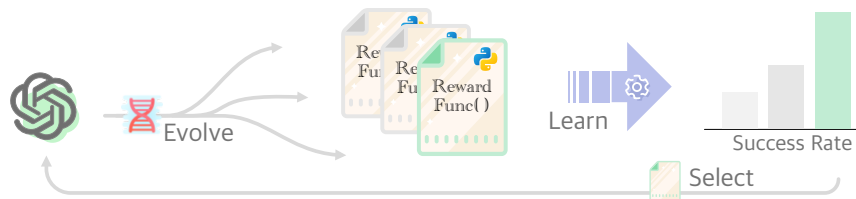


Figure 7.3: Evolutionary search of reward functions for *defined tasks with deterministic success functions*, where the success rate can be reliably computed and used as a fitness measure. However, this approach is too simple for novel skill learning, where the absence of prespecified success criteria necessitates a behavior verification mechanism (*cfr.* Figure 7.4).

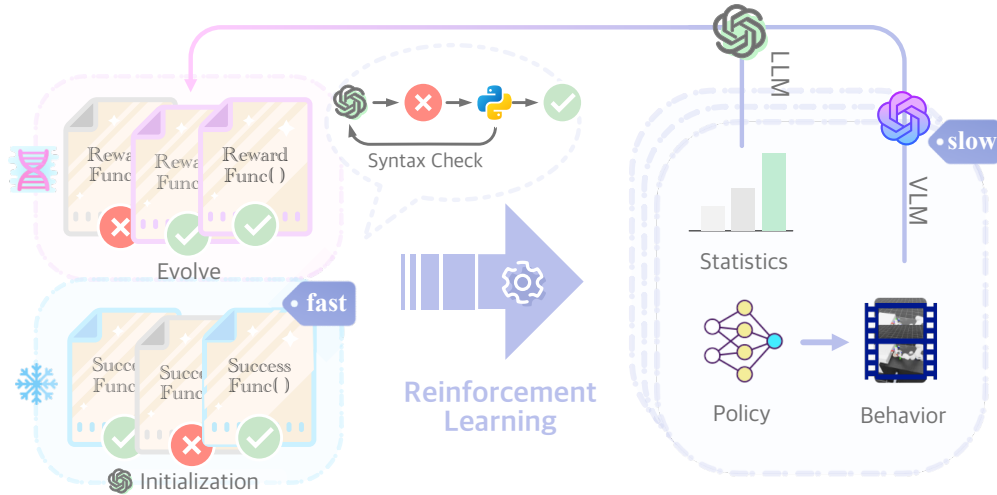


Figure 7.4: The evolutionary skill learning procedure of ASD. An LLM composes a set of reward and success functions (left), corresponding RL to train policies (middle), and evolutionary search with learning statistics (e.g. success rate) and VLM assessment (right).

for each task proposal. As a result, evolutionary selection becomes challenging due to the absence of a unified golden metric to quantify performance. Traditionally in RL, the success determination function for a specific task is programmed by humans into a function that is called at every physics step. Without resorting to human efforts to exhaustively construct such success conditions, we let LLMs generate success functions as well. These are being composed similarly to reward functions, but with a binary output to indicate how the task is completed. Nevertheless, the soundness of the success function requires investigation. Given that the LLM serves dual roles as both a “player” (for reward function optimization) and as a “referee” (for success determination), employing the resultant success rate as the fitness function for evolution may jeopardize learning stability and trustworthiness. See also [Figure 7.7](#) and [Research Question 7.2](#) for further discussion of the undesired behaviors stemming from evolution with incorrect fitness measurement. The success function and reward function form a chicken-egg relation in that 1) the reward function search relies on reliable success determination, and, meanwhile, 2) it is unfeasible to verify the success function before the learning. As a result, it is challenging to have an evolutionary search for both at the same time.

Success Determination: Fast and Slow

Distinguishing whether specific behaviors fulfill a task at each RL step (referred to as *fast success determination*), as opposed to assessing after fixed intervals of execution (as *slow success determination*), is pivotal in RL. Employing fast success determination enables the agent to receive *sparse rewards* (as only success or failure can be notified by the success function) in real-time and terminate actions promptly to prevent potentially adverse explorations. Although the success function generated by LLMs is essential for RL training, it may also be untrustworthy. Therefore, an independent post-training evaluation is required as a complementary measure to ensure reliability. For instance, the human examination of learned RL behaviors, especially from a draft success function to debug, can be regarded as a slow success determination. Intuitively,

VLMs can be applied to analyze robot behaviors. For example, REFLECT [LBS23] uses a multimodal structure to explain execution anomaly. To establish a stable learning cycle, we propose integrating both *fast* and *slow* success evaluations, distinguished by the temporal extent of the underlying processes, to enhance reliability:

- fast*: sample a set of success functions that are used unchanged throughout skill learning, based on which an LLM launches RL training and deterministically selects reward function survivors. The underlying hypothesis is that determining success is much more attainable than programming an applicable reward function.
- slow*: prompt an independent VLM to additionally examine the success of survivor candidates, before passing them on to the next evolutionary generation. In particular, with the context of the environment and task, a VLM is tasked to describe and assess the reinforced robot behaviors that are deemed successful according to success functions (positives), securing a robust learning loop without human supervision. Since there are many more unsuccessful behaviors (negatives), and false negatives are less frequent and less harmful than true negatives³, we do not additionally examine them due to the assessment cost.

Early Misconduct Check

In practice, LLMs may generate unacceptable function designs, *e.g.* trying to import unsupported third-party Python modules or producing nonsensical outputs (see [Appendix D.1.3](#) for an example). Some of the potential bugs can be bypassed by carefully designed prompts, while others should be examined at runtime by a Python interpreter. Instead of directly launching RL and feeding back LLMs all kinds of execution errors at the end, as in Eureka [Ma+24b], we carry out early syntax checks and loop until the function generations meet certain requirements. This separate check reduces unnecessary waiting time for simulation preparations and provides an efficient reward search that focuses only on performance feedback. See [Figure 7.4](#) for an overview of the evolutionary skill-learning procedure. The corresponding prompts can be found in [Appendix D.3](#).

Retrieving Skill Specification with RAG

The process of providing LLMs with environment knowledge necessitates abstraction, which inherently results in information loss. For instance, certain environment parameters, such as the operational constraints of a drawer moving 10 centimeters along the X-axis, are typically embedded within 3D asset properties rather than explicit scene establishment code. Certain aspects of the environment can only be understood through direct interaction. As a result, learning configurations and their associated outcomes from previous trials provide valuable insights for subsequent experiments.

³: False positives appear only when reward functions are better composed than the success functions, being *less frequent* according to the hypothesis above. Besides, failures will not contaminate the existing skill library, being *less harmful* than false positives regarding possible future executions.

Algorithm 3: Agentic Skill Discovery (ASD) for open-skill learning

Input : LLM $\mathcal{M}$, VLM $\mathcal{V}$, Robot $\mathcal{R}$, Environment $\mathcal{E}$, Reinforcement Learning $\mathcal{L}$,
 Task proposal prompt P^{task} , reward function prompt P^{rew} , success
 function prompt P^{succ} ; Max evolutionary search iterations K ;

Output: Discovered skill library Z

```

1 Initialize  $Z \leftarrow \emptyset, \Pi^{\text{pos}} \leftarrow \emptyset, \Pi^{\text{opt}} \leftarrow \emptyset$ ;
2 while  $i \leq N_T$  do
3    $\mathcal{T}_i \leftarrow \mathcal{M}(\mathcal{R}, \mathcal{E}, P^{\text{task}})$ ;
4   // Fix if check fails:  $f_j^{\text{succ}} \leftarrow \mathcal{M}(f_j^{\text{succ}}, \text{Check Log}, \mathcal{R}, \mathcal{E},$ 
       $P^{\text{succ}})$ , same for reward function
5    $F^{\text{succ}} = \{f_j^{\text{succ}}\}_{N_{\text{succ}}} \leftarrow \mathcal{M}(\mathcal{T}_i, \mathcal{R}, \mathcal{E}, P^{\text{succ}})$ ;
6   for  $f_j^{\text{succ}} \in F^{\text{succ}}$  do
7     Initialize  $m_k \leftarrow -1, f_{m_k}^{\text{rew}} \leftarrow \text{null}$ ;
8     while  $k \leq K$  do
9       // The  $\oplus$  operator indicates concatenation
10       $P^{\text{rew}} \leftarrow P^{\text{rew}} \oplus f_{m_k}^{\text{rew}}$ ;
11       $F^{\text{rew}} = \{f_m^{\text{rew}}\}_{N_{\text{rew}}} \leftarrow \mathcal{M}(\mathcal{T}_i, \mathcal{R}, \mathcal{E}, P^{\text{rew}})$ ;
12      for  $f_m^{\text{rew}} \in F^{\text{rew}}$  do
13         $\pi_{i,j,m} \leftarrow \mathcal{L}(\mathcal{T}_i, \mathcal{R}, \mathcal{E}, f_j^{\text{succ}}, f_m^{\text{rew}})$ ;
14         $\text{valid}(\pi_{i,j,m}) \leftarrow \text{Score}(\pi_{i,j,m}, f_j^{\text{succ}}, f_m^{\text{rew}}) > 0$ ;
15         $m_k \leftarrow \arg \max_m \text{Score}(\pi_{i,j,m}, f_j^{\text{succ}}, f_m^{\text{rew}})$ ;
16      if  $\text{valid}(\pi_{i,j,m_k})$  then
17         $\Pi_i^{\text{pos}} \leftarrow \Pi_i^{\text{pos}} \cup \{\pi_{i,j,m_k}\}$ 
18      for  $\pi \in \Pi_i^{\text{pos}}$  do
19        if  $\mathcal{V}(\mathcal{T}_i, \mathcal{R}, \mathcal{E}, \pi)$  then
20           $\Pi_i^{\text{opt}} \leftarrow \Pi_i^{\text{opt}} \cup \{\pi\}$ 
21       $\text{possible}(\mathcal{T}_i) \leftarrow \{\Pi_i^{\text{opt}}\} \neq \emptyset$ ;
22       $Z \leftarrow Z \cup \{\Pi_i^{\text{opt}}\}$ ;
23       $P^{\text{task}} \leftarrow P^{\text{task}} \oplus \mathcal{T}_i \oplus f^{\text{succ}} \oplus f^{\text{rew}} \oplus \text{possible}(\mathcal{T}_i)$ ;
24       $i \leftarrow i + 1$ ;

```

The collection of historical data represents an environment context distillation process, drawing from both LLMs' environment awareness and VLMs' behavioral assessment capabilities, being able to gradually compensate losses introduced through abstraction. Despite lacking a dedicated 3D structure interpretation module, our agent can derive aspects of such environment constraints through iterative experimentation with various learning specifications. This discovered information is subsequently encoded within learning parameters, for example, a selected reward function that incentivizes positive x-coordinate displacement of the drawer handle effectively facilitates successful drawer-opening task completion.

To reduce redundant exploration and effectively integrate prior knowledge, we

implement a knowledge accumulation strategy that enables the LLM to retrieve specifications of previously learned skills relevant to the current learning objective. Retrieval Augmented Generation (RAG) [Gao+24b], which enhances LLM performance by incorporating contextually relevant information from a local data pool into prompts, has been widely adopted. In our skill learning framework, where success and reward functions are identified through evolutionary search, and where many skills within the same environment share common structural patterns, we apply RAG to enhance LLM prompting. Specifically, we retrieve previously verified skill specifications, including their associated success and reward functions, from the evolving skill library and use them to augment the prompt for generating new functions. This process, which we refer to as skill-RAG, provides the LLM with concrete, context-relevant examples that guide its output and reduce ambiguity. By narrowing the evolutionary search space, skill-RAG improves both the efficiency and reliability of open-ended robot skill acquisition. The impact of this augmentation is demonstrated through ablation results in [Table 7.1 on page 114](#).

7.3.3 On-demand Skill Learning with Quest Decomposition

ASD initially learns skills starting from similar environment reset states $s_0 \sim \rho_0$, where ρ_0 indicates the initial state distribution with limited randomness, such as object placement. Consequently, some LLM-suggested skills cannot be trained if the pre-conditions are not satisfied. For example, the skill “placing an object” requires the initial state of having the object picked. An intuitive solution is to configure the learning environment open-ended/reset-free [Gup+21; Wan+24a], where an LLM continually observes the changes and proposes tasks to complete. However, it challenges both the dynamic sensing ability of LLM as well as RL in practice, especially when RL is accelerated by learning in many parallel environments. Another way is to reset the environment to the final state of executed skills, thereby exploring sequentially arranged further skills conditioned on already collected skills. Since the bottom-up bootstrapping of skills leads to an explosion of possibilities, we introduce a top-down on-demand learning strategy for the complex tasks that RL fails to learn, which we term *quests*. Given a quest selected from the failure pool generated during the skill discovery phase, an LLM is tasked with decomposing it into a sequence of subtasks, thereby establishing a Hierarchical Reinforcement Learning (HRL)

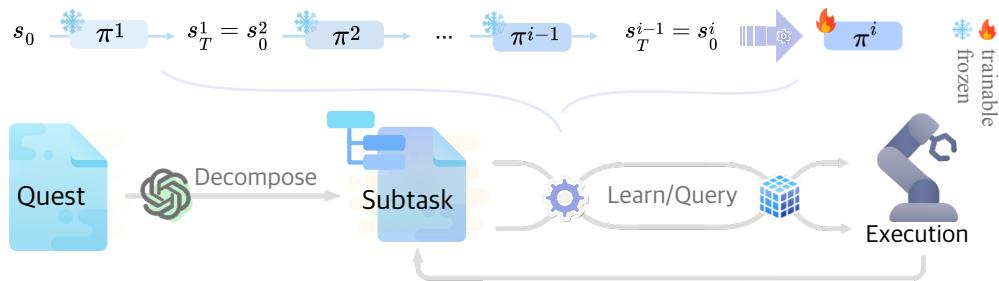


Figure 7.5: *Top:* By chaining together learned skills, ASD can further learn a new skill π^i on demand. This is needed when a complex task is too challenging for RL to learn as a whole. *Bottom:* ASD solves quests, namely challenging tasks, with top-down decomposition and skill learning (purple gear), where the skill library (blue cube) expands for each subtask’s completion.

framework. This decompose-and-conquer strategy has been successfully verified to work [Ahn+22; Zha+23c; Din+23].

However, these approaches only allow a decomposition into a limited set of subtasks that can be completed with known skills (bottom-up for completion). In contrast, our method, as illustrated in Figure 7.5, allows the LLM to come up with novel and contextually appropriate skills to be learned on demand. This capability enhances the framework’s flexibility and generalizability in addressing challenging tasks that require the LLM to construct sophisticated reward functions.

Algorithm 4: Agentic Skill Discovery (ASD) for quest completion

Input : Quest $\mathbb{Q}$, LLM $\mathcal{M}$, Robot $\mathcal{R}$, Environment $\mathcal{E}$ with initial state s_0 , ASD skill learning based on (initialized as) state s : $ASD(\cdot|s)$, collected skill library Z_0

Output: Quest Completion

```

1 Initialize  $s \leftarrow s_0$ ;
  // Task decomposition
2  $\mathcal{T} = \{\mathcal{T}_i\}_{N_T} \leftarrow \mathcal{M}(\mathcal{R}, \mathcal{E}, P^{\text{decompose}}, \mathbb{Q})$ ;
3 for  $\mathcal{T}_i \in \mathcal{T}$  do
4   if  $\mathcal{T}_i \propto \pi_i \in Z_0$  then
5     // Use the existing skill
6      $\pi_i^* \leftarrow \text{Retrieve}(\pi_i, \mathcal{T}_i, Z_0)$ ;
7   else
8     // Learn the on-demand skill
9      $\pi_i^* \leftarrow ASD(\mathcal{T}_i|s)$ ;
10   $s \leftarrow \text{Execute}(\pi_i^*, s)$ ;

```

To avoid the forgetting problem in multitasking policy learning, we learn individual policy networks for each RL launch, which constitute the skill options, and only keep the surviving ones according to their performance in the evolution loop. As for learning of the i -th on-demand skill, since its initial state is reset as the final state of the last stacked skill π^{i-1} , i.e. $s_0^i = s_T^{i-1} \sim \rho_{\pi^{i-1}}$, we initialize the policy weights from the last learned skill rather than randomly, $\pi^i \leftarrow \pi^{i-1}$. This maintains s_0^i at the beginning of learning and ensures a smooth transition from there when being gradually optimized. Algorithm 4 on the current page summarizes the on-demand skill learning process.

7.4 EXPERIMENTS AND RESULTS

7.4.1 Experimental Setup

Environment. We run the simulation in the Isaac Sim simulator⁴ since it supports parallel environment simulation, which dramatically accelerates the trials of RL with

4: <https://docs.omniverse.nvidia.com/isaacsim>, cfr. § 2.3.3 “Isaac Sim” on page 21.

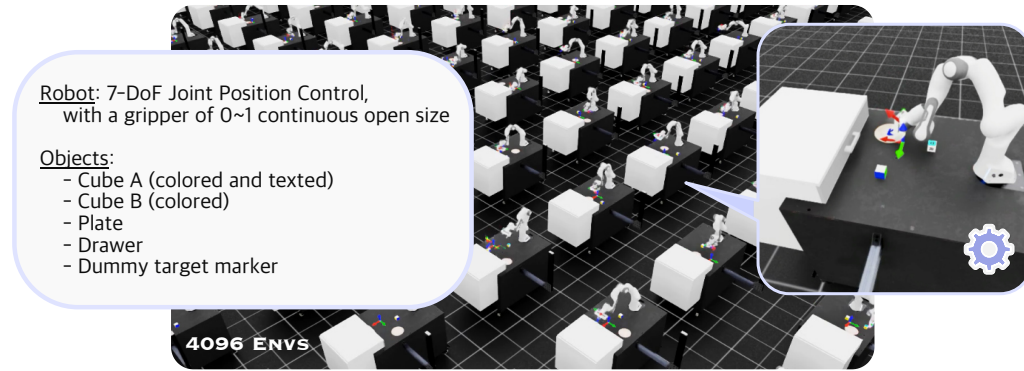


Figure 7.6: The simulated scenario setup in Isaac Sim with parallel environments for RL training.

various reward functions. We set up a table scenario with Franka Emika Panda robotic arm, which has 7 DoF and a two-finger gripper. On the table, several objects are randomly placed in front of the arm, with a drawer that can be opened. It is a single scenario, but it enables multiple tasks. See Figure 7.6 for the environment setup, with all assets⁵ available in Isaac Sim. The experiments aim to answer the following research questions regarding **Objective IV**:

- ▶ **R.Q. 7.1** What kind of tasks will be proposed?
- ▶ **R.Q. 7.2** Can skills be acquired automatically?
- ▶ **R.Q. 7.3** How do RL and learning context influence the learning efficiency?
- ▶ **R.Q. 7.4** Can challenging tasks be completed by chaining learned skills?

LLMs and VLMs Selection. We employ gpt-3.5-turbo to propose tasks and generate reward and fast success functions since it shows a good programming ability while being acceptable regarding cost. For the slow assessment of behaviors generated by policies that have already been positively evaluated using fast success functions, we record the resulting behaviors as videos. Keyframes are then extracted from these videos and analyzed using the advanced VLM GPT-4V(ision)⁶, which verifies the successful completion of the tasks.

Learning Algorithms. With a temperature of 1.0, the LLM samples three success functions in each iteration; based on each, three reward functions are further sampled to launch RL training and to evolve. We set the number of generations of evolutionary search to three. For RL training, we use coordinate states of objects as the state input, and the action space is set to be the robot joint position space. Using joint coordinates instead of Euclidean coordinates avoids control lag caused by inverse kinematics, which would slow down paralleled environments. For optimization, we apply the RL framework `rs_l_rl`⁷ implemented by Orbit [Mit+23], where Proximal Policy Optimization (PPO) [Sch+17] is applied with the same fixed parameters across all potential tasks. For each RL iteration, we configure the maximum permissible physics steps to 250, with 4096 parallel environments, and a total learning duration of 2000 episodes. We train ASD on 6 NVIDIA GeForce GTX 1080 Ti GPUs, where each proposed task takes around 6 hours.

5: Under the license: <https://docs.omniverse.nvidia.com/isaacsim/latest/common/NVIDIA-Omniverse-License-Agreement.html>.

6: <https://openai.com/research/gpt-4v-system-card>

7: https://github.com/leggedrobotics/rs_l_rl

Baseline and Ablation. Given the unpredictable nature of ASD tasks, understanding the learning capabilities of the policy backend is crucial. The ability of code-based LLMs to complete novel tasks has garnered significant research interest [Lia+22; Hua+23; Wan+24a]. In ASD, RL agents are trained to accomplish specified tasks. However, identifying a suitable baseline for evaluating our approach is challenging. To the best of our knowledge, no existing work on semantic skill discovery exists at the time of writing. We omit comparison with Unsupervised Reinforcement Learning (URL)-based skill discovery methods, as they typically produce non-meaningful trajectories without additional human intervention. For task completion without prior skills, we use VoxPoser [Hua+23] as a comparative baseline, limited to evaluating task proposal learning, since, unlike our approach, it is designed for *control*, not *skill discovery*. Moreover, to demonstrate the effectiveness of the skill-RAG technique, we conducted additional experiments where skill learning is halted upon verification, allowing us to quantify the minimum number of GPT calls required. The efficiency gains of skill-RAG are measured through a reduction ratio metric, calculated as the ratio between the minimum number of GPT calls required to master a task with skill-RAG versus the baseline approach without the skill specification augmentation strategy (detailed analysis in [Research Question 7.3](#)).

7.4.2 Analysis

RESEARCH QUESTION 7.1 What kind of tasks will be proposed?

Given the robotic table manipulation scenario, the LLM could potentially propose numerous possible tasks. We stop exploring further task proposals after reaching a number of 24 valid skills. More skill-learning details and statistical reports can be found in [Table D.1](#) in [Appendix D.2 “Skill Learning Reports” on page 161](#). The first column of [Table 7.1](#) presents the proposed tasks in sequential order. From this, we derive the following preliminary observations:

- ▶ The instructional complexity of tasks increases with successive iterations of the proposal, but the associated learning challenges do not necessarily align with this progression.
- ▶ Most of the proposed tasks are meaningful and completable under the setup. Some of the tasks are not appropriately proposed due to LLM’s misconception of the initial environment setup (*e.g.* No.10 for a wrongly deemed initial state, since the drawer is always initialized as closed, it is reasonable to learn to close the drawer) or of the robot’s capabilities (No.19 for difficulties in a guarantee of “without grasping” requirement).
- ▶ By design, all of the tasks are parameter-free language instructions. Hence, the LLM interprets “reach cube A” and “reach cube B” as distinct skills. However, rather than exhaustively enumerating all possible variable permutations (*e.g.* $A \leftrightarrow B$), the LLM employs a strategy that prioritizes semantic diversity in skill proposition.

As can be observed from the task proposal list, the tasks are generally atomic

and meaningful, but there is still potential for improvement, especially in the understanding of the given initial status of the environment. Minimal human effort to examine non-learnable tasks becomes necessary in this case. In this work, we prompt the coding LLM only with text. Future research involving mixed modalities, *e.g.* visual observations or even point clouds [Zha+24b], promises to alleviate this phenomenon.

RESEARCH QUESTION 7.2 Can skills be acquired automatically?

To evaluate whether the proposed tasks have been mastered, we describe the learning status for each proposed task with the following measures:

- ▶ N_O , the number of acquired skill *options*, defined as the count of available policies that satisfy both fast (LLM-based) and slow (VLM-based) success criteria.
- ▶ N_C , the number of acquired skill *candidates*, *i.e.* those policies that are falsely considered positive according to composed fast success functions but failed by the VLM.
- ▶ N_{HO} and N_{HC} , the number of *human-validated options and candidates* as ground truth, respectively, which we define as ground truth.

As shown in the skill option column in Table 7.1, ASD automatically collected many valid skill options. However, the skill candidates column shows that many behaviors were falsely positively evaluated by the coding LLM, necessitating an additional checking mechanism to avoid potential *false learning cycles* (see Figure 7.7). Quantitative analysis demonstrates that fast determination achieves an average precision of $\frac{N_{HO}+N_{HC}}{N_O+N_C} = 44.93\%$, while slow determination yields an average precision of $\frac{N_{HO}}{N_O} = 73.58\%$. This substantial improvement in precision through the application of VLMs for slow success determination reduces false positive outcomes and enhances the stability of the learning cycle and the reliability of acquired skills.

RESEARCH QUESTION 7.3 How do RL and learning context influence learning efficiency?

By alternatively applying VoxPoser [Hua+23] as the skill learning strategy, we show the advantages of RL in learning skills from scratch. As shown in Table 7.1, many tasks can be accomplished using RL but fail when using VoxPoser. This discrepancy can be attributed to VoxPoser’s reliance on extensive human effort to predefine spatial hooks and basic motion primitives, coupled with its lack of the exploration capability inherent to RL. Since the goal of agentic skill discovery is to acquire skills with minimal or no human involvement, we do not meticulously design primitives for VoxPoser but instead, provide only basic movement examples. Consequently, this baseline performs well on tasks requiring simple positional approaches but often fails when more complex manipulations are needed. In contrast, RL-based skill learning, driven by its intrinsic exploration capability, successfully adapts to the environment and masters a broader range of skills.

Table 7.1: Snippet of task proposals based on the table manipulation scenario. The learning results are briefly reported by counting the number of skill options (N_O), the number of skill candidates (N_C), and the number of corresponding validations by human examination (N_{H^*}). The column with RAG shows the minimal GPT calls (for one skill option acquisition) reduction ratio (the smaller, the better) of having skill-RAG over previously without skill-RAG. The skill mastering result of RL and a baseline skill learning strategy VoxPoser (Vox.) are marked as ✓(success, with $\geq 90\%$ success rate) or ✗(failure). The symbol “-” indicates inappropriately proposed tasks within the environment context.

Task Description	N_{HO}/N_O	N_{HC}/N_C	w/RAG	RL	Vox.
1 Reach cube A	4/4	2/2	1.00	✓	✓
2 Reach cube B	8/8	1/1	0.21	✓	✓
3 Reach the plate	7/7	2/2	0.22	✓	✓
4 Pick up the cube A	4/5	0/4	0.98	✓	✗
5 Pick up the cube B	2/2	0/4	0.30	✓	✗
6 Slide cube A from its current position to a target position on the table	3/3	0/6	1.20	✓	✗
7 Open the drawer	1/2	0/10	0.88	✓	✗
8 Pick up the plate	3/3	0/0	0.66	✓	✗
9 Place the plate onto a target position on the table	4/6	0/12	0.88	✓	✗
10 Close the drawer	-/3	-/6	-	-	-
11 Align cube A and cube B to target positions that are apart from each other.	0/0	0/10	0.00	✗	✗
12 Close the drawer with cube A inside.	0/0	0/3	0.00	✗	✗
13 Gripper open/close toggle	1/2	0/4	1.20	✓	✓
14 Slide cube B to the table edge without toppling it, aiming for a target position near the edge.	2/2	0/0	0.83	✓	✗
15 Align end-effector center over the drawer handle without opening or closing the drawer.	2/2	1/2	0.76	✓	✓
16 Navigate the gripper to a target pose above cube B without touching it.	3/4	1/1	0.44	✓	✓
17 Gently push the drawer to a partially open or closed position.	1/1	0/4	0.88	✓	✗
18 Position cube A directly in front of the drawer handle without blocking the drawer from opening.	0/2	0/1	0.00	✗	✗
19 Swap positions of cube A and cube B without grasping.	0/0	0/3	0.00	✗	✗
20 Move end-effector over cube A.	23/23	7/7	0.21	✓	✗
21 Push cube A and cube B close to each other.	1/2	0/31	0.32	✓	✗
22 Move to a target position on the table without interacting with objects.	33/33	1/2	0.18	✓	✓
23 Put cube A into the drawer.	0/1	0/27	0.00	✗	✗
24 Stack cube A on top of cube B.	0/4	0/51	0.00	✗	✗

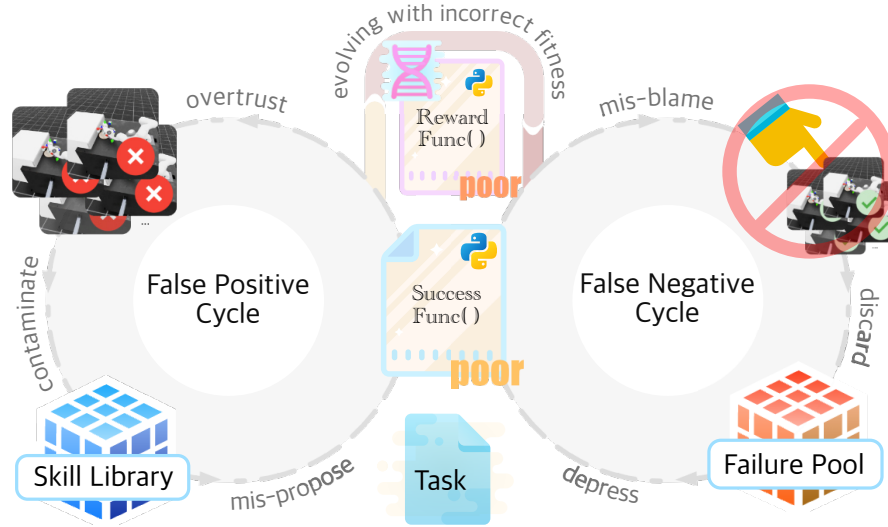


Figure 7.7: Possible failure modes with unassured evolution fitness measurement. *Left:* the LLM-generated success determination may confidently but wrongly assess a task as successful, leading to an undesired *false positive learning cycle* with a contaminated skill library and non-accomplishable future task proposals. *Right:* negative evaluation of indeed successful behaviors will misguide the reward function search and prevent skill acquisition, resulting in a *false negative learning cycle*.

Table 7.1 highlights the advantages of prompting LLMs with previously validated success and reward functions. By retrieving these evolved functions as design cues for the current task, the number of GPT calls required to learn the first skill option is significantly reduced. We argue that, as the complexity of robotic environments increases, accessing skill specifications with RAG will be essential for enhancing the efficiency of agentic skill discovery.

RESEARCH QUESTION 7.4 Can challenging tasks be completed by chaining skills?

Our empirical analysis reveals that certain proposed tasks necessitate extended execution horizons and present significant challenges for LLMs in formulating effective reward functions. These complex tasks, which we term as *quests* in § 7.3.3, typically involve multi-step entity manipulations. For instance, a stacking task requires sequential actions: first grasping one object, and then identifying and aligning it with another object. In our observations, even the highest-performing behaviors predominantly focused on various cube-grasping techniques but failed to progress beyond holding the cube in an elevated position. As illustrated in Figure 7.5, ASD overcomes this limitation by utilizing LLMs to decompose a quest into sequential short-horizon subtasks. Table 7.2 showcases two illustrative cases where ASD initially struggled to master complex tasks as unified objectives but succeeded by systematically decomposing them and addressing the subtasks individually. This demonstrates that the policy π^i can be effectively learned and integrated hierarchically by building on existing skills, underscoring ASD’s potential for solving long-horizon tasks.

Table 7.2: Snippet of quest completion demonstrating the integration of accumulated skills from the library with on-demand learned skills, where the latter are conditionally acquired using the end states of executed skills as initial states, after a *replay* of already acquired skills from Table 7.1.

Quest Decomposition		N_{HO}/N_O	N_{HC}/N_C
Stack cube A on top of cube B			
1)	Pick up cube A	▷ <i>replay</i>	▷ <i>replay</i>
2)	Place cube A on top of cube B carefully, aligning their surfaces to stack them	1/2	0/22
Put cube A on top of the plate			
1)	Pick up cube A	▷ <i>replay</i>	▷ <i>replay</i>
2)	Place cube A on top of the plate	3/3	0/3

7.5 CONCLUSION

Agentic Skill Discovery (ASD) addresses a broad vision for agentic AI systems [Sha+23; Seq24; Zha+24a; Qia+24], enabling robots to understand complex embodiments and autonomously pursue intricate goals with minimal human intervention. By using LLMs to devise, motivate, and improve necessary learning processes, we have shown that language-conditioned robotic skills can be discovered from scratch, where RL and RAG techniques are beneficial for the efficacy and efficiency of skill learning. Using a VLM for third-party behavior assessment prevents the skill library from being influenced by false positives. Furthermore, ASD also promises to tackle challenging, long-horizon tasks by dividing and conquering on demand and thereby further effectively extending the skills.

7.6 LIMITATIONS AND FUTURE WORK

Abstraction and Environment Diversity. As discussed in Research Question 7.1, relying on text-based LLMs to interpret environmental information inevitably introduces a level of context abstraction. This abstraction may obscure fine-grained environmental cues critical for precise behavior learning and evaluation. Moreover, using LLMs to describe and assess robot behavior introduces potential biases and limitations in the evaluation process. A valuable future direction would be to develop and fine-tune specialized robot behavior assessment models [Ma+23], leveraging existing multimodal robotic datasets to improve precision and reliability. Additionally, while the current experimental validation focuses on a robotic arm scenario, which effectively demonstrates the framework’s capabilities, it remains a relatively constrained task domain. Extending the framework to more complex and diverse scenarios, such as deformable object manipulation, multi-agent coordination, or long-horizon tasks, will be important to further evaluate its generality and robustness.

Simulation and Real-World Applicability. In addition, the applicability of this method to

real-world scenarios and diverse embodiments could be evaluated to fully explore its *agentiveness* [Sha+23], especially in cases where parallel learning is necessary. Learning-based methods are typically data-intensive, making direct training on real robots slow, resource-exhaustive, and often impractical. To address this, ASD relies on parallel simulation for scalable skill discovery. One emerging direction involves building digital twin environments to safely and efficiently explore policies before transferring them to physical systems. Another strategy is to first collect real-world data to build a simulation model (real-to-sim), then train in simulation, and finally deploy back to the real robot (sim-to-real [Höf+21; Gäd+22]). These real-to-sim-to-real approaches [Li+24; Tor+24] are increasingly adopted in recent research to improve transferability and reduce the cost of real-world trials. Moreover, even in the absence of a physical robot, the ASD framework remains useful, *e.g.*, in discovering reusable skills for digital agents, such as virtual character animation or game behavior modeling.

Toward Long-Horizon Autonomy. The current implementation focuses on acquiring basic skills; however, ASD lays a foundation for scalable autonomy. Future research should explore mechanisms for chaining acquired skills sequentially or integrating them into a graph-structured representation of policies to enable long-horizon tasks. Robustness is another key factor, *i.e.* learned skills must remain effective across varying contexts. To this end, methods such as domain randomization and task-level precondition verification should be investigated to improve skill generalization and reliable activation.

Extending Embodied Autonomy

To advance embodied autonomy, a series of collaborative research have been conducted that explore complementary approaches to learning, reasoning, and decision-making in robotic systems.

- ▶ **Reward derivation beyond direct environmental signals:** Reward models can be learned from data or inferred through reasoning with generative models, marking a shift from traditional, hand-designed feedback toward autonomously constructed objectives.
- ▶ **Grounding LLM planning:** Task planning using LLMs is enhanced through integration with symbolic structures, enabling more systematic reasoning and long-horizon manipulation. Object-centric environmental representations provides the semantic grounding needed for effective and generalizable planning.
- ▶ **Transparency and interpretability for trustworthy autonomy:** As autonomy increases, so does the need for systems that are transparent and interpretable. These collaborative research contribute to explainable agency through techniques such as reward decomposition, causal abstraction, and post-hoc reasoning with LLMs, helping observers understand and align with the agent's decision-making processes.

Despite the rapid progress in embodied AI and robotic learning, several critical gaps remain underexplored in the literature. (1) While Reinforcement Learning (RL) has shown strong potential in autonomous behavior acquisition, many real-world scenarios lack well-defined or reliable reward functions. This issue is particularly evident in tasks such as those explored in [Chapter 7](#), where reward functions emerge through evolutionary search rather than being explicitly specified. Such cases challenge conventional RL paradigms and call for more systematic investigations into learning with derived or adaptive reward signals. (2) Although LLM have recently been adopted for task planning in robotics, their generated plans often lack guarantees of computational correctness, especially in long-horizon or high-DoF manipulation tasks. The absence of formal validation or planning consistency presents a significant bottleneck in deploying these models for reliable control. (3) The pace of advancement in AI capabilities has far outstripped progress in interpretability and transparency. As intelligent systems become increasingly autonomous and complex, enhancing their explainability is crucial to ensure safety and controllability.

Motivated by these limitations, this chapter organizes collaborative research aimed at broadening the scope of embodied autonomy into the following topics:

- ▶ Reinforcement Learning (RL) with derived rewards ([§ 8.1](#)), *i.e.* with rewards

coming from a separate reward model rather than, in a standard form, from the environment directly, including both inductive reward modeling from joint learning (§ 8.1.1) and deductive reward modeling as reasoning results by generative foundation models (§ 8.1.2). See Figure 8.1 for an illustration of this type of RL.

- LLM-based task planning (§ 8.2), where LLMs are used to generate action plans, especially for bimanual manipulation. This includes both direct controls via LLMs and hybrid approaches combining LLMs with symbolic planning frameworks (*e.g.*, PDDL integration). See Figure 8.4 for illustration.
- Enhancing transparency, interpretability, and reliability of learning models (§ 8.3), with a focus on explainable agency. This includes approaches such as reward decomposition with abstract action spaces, causal state distillation for uncovering decision rationale, and the use of LLMs for post-hoc mental modeling of agent behavior. These methods aim to make intelligent systems more understandable and trustworthy to human users.

8.1 REINFORCEMENT LEARNING WITH DERIVED REWARDS

As is shown in Figure 8.1, RL with derived rewards extends standard RL to apply on fields where the oracle reward functions are not available. The derived rewards, $\hat{r}$, being an estimation of the real rewards, assist an agent to explore and optimize its policies. Usually, the reward models can be inductive, *i.e.* jointly learned with the online collected data by an RL agent, or be deductive, *i.e.* given by an external, pre-trained reasoning model. The different usage of inductive *vs* deductive emphasizes the distinction of the nature of a learning model (usually jointly trained) with collected trajectories *vs* reasoning by a generative model with built-in knowledge. Both ways introduce noise to the rewards, challenging the stability of RL optimization but may be from different aspects.

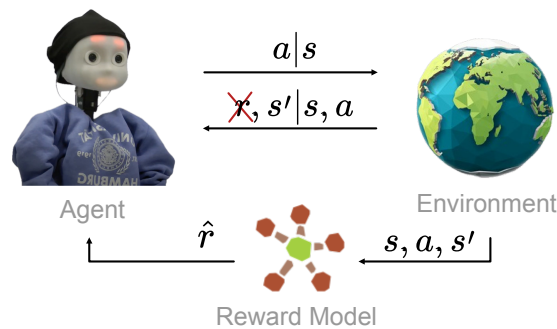


Figure 8.1: Reinforcement Learning (RL) with derived rewards, *i.e.* rewards from separate reward models, *cfr.* Figure 3.5 “Reinforcement learning paradigm” on page 41.

8.1.1 Reinforcement Learning with Inductive Rewards

CONTRIBUTION 8.1 Related resulting publications are

- ▶ [Li+23c]^a (on internally rewarded RL and reward denoising): My contributions in the work mainly reside in the formal formulating this special RL problem (*i.e.* RL with inductive reward models), deriving supporting theory of the formulation, evaluation criteria discussion, and interpretation of experimental results.
- ▶ [Zha+25]^b (on LLM alignment tuning with selective samples): My contribution mainly in bringing active learning to the discussion, explanation for experimental results, and co-coming up with core algorithms accordingly.

^a [Li+23c]: Li* et al. (2023), ‘Internally Rewarded Reinforcement Learning’^b [Zha+25]: Zhang et al. (2025), ‘REAL: Response Embedding-Based Alignment for LLMs’

Inductively rewarded RL, or Internally Rewarded Reinforcement Learning (IRRL) [Li+23c], represents a group of RL algorithms whose rewards come from a jointly learned reward model, in comparison with standard RL whereas an oracle reward function is deemed known. This special form of joint learning process may face a slow cold start challenge¹: initially, the policy is poor such that the collected trajectories are non-informative to train a reward model (*e.g.* training a classifier with noisy observations), and meanwhile, the reward signals back from the poorly learned reward model can be very noisy, further misleading the policy adaptation. See Figure 8.2 for an illustration of this unstable learning with a cold start.



Figure 8.2: The *cold start* issue in Internally Rewarded Reinforcement Learning (IRRL), *i.e.* biased initial learning with inductive rewards, where the randomly initialized policy is poor to get sufficient observations for the training of reward model, and vice versa, the poorly trained reward model produces noisy rewards back to the agent to learn a policy.

IRRL encompasses a broad range of applications, including skill discovery, active vision, and Reinforcement Learning from Human Feedback (RLHF) / Reinforcement Learning from AI Feedback (RLAIF), where rewards are derived from a learned reward model rather than provided directly by the environment.

Skill Discovery. A common formulation involves sampling a latent skill $z \sim p(z)$, generating a trajectory $\tau \sim \pi_\theta$, and maximizing the mutual information $I(z; \tau)$ between the skill and the resulting behavior [Las+21b; Yan+25; BSK21; Rho+25;

1: The phenomenon is less obvious for LLM alignment tuning because the LLM, as a generation policy, has been pre-trained and supervised fine-tuned before Reinforcement Learning from Human Feedback (RLHF).

Kam+22; KPK21; Las+21a; Sha+20; Eys+19]. This objective is typically optimized via the surrogate:

$$\max_{\theta, \psi} \log q_{\psi}(z|\tau),$$

where q_{ψ} denotes a learned reward model that approximates the oracle reward.

Active Vision. It involves a target task with classification label $y \sim p(y)$, where the prior distribution $p(y)$ is typically known. The objective is to maximize the information gain relevant to solving the given task [Baj88; Li24]. Taking a robotic question-answer task as an example, the robot has a limited observation horizon each time step, and it is supposed to collect sufficiently minimal observations over time to answer a, for example, classification problem (see Figure 8.3). The objective to maximize can be

- ▶ accuracy-based reward: $\mathbf{1}(\hat{y} = y)$, where $\hat{y}$ is the predicted classification from the model with parameters ψ ;
- ▶ or similarly to above, estimated posterior $q_{\psi}(y|\tau)$, which can be derived from an approximation of mutual information objective $I(y; \tau)$, providing a more smooth estimation of the quality of currently observed information.

RLHF / RLAIIF. The problems usually start with paired annotations $\mathcal{D} = \{x, y^+, y^-\}_N$, with x being the input prompts, and y^+ and y^- are preferred and dispreferred responses respectively [Chr+17; Sti+20; Cas+23; Ouy+22; Raf+23; Lee+24; Xio+24], etc.. With reward modeling [Ouy+22; Raf+23], $p_{\psi}(y^+ > y^-) = \sigma(\Delta r_{\psi})$, where $\Delta r_{\psi} = r_{\psi}(x, y^+) - r_{\psi}(x, y^-)$. Maximizing the log-likelihood of positive-negative classification leads to a training objective of the reward model (and also a policy implicitly, see Direct Preference Optimization (DPO) [Raf+23] form):

$$\max_{\psi} \log p_{\psi}(y^+ > y^-) = \max_{\psi} \log \sigma(\Delta r_{\psi}).$$

Starting from the RL objective but with $\mathbb{D}_{KL}(\pi_{\theta} || \pi_{\text{ref}})$ constrain removed for simplicity:

$$\max_{\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(y|x)} r_{\psi}(x, y),$$

it can be derived that $r_{\psi}(x, y) \propto \log \pi_{\theta}(y|x)$ [Raf+23], and the final optimization gradient of Direct Preference Optimization (DPO) is

$$\sigma(-\Delta r_{\psi}) \cdot [\nabla \log \pi(y^+|x) - \nabla \log \pi(y^-|x)].$$

This formula for paired data can be further extended as

$$\sum \frac{\pi(y_i^-|x)}{\pi(y^+|x) + \sum \pi(y_i^-|x)} [\nabla \log \pi(y^+|x) - \nabla \log \pi(y_i^+|x)]$$

for one-vs-many contrastive preference.

As is observed, the learning objectives, e.g. the reward format to be maximized, are quite similar. Some of the rewards reveal the learning process, i.e. via an estimation of posterior, which can be regarded as an optimization over a weighted likelihood. A

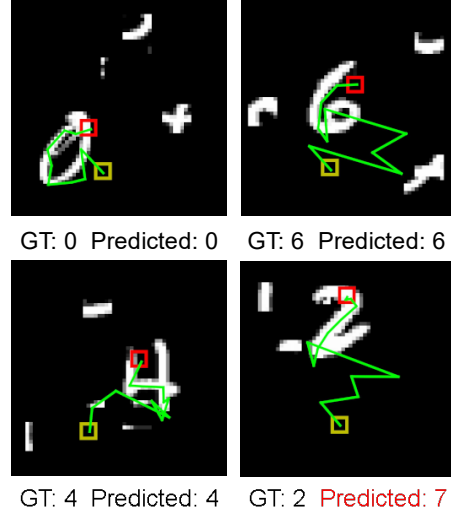


Figure 8.3: Hard attention (a case of active vision) example where the agent can only observe partially and is tasked to collect sufficient information to answer the question: "Which number is currently being observed?". This is a RL problem to optimize the observation policy but with rewards coming from only a jointly trained classification model $q_\psi(y|\tau)$. Image adapted from [Li+23c].

similar idea of distinguishing learning challenges can be also found in [Zha+25], where the learning difficulty of an LLM is controlled by embedding similarity, assuming that hard pairs are too challenging for LLMs and are also less representative to effectively deliver information, while easy pairs, or, even better, centroid pairs (out of clustering centers), are both consumable and informative for LLMs.

One typical challenge resides in IRRL is that, because the reward model is jointly optimized with the evolution of a policy, the derived rewards can be noisy and thus lead to unstable policy learning, possibly resulting in a vicious cycle inside which both the policy and reward model are poorly optimized [Li+23c]². To mitigate this issue, we analyze the noise statistics and propose a clipped linear reward shaping function $f_r(\cdot)$ applied on the original reward signal (from a reward model) $r_\psi(\tau, y) = \log q_\psi(y|\tau) - \log p(y)$, where $p(y)$ is a prior uniform distribution for sampling y :

$$r_\psi^{\text{linear}} = f_r(r_\psi) = \max[q_\psi(y|\tau) - p(y), 0].$$

Despite the simplicity, the clipped linear reward shaping turns to be effective for the IRRL problems mentioned above. Experimental details can be found in [Li+23c]. Note that, the requirement for a consistent optimum is that the function $f_r(\cdot)$ should be an *increasing* function. There are countless functions that meet this minimal requirement, but not all of them are effective. To derive an effective function, a perspective from general f -mutual information [Bel+18; Poo+19; EGI20; Rak+21], as a replacement of the standard mutual information derived from Kullback-Leibler divergence [Li+23c].

[Zha+25]: Zhang et al. (2025), 'REAL: Response Embedding-Based Alignment for LLMs'

2: Evidence can also be found in VIC [GRW17], which resort to an implicit option control to bypass this learning instability.

8.1.2 Reinforcement Learning with Deductive Rewards

CONTRIBUTION 8.2 The related resulting publication is [Chu+24b]^a (on accelerating RL with LLM guidance/feedback): My contributions primarily include co-developing the core methodology for utilizing LLMs to reason about informative rewards that guide RL agents in acquiring low-level control.

^a [Chu+24b]: Chu et al. (2024), ‘Accelerating Reinforcement Learning of Robotic Manipulations via Feedback from Large Language Models’

As is discussed in [Chapter 6 on page 83](#), the reasoning is one of the powerful emergent abilities of LLMs. When grounded into a specific environment and a given task, LLMs are able to generalize the internal knowledge and reason about (1) a suitable solution $\pi_{\text{LLM}}(a|s)$ or (2) the judgment over agent behaviors $r_{\text{LLM}}(s, a)$ [Chu+24b; Wan+24c], with which the agent can either, at least partially, imitate or optimize the learning objective.

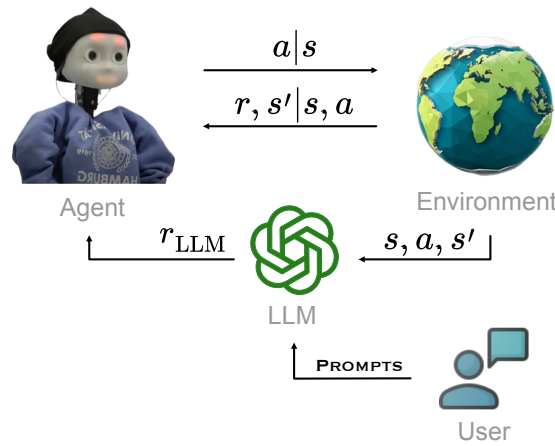


Figure 8.4: Reinforcement Learning (RL) with deductive rewards, where the user only provides once the environment and task context as prompts to the LLM, and the LLM thus *deduces* proper rewards r_{LLM} for the agent. It serves as a complementary reward signal to the environment rewards, leading to an acceleration of the overall learning process.

In previous research studies, the learning guidance for agents is usually provided by domain experts [Chr+17], which is costly and time-consuming. In [Chu+24b], we propose Language agent feedback interactive Reinforcement Learning (Lafite-RL) framework to accelerate RL agent learning with additional *corrective* and *directive* teaching provided by LLMs. The framework starts with a human prompting an LLM with the environment, robot, and task configurations, conditioned on which, the LLM deduces proper additional rewards for the agent to optimize with³. See [Figure 8.4](#) as an illustration. The experiments conducted on several robotic manipulation tasks using RL Bench [Jam+20] (details can be found in [Chu+24b]) demonstrate that the LLM-guided RL, with Vicuna-13B v1.5 model as the language model backend, outperforms the baseline setting with only environment rewards by a large margin in terms of higher success rate and shorter episode length. The corresponding results indicate that large-scale trained foundation models are potentially good *interactive*

³: There also exists works such as [Wan+24c] using VLMs to annotate preference between two trajectories, instead of providing direct rewards, similar to RLAI in LLM training [Lee+24; Li+23a].

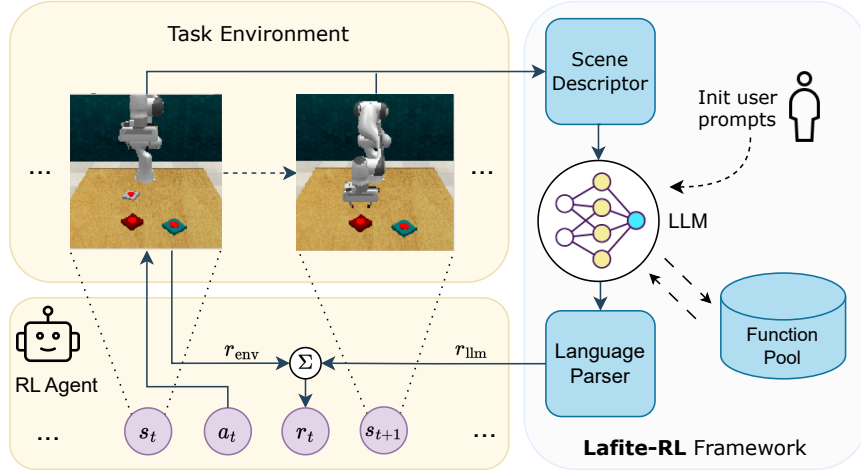


Figure 8.5: Language agent feedback interactive Reinforcement Learning (Lafite-RL) learning paradigm, where the user provides the environment and task context as prompts to the LLM, which then deduces proper rewards r_{LLM} for the agent. The agent learns from both the environment rewards and the LLM rewards, leading to an acceleration of the overall learning process.

teachers, *i.e.* being Process-supervised Reward Model (PRM) rather than just being a Outcome-supervised Reward Model (ORM), as was applied in many previous works, for low-level RL training.

8.2 LLM-BASED EMBODIED PLANNING

CONTRIBUTION 8.3 The related resulting publications are:

- [Sun+24]^a (on object state-sensitive LLM planning): My contributions mainly reside in co-developing the core methodology, designing the experimental robotic scenario, and contributing to the design and analysis of the LLM prompting strategies.
- [Chu+24a]^b (on LLM-based bimanual planning): My contributions include participating in the system design, implementing key modules for integrating LLMs with task and motion planning.
- [Chu+25]^c (on LLM + multi-agent PDDL planning for bimanual manipulation): My contributions are in co-leading the core method design and experimental validation in simulated bimanual manipulation tasks.

^a [Sun+24]: Sun et al. (2024), ‘Details Make a Difference: Object State-Sensitive Neurorobotic Task Planning’

^b [Chu+24a]: Chu et al. (2024), ‘Large Language Models for Orchestrating Bimanual Robots’

^c [Chu+25]: Chu et al. (2025), ‘LLM+MAP: Bimanual Robot Task Planning Using Large Language Models and Planning Domain Definition Language’

Previously in § 3.3.3 “Integration: Planning and Learning with Foundation Models” on page 44 and also in Chapter 5 on page 69, where Matcha agent utilized multimodal cues to guide planning with large language models, general LLM-based task planning frameworks was introduced. This section presents several collaborative efforts that further explore the role of LLMs in planning, with a particular emphasis on object

state sensitivity and bimanual manipulation.

8.2.1 Object State-Sensitive Agent

Ambiguities appear in many cases including both from the user needs and the environment status, and minimizing the ambiguities leads to an automatic and efficient task completion. In [Sun+24], we investigated whether LLMs and Vision Language Models (VLMs) can generate plans that are sensitive to object states (see Figure 8.6 as an illustration). This led to the development of Object-State Sensitive Agent (OSSA), a VLM-based agent that can reason about the states of objects, for example, a *whole intact apple* or a *sliced apple*, in the environment and generate plans accordingly. This is particularly important for daily-life scenarios where the robot needs to understand the context of objects and tasks.

To carry out the experiments and bring research focus, we provide a benchmark dataset⁴ involving 40 scenarios with 184 objects for researches in state identification and planning. To investigate the capabilities of VLMs in this regard, a modular method (which is comprised of separable vision detection module, GRiT [Wu+24], for dense captioning) and VLM-only, *i.e.* monolithic approaches (that incorporate unified VLM, GPT-4V in this case) are compared to handle state-sensitive planning. Experiments on tabletop scenarios demonstrate that the monolithic approach outperforms the modular method in both detection accuracy and manipulation planning. Although the modular method is specifically trained for object detection, it struggles to accurately recognize object states. This limitation can be attributed to the lack of diverse training data covering various object states and its inability to incorporate contextual information from the table for reasoning. In contrast, VLMs exhibit strong reasoning capabilities and can effectively interpret the table context, leading to improved object

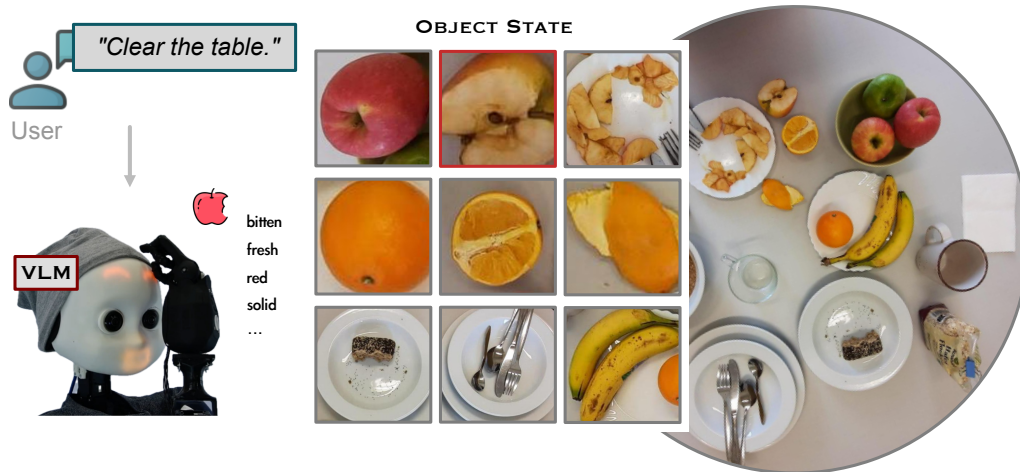


Figure 8.6: The Object-State Sensitive Agent (OSSA) performs context-aware task planning. On a real-world, everyday table, various objects exist in different states—such as fresh fruits, leftovers, and clean plates. The robot must take these object states into account and make conditional decisions accordingly. For example: “Throw away orange peels, put the remaining half apple in the fridge.” Figure adapted and rearranged from [Sun+24].

⁴ See <https://github.com/Xiao-wen-Sun/OSSA> for open-source dataset.

state recognition and more effective planning.

8.2.2 Bimanual Planning

LLM Prompting for Bimanual Manipulation

In [Chu+24a], for complex embodied tasks, we introduced the framework **L**anguage-model-based **B**imanual **O**rchestration (LABOR), a framework that enables direct LLM-driven planning for bimanual coordination in long-horizon tasks. With categorization of bimanual manipulation types and prompting strategies, this work showcased the ability of LLMs to reason about spatial-temporal relationships (see Figure 8.7) between two arms. Experiments on tasks requiring coordinated two-arm execution show that LABOR outperforms the baseline method in task success rate. These results suggest that coordinated planning, particular for long-horizon tasks, demands more sophisticated reasoning than what a baseline LLM can provide. However, this complexity can be effectively addressed using contextual prompts enriched with domain knowledge of bimanual manipulation (*cfr.* Figure 8.8).

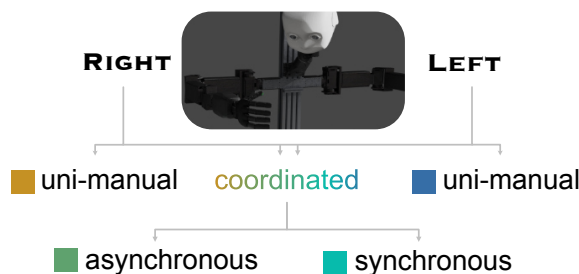


Figure 8.7: Bimanual coordination with spatiotemporal control types. Figure adapted from [Chu+24a].

LLM + Symbolic Planning for Bimanual Manipulation

Complementing LABOR, **LLM + Multi-Agent Planning** with PDDL (LLM+MAP) [Chu+25] tackled the limitations of LLMs in logical consistency by integrating them with symbolic planning through PDDL⁵. LLM+MAP, illustrated in Figure 8.9, leverages GPT-4o’s reasoning abilities to support multi-agent, specifically dual arms

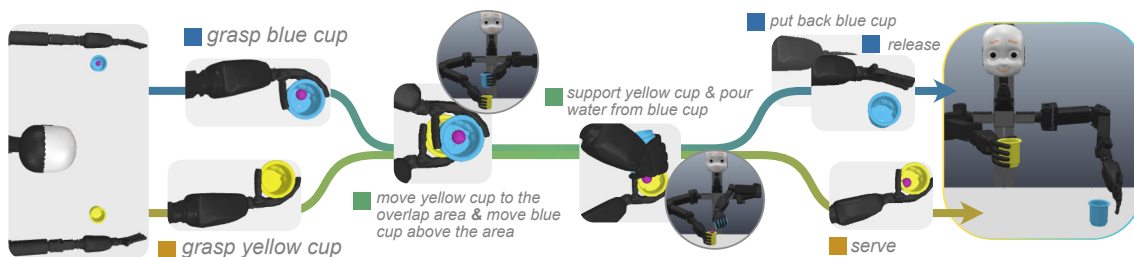


Figure 8.8: An illustration of long-horizon task execution requiring spatial and temporal coordination between two robotic arms of NICOL. The system leverages LLMs for high-level planning and policy generation, allocating subtasks to both arms. This showcases the integration of semantic reasoning with embodied control in dual-arm robotic systems (*cfr.* Figure 8.7 for coordinating types). Figure adapted from [Chu+24a].

5: See also Figure 3.7 “Symbolic planning with PDDL” on page 46 for LLM with PDDL integration.

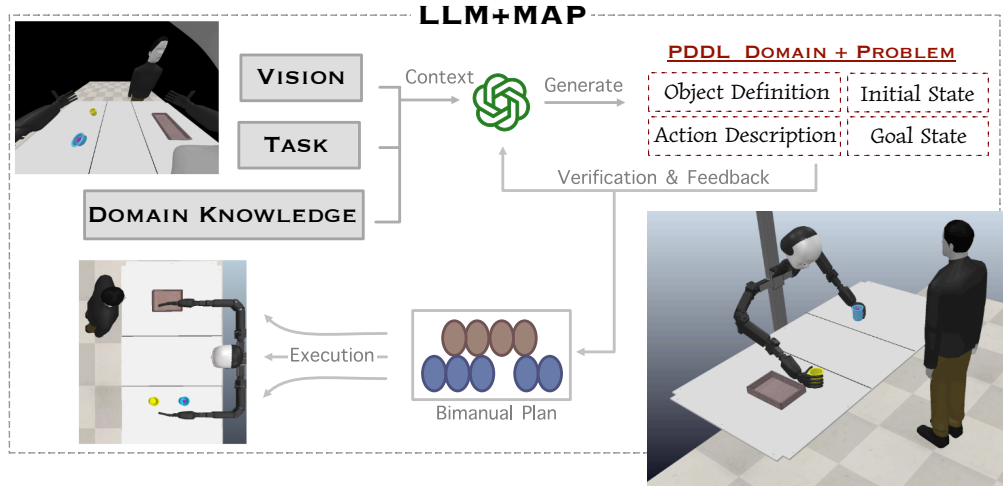


Figure 8.9: An illustration of the **LLM + Multi-Agent Planning with PDDL (LLM+MAP)** architecture. The framework begins with a vision detection module that provides a description of the scene, which, combined with a high-level task description and domain knowledge of bimanual embodiment, is used to prompt LLMs to generate symbolic conditions in PDDL configuration files. These conditions are then consumed by multi-agent solvers for a bimanual plan. After verification, the symbolic plan is translated into low-level control commands for execution. Figure adapted from [Chu+25].

in this case, coordination and efficient task decomposition for bimanual manipulation, significantly improving planning efficiency and robustness.

Experiments compared LLM+MAP with direct LLM-based planning using GPT-4o, OpenAI-o1 [Ope24], and Deepseek-R1 [Dee+25] as baselines on several tabletop manipulation tasks requiring bimanual coordination. The results show that integrating symbolic multi-agent planning with LLMs significantly enhances performance in long-horizon bimanual task planning, yielding higher success rates, faster planning times, and shorter execution paths. Besides, we find that strong reasoning models like Deepseek-R1 and OpenAI-o1 outperform other non-reasoning models, *e.g.* GPT-4o and DeepSeek-V3 [Dee+24], a lot, which indicates that, especially for long-horizon task planning, strong reasoning ability is essential, *cfr.* [Chapter 6 on page 83](#).

Together, these works on bimanual planning highlight the versatility and growing maturity of LLMs in planning tasks that demand contextual awareness, multi-step reasoning, and physical coordination in robotic systems.

8.3 EXPLAINABILITY IN AI AGENTS

CONTRIBUTION 8.4 The related resulting publications are:

- ▶ [Lu+23]^a (on RL decision visualization): My contributions are in co-proposing the core method for RL behavior explanation, and developing LLM-based interface for verbal explanation.
- ▶ [Lu+24]^b (on causality disentanglement and reward decomposition): My contributions are in formulating the disentanglement problem, coming up with sparsity, orthogonality, and sufficiency as optimization objectives, and deriving the principal theory support.

- [Lu+25]^c (on mental modeling RL agent): My contributions are in co-developing the agent mental modeling paradigm and in-depth discussion on evaluation criteria.

^a [Lu+23]: Lu et al. (2023), ‘A Closer Look at Reward Decomposition for High-Level Robotic Explanations’

^b [Lu+24]: Lu et al. (2024), ‘Causal State Distillation for Explainable Reinforcement Learning’

^c [Lu+25]: Lu et al. (2025), ‘Mental Modelling of Reinforcement Learning Agents by Language Models’

As intelligent robotic agents grow in complexity, particularly those driven by RL, explaining their behaviors to human users becomes increasingly vital. This section highlights collaborative efforts toward Explainable AI (XAI) in robotic systems, focusing on both model-intrinsic and post-hoc interpretability approaches, including the integration of LLMs for more intuitive human interaction.

Explainability with Q-Map, Statistics and LLMs

In [Lu+23], we addressed the challenge of providing non-ambiguous, human-understandable explanations of RL agent behavior by reward decomposition within abstract action spaces. The proposed explainable Q-Map framework grounds decision-making in task-relevant object properties and provides visual and textual explanations that align more closely with human reasoning. By extracting statistical features from the Q-values and embedding them into textual templates, we can generate explanations that are more interpretable to human users. Furthermore, with the integration of LLMs, we can enhance the explanations by providing a more interactive and natural language interface, allowing users to query and reason over the agent’s behaviors. See Figure 8.10 for an illustration of the explainable Q-Map framework.



Figure 8.10: Explainable Q-Map framework for RL agents. The framework provides visual and textual explanations of the agent’s decision-making process, enhancing interpretability and user understanding further through interactive query with LLMs. Image adapted from [Lu+23].

Reward Decomposition and Causal State Distillation

In [Lu+24], we further extend reward decomposition with a causal learning framework, which captures the cause-and-effect relationships between states, actions, and rewards (see Figure 8.11.), to uncover interpretable latent structures, *i.e.* causal factors $\{\alpha_i\}_N$. The optimization aims to distill causal factors that are

- sparse, with objective $\max \sum \mathbb{L}(s \rightarrow \alpha_i)$, where $\mathbb{L}(\cdot)$ is the information loss after masking out information the state s to get the i -th causal factor α_i .
- orthogonal, with objective $\min \sum I(\alpha_i; \alpha_j), 1 \leq i, j \leq N, i.e.$ minimizing mutual information between pairs of causal factors.
- and sufficient, with objective $\min || \sum r_\theta(\alpha_i, a) - r ||_2^2$, where r is the total reward supervision and the causal reward models $r_\theta(\cdot)$ should be sufficiently enough to uncover the true rewards, given causal factors $\{\alpha_i\}_N$ and action a .

After the disentanglement, the framework generates more informative and robust local explanations, enhancing transparency in RL-driven systems. See Figure 8.11 as an illustration of reward causal factor disentanglement.

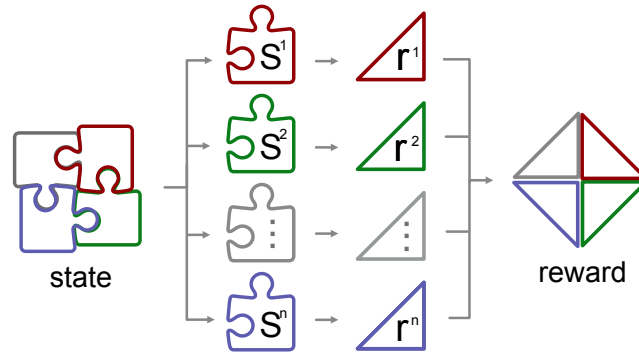


Figure 8.11: State disentanglement and reward decomposition in RL, where the causal factors are extracted from the state-action-reward transition, which are then used to generate explanations of the agent’s decision-making process.

Mental Modeling of RL Agents

In work [Lu+25], we have specifically studied how LLMs can *mental model* the environment transition and also an RL agent decision. The mental modeling process involves prompt LLMs with, similarly, environment information, and additionally the history of agent behaviors. With such context, an LLM is tasked to reason about an agent’s internal behavioral logic and its effect on the surroundings, *i.e.* dynamics transition. See Figure 8.12 as an illustration of mental modeling. Experiments in robotic control scenarios and Atari games demonstrate that modern LLMs are unable

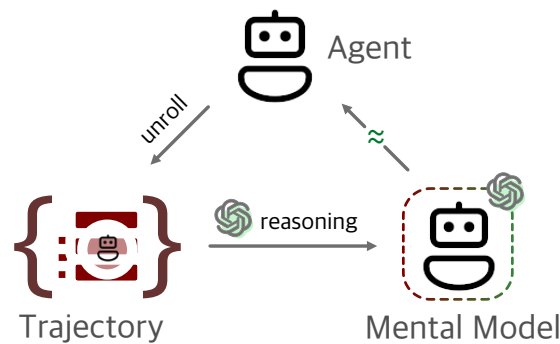


Figure 8.12: Mental modeling of RL agents by LLMs, where the RL agent unrolls trajectories, and via which the LLM *mental model* the agent’s internal decision-making mentalism. After this process, the LLM becomes a “spokesman” for the RL agent for better human-robot interaction and decision transparency. Image adapted from [Lu+25].

to fully construct accurate mental models of RL agents based solely on in-context learning from behavioral histories. This highlights both the current limitations and the promising avenues for future research on the use of LLMs for mental modeling.

8.4 CONCLUSION

This chapter extends the core concepts introduced in earlier chapters by addressing three critical challenges in advancing embodied AI:

- ▶ Lack of well-specified reward functions or guidance in autonomous robot learning: We explore RL with derived rewards, leveraging both joint reward modeling and LLM-based reasoning as guidance.
- ▶ Unreliability of LLM-generated plans for long-horizon robotic control: We mitigate this limitation by domain-specific prompting and grounding LLM planning in symbolic computation frameworks.
- ▶ Widening gap between AI capabilities and interpretability: We address this by developing transparent modules and auxiliary mechanisms to enhance interpretability and trust in decision-making processes.

8.5 LIMITATIONS AND FUTURE WORK

Despite the progress made, several limitations persist. Joint learning of rewards and policies can lead to instability, and LLM-generated rewards, though creative, often lack consistency and depend heavily on task-specific prompting. Integrating LLM planning with symbolic frameworks improves structure but struggles with scalability, as symbolic abstractions can be brittle and overly simplistic. Interpretability efforts, such as causal distillation and post-hoc explanations, offer only partial insight and may trade off performance, limiting their applicability in dynamic real-world settings.

These limitations highlight the need for continued research toward more reliable, generalizable, and interpretable embodied autonomy. Future directions include establishing rigorous evaluation benchmarks, improving the synergy between data-driven and symbolic methods, and enabling introspection and self-correction in autonomous agents.

DISCUSSION, FUTURE DIRECTIONS, AND CONCLUSIONS

9

9.1 DISCUSSION AND FUTURE DIRECTIONS

Building upon the accomplishments detailed in previous chapters, this chapter starts by reviewing recent advancements, remaining challenges in embodied autonomy, highlighting key progress, and outlining future research directions.

9.1.1 Advances on Reasoning, Adaptation, and Multi-Agents

AI develops rapidly and quickly, pushing beyond the scope and timeline of the work presented in this thesis. As discussed in [Chapter 6](#), reasoning remains a central component in achieving robust and generalizable intelligence, and it continues to evolve rapidly with emerging methods. In parallel, agentic systems, such as the autonomous robotic agent explored in [Chapter 7](#), are gaining traction as a promising paradigm for interactive and adaptive intelligence. While this thesis touches on dual-arm control capabilities, the broader trend toward sophisticated multi-agent collaboration underscores the need to further investigate coordination, communication, and scalability. This section provides an overview of recent developments in these directions, highlighting their growing importance for future AI systems.

On Advanced Reasoning

In previous chapters of [Part II](#), several key works benefit from the powerful reasoning ability of advanced AI models. As is also discussed in [Chapter 6](#), ways to improve LLM reasoning ability currently mainly result in inference-time compute scaling, either via fine-tuning (*e.g.* DeepSeek-R1 with RL tuning [Dee+25]) or through prompting [Wei+22a].

Prolonged reasoning pattern intuitively mimics additional mind effort of human thinking when facing complex problems, and scaling inference-time compute generally achieves improved task-solving ability. However, it also leads to increased, sometimes excessive or unacceptable (in terms of waiting time), consumption of computational resources, regardless of problem complexity, which necessitates a long-to-short compression while maintaining overall performance.

Reasoning length compression. Recent efforts aim to reduce reasoning length, especially a preferred short answer for easy problems, to avoid excessive computing and user waiting time. For example, O1-Prunner [Luo+25] applies length-aware reward, along with an accuracy-maintained reward, to fine-tune Marco-o1-7B [Zha+24d] and QwQ-32B [Qwe+25] models to shorten reasoning thoughts while maintaining non-decreased accuracy on math problems.

Inductive vs deductive reward modeling. Since in the process of Reinforcement Learning from Human Feedback (RLHF)¹, the reliability of reward models directly influences the resultant language models. DeepSeek-R1 incentivizes reasoning ability on math and coding problems based on a rule-based reward function, *i.e.* without the dependency on reward modeling. However, for more general problems beyond such domains in which the result can be verified with rules, (explicit or implicit) reward models are still mandatory. As a result, the improvement of the reliability of rewards also attracts research. For example, [Liu+25b] improves reward modeling performance with scaled inference compute, *i.e.* via deductive inference rather than previously commonly applied inductive inference via an end-to-end model with scalar output.

Multimodal reasoning. The challenge of representing information in text, along with the lack of domain-specific or environment-attached data for LLM training, constrains the comprehension of real-world scenarios [Xi+25]. Many works arise to improve the reasoning ability of multimodal foundation modules [She+25; Pan+25], typically with RL to incentivize strong reasoning patterns as is applied in DeepSeek-R1 [Dee+25].

On Autonomous Adaptation

Building an autonomous agent capable of adapting to unforeseen environments and continuously evolving its abilities is an immense challenge [Xi+25].

Traditionally, in skill discovery, the main focus is to maximize options, determined by $s_0, s_T, \pi(s_0 \rightarrow s_T)$, *i.e.* initial and final states and a policy $\pi(\cdot)$ being learnable to achieve the transition. The policy is vital, otherwise it will be meaningless if the agent only knows what can be done but doesn't know how to. Skill discovery belongs to the scope of Unsupervised Reinforcement Learning (URL), usually done by incentivizing an agent with rewards derived from the mutual information between the option z and transition states (s_0, s_T) , being different to the traditional motivation of learning a set of options for the completion of a given extrinsic task. In other words, skill discovery works exhaustively to find available states an agent can reach, regardless of their personal intentions [GRW17].

This can be time-consuming and sometimes excessively redundant for robotic control. In Chapter 7, a semantically motivated skill discovery framework is introduced to efficiently explore affordable skills attached to an environment setting, lying in the scope of bypassing URL, but directly learns semantically meaningful skills. Besides, there are also other recent works trying to limit the skill-searching space. For example, works to constrain a semantic subspace for an RL agent to explore via additional rewarding supervision from foundation models [Rho+25; Yan+25]. Voyager [Wan+24a] expands skills, represented as programming codes, with a self-verification mechanism in a virtual gaming environment, which may, however, non-applicable for real robotic environments where spatial reasoning and reward modeling are required.

To assess agent behaviors, such as determining task success or assigning rewards,

1: *cfr.* § 8.1.1 “Reinforcement Learning with Inductive Rewards” on page 119.

some methods require analyzing sequences of states rather than relying on a single static snapshot [KNK21]. This is essential for capturing dynamic properties, such as whether an object is "slowly moving," without relying on predefined, task-specific features like velocity. Robot sensors can be leveraged to detect transitions in predicate states, providing a richer temporal context for behavior evaluation. As large-scale self-supervised robot skill learning continues to advance, it becomes increasingly important to develop mechanisms, whether rule-based [KNK21] or model-based, *e.g.* [Ma+23], that can reliably monitor sensor streams and skill executions to assess learning progress and guide further adaptation.

On Multi-Agent Systems

Effectively allocating long-horizon tasks among multiple agents involves several critical steps: understanding and decomposing the overall task, assigning subtasks to individual agents based on their specific capabilities and availability, and sequencing executions to ensure efficient collaboration and coordination.

In multi-robot systems, either homogeneous or heterogeneous, coordination and cooperation are essential for successful team performance [Azp+23]. Recent advances have explored leveraging RL and LLMs for heterogeneous multi-agent robot task planning, including task decomposition, coalition formation, and task allocation [KVM24; Cem+25; Hon+24; STT24].

As discussed in § 8.2 “LLM-based Embodied Planning” on page 124, evidence suggests that for long-horizon tasks involving multiple robotic entities, the overall success and effectiveness heavily depend on the reasoning capabilities of LLMs. We incorporate computational search methods, such as symbolic planning, to generate coordinated solutions [Chu+25]. However, these approaches face limitations in flexibility and may incur significant computational costs, especially for complex tasks. Consequently, enhancing the reasoning abilities of LLMs, particularly through exposure to multi-agent domain data, remains a critical direction for enabling more scalable and effective robot collaboration, including bimanual and broader team-based scenarios.

9.1.2 Remaining Challenges

Despite promising progress in integrating LLMs with robotics, a number of persistent challenges, outlined below, along with existing effort and remaining research gaps.

Data Efficiency and Generalization

Learning effective control or reasoning strategies remains highly data-intensive. Self-supervised learning, goal relabeling, and foundation models pretrained on diverse tasks aim to reduce reliance on task-specific data. However, generalization across embodiments and environments remains limited, as most systems lack grounded inductive biases.

[Ma+23]: Ma et al. (2023), ‘Liv: Language-Image Representations and Rewards for Robotic Control’

- Projects such as Open-X-Embodiment [ONe+24] aim to reduce the barrier to data access within the robotics community by promoting a universal protocol for robot data structures and encouraging data availability through a shared platform. While such efforts support large-scale policy learning, particularly through imitation learning, current methods still face limitations. Despite leveraging diverse cross-embodiment datasets, learning detailed and transferable control policies remains inefficient for advanced Vision-Language-Action Model (VLA). Downstream behaviors typically require substantial fine-tuning tailored to specific robot configurations and environments, highlighting that generalization across embodiments and tasks continues to be a major open challenge.
- Sim-to-real transfer is a cheap alternative to collect robot data, but the discrepancies between simulation and the real world hinder policy deployment. Techniques like domain randomization, dynamics adaptation, and modular world models such as Cosmos [NVI+25], which separates physics from perception, support transferability. Nonetheless, cross-domain consistency and embodiment-aligned abstraction learning remain underexplored.

Skill Compositionality

Efficient skill learning where humans train a robot with minimal but sufficient training effort to make the robot compositionally generalize its skills remains a challenge to explore [Vij+25]. Besides, the learned behaviors are often hard to adapt or reuse. Efforts like behavior trees, skill graphs [BSK21], and LLM-guided selection [Ahn+22] aim to modularize action selection. Still, most skills lack clean interfaces or composable structures. Progress may come from integrating symbolic representations with neural modules and using programmatic abstractions to support flexible reuse and transfer.

Reasoning, Planning, and Grounding

The gap between high-level task planning and low-level control still challenges modern autonomous robots. Despite the aforementioned effort to collect massive data, integrating reasoning, planning, and grounding across language, perception, and control from high level to lower is still promising because many of the pre-trained large foundation models are off-the-shelf to use, as we human control robot motions usually by a higher level algorithm design through, for example, crafting reward functions for RL or motion planning objective to optimize, which can be, at least partially, replaced by agentic AI workflows. VLMs, affordance-based methods, and LLM-guided parsing provide partial solutions for semantic grounding, enabling robots to interpret and act on abstract instructions. However, ambiguity handling [Sun+24], spatiotemporal dependency identification [Chu+25], and dynamic environment contexts, *e.g.* open-ended world exploration [Wan+24a; ZWW24], are considered essential to avoid

[ONe+24]: O'Neill et al. (2024), 'Open X-embodiment'

misinterpretations or failures. These challenges are amplified in complex, multi-agent, or long-horizon tasks, where scalable planning and coordination are of the essence. While Hierarchical Reinforcement Learning (HRL), decentralized policies, and symbolic decomposition approaches offer some relief, they remain limited in adaptability and robustness. To enable scalable and generalizable autonomy, future systems must support real-time, scene-aware grounding and flexible reasoning, *e.g.* a generative robot reward model being capable of reasoning about the learning process with robot and environment status, that adapts to uncertainty, interaction, and task complexity.

9.1.3 Future Directions

Building on the findings of this thesis and discussion above, several promising directions are identified for future research to further advance the autonomy and adaptability of embodied agents:

Transferable simulation. Simulation provides an efficient and flexible means to generate large-scale data. With improved fidelity and transfer techniques, addressing the sim-to-real gap can enable robust policy training at scale without extensive real-world interaction.

Learning from cross-embodiment data. To generalize across different robots and environments, future work should develop algorithms that can efficiently leverage heterogeneous datasets. This includes designing architectures and learning strategies that promote embodiment-agnostic representations and transferable skills.

Planning with VLA. While VLA models demonstrate strong generalist potential, their integration with model-based planning and safety mechanisms remains limited. Future research should explore how to align abstract instruction grounding with reliable action planning and execution.

Self-determined learning. Moving beyond imitation, self-determined learning through reinforcement is essential for open-ended autonomy. Future directions include the development of intrinsic motivation, adaptive curricula, and autonomous skill discovery in complex, dynamic environments. A promising path begins with high-fidelity simulation, either using external simulators or internally learned world models, to support efficient exploration and skill acquisition.

9.2 CONCLUSIONS

This thesis has investigated the conceptual foundations and practical advancements in enabling autonomous embodied agents to explore their environment and engage in self-development. Central to this work is the integration of world modeling, semantic grounding, policy learning, and self-determination, which together form the conceptual paradigm guiding this research. Under this paradigm, a comprehensive set of research objectives aimed at advancing the capabilities of autonomous agents has been extensively addressed. Each objective was explored rigorously through

empirical studies and theoretical contributions.

- ▶ Firstly, self-deterministic agents capable of leveraging non-verbal multimodal cues to autonomously explore and develop abilities beyond immediate task requirements were constructed (**Objective I**). The introduction of the Intrinsic Sound Curiosity Module (ISCM) framework demonstrated the effectiveness of crossmodal learning cues in enhancing exploration and representation learning.
- ▶ Secondly, an interactive multimodal perception framework was developed, enabling agents to actively gather, integrate, and semantically interpret diverse sensory inputs for context-aware decision-making (**Objective II**). The Multimodal environment chatting (Matcha) framework exemplifies this approach by incorporating Large Language Models (LLMs) and performing multimodal fusion at the decision level, thereby improving agent performance in complex environments.
- ▶ Thirdly, agent reasoning capabilities were enhanced to interpret complex instructions and make informed decisions (**Objective III**). The Logical Thoughts (LoT) method significantly improved zero-shot Chain-of-Thought (CoT) reasoning, boosting inference-time reasoning and decision-making across multiple domains and model scales.
- ▶ Finally, autonomous agents with advanced self-determination were constructed, capable of verbally sensing environmental context and autonomously discovering meaningful skills (**Objective IV**). The Agentic Skill Discovery (ASD) framework enabled agents to identify and acquire new capabilities efficiently in novel environments through self-determined Reinforcement Learning (RL).

Collectively, these achievements mark significant progress toward enabling autonomous agents to explore novel environments and develop independently, advancing the field of self-developing embodied systems. Recent advancements and future directions further underscore the growing potential of integrating world modeling, semantic grounding, policy learning, and robotic self-determination to build more capable and adaptable agents.

APPENDIX

A

PREDICTION ERROR AND GAUSSIAN MODELING

In this appendix, the mathematical connection between practical implementation of $\|s' - \hat{s}'\|_2$ (where $\hat{s}' = \mu_\theta(s, a), \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$, is the *regression* on next state in latent space) and general quality measurement with log-likelihood function $\log p_\theta(s'|s, a)$ will be discussed, serving as a theoretical complement to § 3.4.1 “Intrinsic Motivation” on page 48 and a foundation for the practice in Chapter 4 “Sound Guides Representations and Explorations” on page 55.

Supposing a Gaussian modeling of $s'|s, a \sim \mathcal{N}(s'|\mu_\theta(s, a), \Sigma)$, the Probability Density Function (PDF) is

$$p_\theta(s'|s, a) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}[s' - \mu_\theta(s, a)]^T \Sigma^{-1} [s' - \mu_\theta(s, a)]}.$$

The intrinsic reward is computed as

$$\begin{aligned} r^{\text{intr}}(s, a, s') &= -\log p_\theta(s'|s, a) \\ &= \frac{n}{2} \log 2\pi + \frac{1}{2} \log |\Sigma| + \frac{1}{2} [s' - \mu_\theta(s, a)]^T \Sigma^{-1} [s' - \mu_\theta(s, a)]. \end{aligned}$$

Simplifying $\Sigma = \sigma^2 I$ with an assumption of the same noise variance across dimensions, we have $|\Sigma| = \sigma^{2n}$ and $\Sigma^{-1} = \sigma^{-2} I$, and

$$r^{\text{intr}}(s, a, s') = \frac{n}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \|s' - \mu_\theta(s, a)\|_2^2.$$

This shows that the log-likelihood is directly linked to the squared prediction error $\|s' - \hat{s}'\|_2$. If further supposing a fixed, not learned, dynamics modeling variance σ , the intrinsic reward can be assigned as $r^{\text{intr}}(s, a, s') = \alpha \|s' - \mu_\theta(s, a)\|_2$, *i.e.* the scaled prediction error in continuous state space.

A similar formulation can be derived using the Mahalanobis distance between the next state s' and the Gaussian prediction $\mathcal{N}(s'|\mu_\theta(s, a), \Sigma)$:

$$(s' - \hat{s}')^T \Sigma^{-1} (s' - \hat{s}'),$$

which serves as a measure of modeling quality, incorporating both prediction error and the model’s estimated uncertainty.

IMPACT SOUND SIMULATION

B.1 PHYSICS-BASED SOUND SIMULATION THEORY

This appendix provides a brief overview of the physics-based sound simulation theory, which is relevant to the sound generation in the context of this thesis (*cfr.* § 2.3.1 “ThreeDWorld” on page 20 for the discussion on simulating impact sound, and § 4 “Sound Guides Representations and Explorations” on page 55 for relevant research work).

Impact sounds can be modeled as vibrations in a physical system. When an object is struck, the impact excites various vibrational modes that produce sound. A mass-spring-damper system provides a simplified but effective model for these vibrations. Considering an object with mass m , which experiences an external impact force $F(t)$, this mass is connected to a spring (representing the object’s elasticity) and a damper (representing energy dissipation during impact). The spring force follows Hooke’s law: $F_s = -kx(t)$, where k is the spring constant and $x(t)$ is the displacement on time step t . The damper exerts a force proportional to the velocity $F_d = -c\dot{x}(t)$ where c is the damping coefficient. This leads to a motion equation:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t),$$

Taking the Laplace transform $\mathcal{L}\{\cdot\}$ [OWN96] of the entire equation leads to

$$m[s^2X(s) - sx(0) - \dot{x}(0)] + c[sX(s) - x(0)] + kX(s) = F(s). \quad (\text{B.1})$$

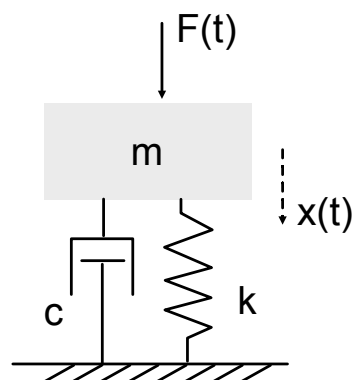


Figure B.1: Mass-spring-damper system illustration with a mass m , spring with k as its spring constant, and damper with c as its damping coefficient. The system is subject to an external force $F(t)$ and experiences displacement $x(t)$.

Equation B.1 can be simplified by supposing static initial state, *i.e.* $x(0) = 0$ and $\dot{x}(0) = 0$, *i.e.* with *free response* neglected:

$$ms^2X(s) + csX(s) + kX(s) = F(s).$$

Therefore, the *forced response* is:

- In Laplace space, as multiplication

$$X(s) = F(s)H(s),$$

where $H(s) = \frac{1}{ms^2 + cs + k} = \frac{1}{m[s^2 + 2\eta\omega_0s + \omega_0^2]}$, $\omega_0 = \sqrt{\frac{k}{m}}$ is the natural (undamped) frequency of the system, $\eta = \frac{c}{2\sqrt{mk}}$ is the damping ratio.

- In time domain, as the convolution of $h(t)$ and $f(t)$

$$\begin{aligned} x_f(t) &= h(t) * f(t) \\ &= \int_{-\infty}^{+\infty} h(t - \tau)f(\tau)d\tau, \end{aligned}$$

In the case of under-damping (most common for real-world impacts), the solution has an oscillatory form. The inverse Laplace transform of $H(s)$ gives the impulse response

$$\begin{aligned} h(t) &= \mathcal{L}^{-1}\{H(s)\} \\ &= \frac{1}{m\omega_d} e^{-\eta\omega_0 t} \sin(\omega_d t), \end{aligned}$$

where $\omega_d = \omega_0\sqrt{1 - \eta^2}$ is the damped natural frequency.

Finally, the forced response in the time domain can be simulated for a specified force $f(t)$:

- For $f(t) = \delta(t)$ (*Dirac delta impulse force*), the system's response is a damped sinusoidal oscillation:

$$x_f(t) = \frac{1}{m\omega_d} e^{-\eta\omega_0 t} \sin(\omega_d t).$$

- For $f(t) = e^{-t^2/\sigma^2}$ (*Gaussian impulse*), which models a smooth, bell-shaped force

$$\begin{aligned} \mathcal{L}\{\ddot{x}(t)\} &= s^2X(s) - sx(0) - \dot{x}(0) \\ \mathcal{L}\{\dot{x}(t)\} &= sX(s) - x(0) \\ \mathcal{L}\{x(t)\} &= X(s) \end{aligned}$$

Note that s denotes the Laplace variable in this context, rather than the agent's state used elsewhere.

applied over time, the response involves a convolution integral:

$$x_f(t) = \frac{1}{m\omega_d} \int_{-\infty}^{+\infty} e^{-\eta\omega_0(t-\tau)} \sin(\omega_d(t-\tau)) e^{-\tau^2/\sigma^2} d\tau,$$

which can be computed numerically for a given σ in practice.

For more complex objects, multiple modes of vibration can exist. This can be modeled by considering multiple mass-spring-damper systems, each corresponding to a different vibration mode¹. Thus, for a realistic sound model, the overall displacement and sound signal can be computed by summing the contributions from different modes:

$$x(t) = \sum_i x_{f,i}(t) \quad (\text{B.2})$$

$$= \sum_i h_i(t) * f(t), \quad (\text{B.3})$$

where the sums run over different modes $h_i(t)$ (if more than one is considered).

B.2 SOUND SIMULATION PRACTICE FOR CERAMIC OBJECTS

In this appendix section, an implementation of impact sound simulation on ceramic objects will be introduced. Unlike bell-like sounds, which have prolonged ringing, ceramic impacts exhibit higher resonant frequencies, increased damping, and a pronounced high-frequency transient. This section presents a modal synthesis method to generate ceramic-like impact sounds. The Python code is provided to reproduce this simple demo.

The first step involves modeling the impact force as a short Gaussian pulse as: $f(t) = e^{-t^2/2\sigma^2}$, if $t_0 \leq t \leq t_0 + \tau$ else $f(t) = 0$, with $\sigma = \tau/8$, where τ is the pulse duration (e.g. $\tau = 0.002$ s) and t_0 is the impact onset time (e.g. $t_0 = 0.01$ s).

```
import numpy as np
import scipy.signal as signal
import sounddevice as sd
import matplotlib.pyplot as plt

fs = 44100
duration = 1.0
t = np.linspace(0, duration, int(fs * duration), endpoint=False)

# Gaussian impact impulse
def generate_impact_impulse(fs, impulse_duration=0.002):
```

¹: Though there exist advanced simulation models [TCM19], further discussion is beyond the scope of this thesis.

```

t_imp = np.linspace(-impulse_duration/2, impulse_duration/2, int(fs *
    impulse_duration), endpoint=False)
sigma = impulse_duration / 8
impulse = np.exp(-t_imp**2 / (2 * sigma**2))
impulse /= np.max(impulse)
return impulse

impulse = generate_impact_impulse(fs, impulse_duration=0.002)
impulse_full = np.zeros_like(t)
start_idx = int(0.01 * fs)
impulse_full[start_idx:start_idx + len(impulse)] = impulse

```

Then, a brief high-pass filtered noise burst $n(t)$ can be simulated to capture the brittle transient. The filtered noise $n_{hp}(t)$ is multiplied by an exponential decay envelope $E_{burst}(t) = \exp(-\lambda t)$, $t \in [0, T_b]$, where $\lambda = 50$ and $T_b = 0.01$ s. The final noise burst is $n_{burst}(t) = n_{hp}(t)E_{burst}(t)$.

```

# Noise
def generate_noise_burst(fs, burst_duration=0.01):
    t_burst = np.linspace(0, burst_duration, int(fs * burst_duration),
        endpoint=False)
    noise = np.random.normal(0, 1, len(t_burst))
    b, a = signal.butter(4, 2000/(fs/2), btype='high')
    noise_filtered = signal.filtfilt(b, a, noise)
    envelope = np.exp(-50 * t_burst)
    noise_filtered *= envelope
    return noise_filtered

noise_burst = generate_noise_burst(fs, burst_duration=0.01)
noise_full = np.zeros_like(t)
noise_full[start_idx:start_idx + len(noise_burst)] = noise_burst

```

On the assumption of N modes with frequencies f_i and damped impulse responses, the system's characteristics of the i -th mode can be expressed as

$$h_i(t) = e^{-\delta_i t} \sin(2\pi f_i t + \phi_i), \quad t \geq 0,$$

where δ_i is the damping factor and ϕ_i is a random phase. The response to the impact is computed as a convolution $x_i(t) = h_i(t) * f(t)$, and, thus, the total modal response is

$$x(t) = \sum_{i=1}^N A_i x_i(t),$$

where A_i are modal amplitudes.

```

# Modes
modes = np.array([1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500,
    6000])
damping = np.linspace(0.2, 0.5, len(modes))
amplitudes = np.ones(len(modes))
phases = np.random.uniform(0, 2*np.pi, len(modes))

```

Finally, the impact sound can be simulated by combining the noise burst with the modal response.

```
# Convolve
def modal_impulse_response(frequency, damping_factor, phase, fs,
    resp_duration=1.0):
    t_resp = np.linspace(0, resp_duration, int(fs * resp_duration),
        endpoint=False)
    h = np.exp(-damping_factor * t_resp) * np.sin(2 * np.pi * frequency *
        t_resp + phase)
    return h
output = np.zeros_like(t)
for f, d_val, amp, phase in zip(modes, damping, amplitudes, phases):
    h_mode = modal_impulse_response(f, d_val, phase, fs, resp_duration=
        duration)
    mode_response = signal.fftconvolve(impulse_full, h_mode, mode='full')
    [:len(t)]
    output += amp * mode_response

mix_ratio = 0.7
output = (1 - mix_ratio) * output + mix_ratio * noise_full
overall_decay = np.exp(-10 * t)
output *= overall_decay
output /= np.max(np.abs(output))

plt.figure(figsize=(10, 4))
plt.plot(t[:1000], output[:1000])
plt.xlabel("Time [s]")
plt.ylabel("Amplitude")
plt.show()

sd.play(output, fs)
sd.wait()
```

After running the program, a short, simulated impact sound for ceramic materials should be played.

This appendix provides supplementary information on the LoT prompting method (*cfr.* Chapter 6 “Enhancing Reasoning via Logic-Guided Inference Scaling” on page 83), including pseudo codes for the reasoning process, and a detailed diagram and prompts for the workflow.

C.1 CHATGPT DEDUCTION EXAMPLE WITH *vs* WITHOUT CONTRAPOSITION

Figure C.1 shows the full ChatGPT conversation of the “football” example introduced earlier in § 6.1 on page 85, indicating the benefits of explicitly prompting LLMs with deduction knowledge such as logic principles.

C.2 PROOF OF CONTRAPOSITION

We will prove contraposition using *reductio ad absurdum*:

Known premises $P \rightarrow Q$ (if P is true, then Q is true) and $\neg Q$ (Q is false), prove $\neg P$.

Proof $P \rightarrow Q, \neg Q \vdash \neg P$:

- | | | |
|----|-------------------|----------------------|
| 1. | $P \rightarrow Q$ | P |
| 2. | $\neg Q$ | P |
| 3. | P | A |
| 4. | Q | $\rightarrow$ E 1, 3 |
| 5. | X | C 2, 4 |
| 6. | $\neg P$ | $\neg$ I 3–5 |

Inside the proof, **P** stands for the known premise, **A** for assumption, $\rightarrow$ E for the elimination of “ $\rightarrow$ ” symbol by following the conditional statement, **C** for contradiction assertion (followed by an “X” which indicates the branch with an assumption is *closed*) and $\neg$ I for the introduction of negation according to the rule of *reductio ad absurdum*.



Anonymous

If Tom plays foot- ball outside, then John will also join to play; if John plays football, then Mary won't go outside. Knowing that Mary is outside, is Tom playing football?



ChatGPT

Based on the given information:

1. If Tom plays football outside, then John will also join to play.
2. If John plays football, then Mary won't go outside.

Given that Mary is outside, we can deduce the following:

- Since Mary won't go outside if John plays football, and Mary is outside, it implies that John is not playing football.
- Since John is not playing football, it doesn't necessarily mean that Tom is playing football, as the first statement only establishes a condition for John joining if Tom plays, not vice versa.

Therefore, we cannot conclusively determine whether Tom is playing football or not based solely on the given information.



Anonymous

use contraposition to deduce



ChatGPT

Let's use contraposition to deduce:

1. If Tom plays football outside, then John will also join to play.
Contrapositive: If John doesn't join to play, then Tom isn't playing football outside.
2. If John plays football, then Mary won't go outside.
Contrapositive: If Mary goes outside, then John isn't playing football.

Given that Mary is outside, we can deduce:

- According to the contrapositive of the second statement, if Mary goes outside, then John isn't playing football.
- Since John isn't playing football, according to the contrapositive of the first statement, Tom isn't playing football either.

Therefore, if Mary is outside, it implies that Tom isn't playing football.

Figure C.1: An example conversation with ChatGPT where the language model fails to correctly deduce the answer initially, but when being prompted to use the idea of “contraposition”, it successfully reaches the desired result.

C.3 PSEUDO CODES FOR CMPS-LOT PROMPTING

Algorithm 2 on page 91 and Algorithm 5 are the pseudo-code of the function to compute the reasoning trace of LoT, where the *difference* regarding discovering contradiction is underlined for clarity.

P is the known premises, *e.g.* question context, and an LLM is employed with various purposes in this context. By prompting the LLM to generate post hoc inferences and subsequently exposing them as discernible options for differentiation, the process facilitates a more convenient verification of entailment, as opposed to relying on the model to independently discover contradictions.

Algorithm 5: Cmps-LoT Reasoning**input** : Problem/Premise P , LLM model**output**: Verified thoughts collection $\mathcal{T}$

```

1 Initialize  $\mathcal{T} \leftarrow \{P\}$  ;
2  $T_1, T_2, \dots, T_N \leftarrow \text{Regex}[\text{LLM}(\mathcal{T})]$ ,  $i \leftarrow 1$  ;
3 while  $i \leq N$  do
4    $E_i^- \leftarrow \text{PostHocLLM}(E | \neg T_i; \mathcal{T})$  ;
5    $C \leftarrow \text{LLM}(E_i^- | \mathcal{T})$  ;
6   if  $C$  is False then
7      $T'_i \leftarrow \text{LLM}(T | \mathcal{T}; T_i; E_i^-)$ ,  $T_i \leftarrow T'_i$  ;
8      $\{T_{>i}\}_{N'} \leftarrow \text{LLM}(\mathcal{T} \cup T_i)$ ,  $N \leftarrow N'$  ;
9    $\mathcal{T} \leftarrow \mathcal{T} \cup T_i$ ,  $i \leftarrow i + 1$  ;

```

Table C.1: Worsening rate ($\downarrow$) and improvement rate ($\uparrow$) when LoT is introduced. Numbers are in %. Note that the accuracy of the number of candidates can significantly impact the outcome.

	Impact	GSM8K	AQuA	Date	SocialQA	Cau.Eff.	Objects	Letter	OddOut
Vicuna-7b	$\uparrow$	0.39	1.51	0.00	3.97	0.00	3.03	0.00	1.56
	$\downarrow$	0.92	10.91	0.00	8.11	0.00	2.94	0.00	0.00
Vicuna-13b	$\uparrow$	3.89	4.88	2.06	3.85	0.00	6.52	2.05	2.67
	$\downarrow$	0.00	8.89	1.74	8.08	0.00	12.90	0.00	8.84
Vicuna-33b	$\uparrow$	0.37	8.02	5.50	20.00	20.83	6.61	0.00	7.84
	$\downarrow$	0.51	10.45	0.00	6.67	0.00	6.25	4.55	5.71
GPT-3.5-turbo	$\uparrow$	12.63	5.71	10.17	1.79	0.00	3.83	0.99	12.50
	$\downarrow$	2.01	0.67	6.59	0.69	0.00	6.59	2.04	2.85
GPT-4	$\uparrow$	6.67	9.68	21.05	0.00	0.00	0.00	12.50	25.00
	$\downarrow$	0.10	0.00	1.79	0.00	0.00	0.00	0.00	0.00

C.4 WORSENING AND IMPROVEMENT RATES

The *worsening rate* computes as $\frac{\#(\text{correct} \rightarrow \text{wrong})}{\#(\text{correct} \rightarrow *)}$, where “#” means count and “*” indicates arbitrary correct/wrong candidates. Similarly, the *improvement rate* computes as $\frac{\#(\text{wrong} \rightarrow \text{correct})}{\#(\text{wrong} \rightarrow *)}$.

From Table C.1, we can have a closer look at the intervention impact of LoT. For example, for small-sized language models such as Vicuna-7b, it is riskier to exert extra intervention, as the model may fail to follow. Indeed, larger models generally benefit from the proposed self-improvement procedure. For instance, GPT-4 exhibited enhanced accuracy on the Date Understanding, LastLetter, and OddOneOut tasks, with the improvement rate significantly surpassing the worsening rate, indicating that the LoT revisions are more trustworthy than the default ones, resulting in better performance.

C.5 LoT PROMPTING DIAGRAM IN DETAIL

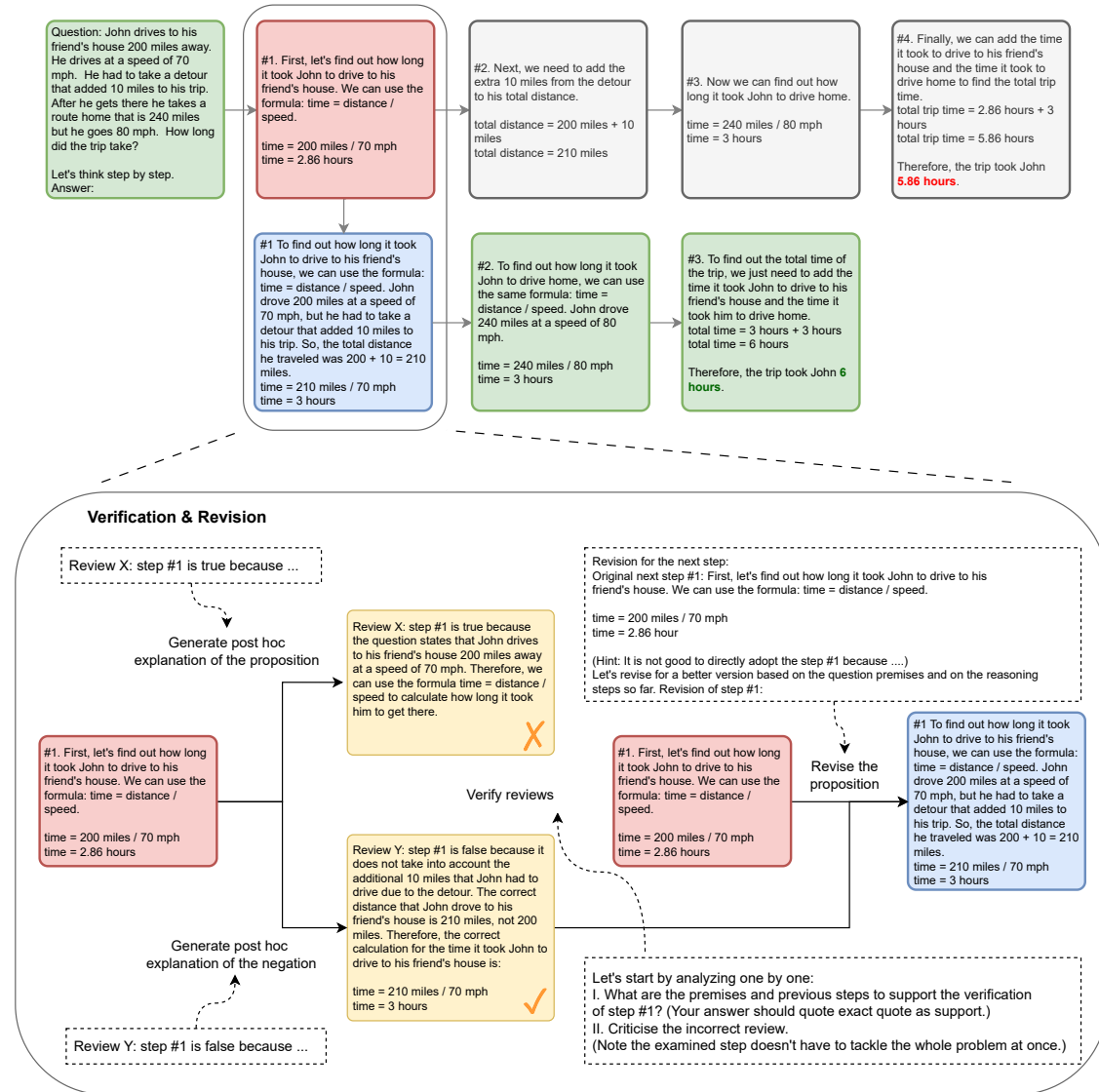


Figure C.2: Illustration of the LoT workflow applied to an arithmetic task from the AQuA dataset. Each reasoning step undergoes verification via two independently generated post hoc reviews by the LLM (●). In this example, step #1 fails verification (●) as the discriminator agrees with “Review Y,” which correctly identifies the error. The LLM then revises the step (●) and regenerates the subsequent steps. This process continues until all steps are verified as valid (●). Dotted boxes highlight key prompt snippets; full prompts appear in [Research Question 6.3](#) on page 95 and [Appendix C.6](#) on this page.

C.6 PROMPTS AND RESULTS FOR CASES

C.6.1 GSM8K Success Case

Initial Zero-shot-CoT steps:

- 1 Question: Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for \$5.00. How much does he spend on yogurt over 30 days?

2

- 3 Let's think step by step.
- 4 Answer:
- 5 #1. First, we need to determine how many packs of 4 yogurts Terry will need to buy over 30 days. To do this, we divide 30 by 4: $30 / 4 = 7.5$
- 6 #2. Since Terry can't buy a fraction of a pack, we round up to the nearest whole number: 8
- 7 #3. Next, we need to determine how many individual yogurts Terry will buy. To do this, we multiply the number of packs by the number of yogurts per pack: $8 * 4 = 32$
- 8 #4. Finally, we need to determine how much Terry will spend on yogurt. To do this, we divide the total number of yogurts by the number of yogurts per dollar: $32 / 4 = 8$
- 9 Therefore, Terry will spend \$8.00 on yogurt over 30 days.
-

Verification of step #1:

-
- 1 Question: Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts for \$5.00. How much does he spend on yogurt over 30 days?
- 2
- 3 Let's think step by step.
- 4 Answer:
- 5 Verification of the next step:
- 6 #1. First, we need to determine how many packs of 4 yogurts Terry will need to buy over 30 days. To do this, we divide 30 by 4: $30 / 4 = 7.5$
- 7
- 8 Let's check two different reviews (X and Y).
- 9 Support the more plausible one and criticise the other one.
- 10 Review X: <review> step #1 is TRUE because Terry eats 2 yogurts a day and there are 4 yogurts in each pack. So, to determine how many packs of 4 yogurts Terry will need to buy over 30 days, we divide 30 by 4. </review>
- 11 Review Y: <review> step #1 is FALSE because the question states that Terry eats 2 yogurts a day, not 4. Therefore, we cannot assume that Terry needs to buy packs of 4 yogurts.
- 12
- 13 Instead, we need to calculate how many individual yogurts Terry needs to buy over 30 days. Since Terry eats 2 yogurts a day, we multiply 2 by 30: $2 * 30 = 60$.
- 14
- 15 Therefore, Terry needs to buy 60 individual yogurts over 30 days.
- 16
- 17 Next, we need to determine the cost of each individual yogurt. The sale price is 4 yogurts for \$5.00, so each yogurt costs $5.00 / 4 = \$1.25$.
- 18
- 19 Finally, we can calculate how much Terry spends on yogurt over 30 days by multiplying the number of yogurts (60) by the cost per yogurt (\$1.25): $60 * \$1.25 = \75 .
- 20
- 21 Therefore, Terry spends \$75 on yogurt over 30 days. </review>
- 22
- 23 Let's start by analyzing one by one:
- 24 I. What are the premises and previous steps to support the verification of step #1? (Your answer should quote exact quote as support.)
- 25 II. Criticise the incorrect review.
- 26 (Note the examined step doesn't have to tackle the whole problem at once.)

27 Finally, identify whether step #1 is true or false.
 28
 29 Analysis and conclusion:
 30 I. The premise to support the verification of step #1 is that Terry eats 2
 yogurts a day and the yogurts are on sale at 4 yogurts for \$5.00.
 31
 32 II. The incorrect review (Review X) is incorrect because it misinterprets the
 information given in the question. The question clearly states that Terry
 eats 2 yogurts a day, not 4. Therefore, the calculation of how many packs
 of 4 yogurts Terry needs to buy over 30 days is not relevant to the problem.
 33
 34 III. Step #1 is false. The correct calculation should be to determine how many
 individual yogurts Terry needs to buy over 30 days, which is 2 yogurts per
 day multiplied by 30 days, resulting in 60 individual yogurts.
 35
 36 Therefore, Terry needs to buy 60 individual yogurts over 30 days.

Revision:

1 Question: Terry eats 2 yogurts a day. They are currently on sale at 4 yogurts
 for \$5.00. How much does he spend on yogurt over 30 days?
 2
 3 Let's think step by step.
 4 Answer:
 5 Original next step #1: First, we need to determine how many packs of 4 yogurts
 Terry will need to buy over 30 days. To do this, we divide 30 by 4: $30 / 4$
 $= 7.5$
 6
 7 (Hint: It is not good to directly adopt the step #1 because there is a review
 says <review> the question states that Terry eats 2 yogurts a day, not 4.
 Therefore, we cannot assume that Terry needs to buy packs of 4 yogurts.
 8
 9 Instead, we need to calculate how many individual yogurts Terry needs to buy
 over 30 days. Since Terry eats 2 yogurts a day, we multiply 2 by 30: $2 * 30$
 $= 60$.
 10
 11 Therefore, Terry needs to buy 60 individual yogurts over 30 days.
 12
 13 Next, we need to determine the cost of each individual yogurt. The sale price
 is 4 yogurts for \$5.00, so each yogurt costs $\$5.00 / 4 = \1.25 .
 14
 15 Finally, we can calculate how much Terry spends on yogurt over 30 days by
 multiplying the number of yogurts (60) by the cost per yogurt (\$1.25): $60 *$
 $\$1.25 = \75 .
 16
 17 Therefore, Terry spends \$75 on yogurt over 30 days. </review>.)
 18 Let's revise for a better version based on the question premises and on the
 reasoning steps so far.
 19 Revision of step #1:
 20 First, we need to determine how many individual yogurts Terry will need to buy
 over 30 days. Since Terry eats 2 yogurts a day, we multiply 2 by 30: $2 * 30$
 $= 60$.
 21

22 Next, we need to determine the cost of each individual yogurt. The sale price
 is 4 yogurts for \$5.00, so each yogurt costs $\$5.00 / 4 = \1.25 .
 23
 24 Finally, we can calculate how much Terry spends on yogurt over 30 days by
 multiplying the number of yogurts (60) by the cost per yogurt (\$1.25): $60 * \$1.25 = \75 .
 25
 26 Therefore, Terry spends \$75 on yogurt over 30 days.
 27 ...

C.6.2 GSM8K Failure Case

Initial Zero-shot-CoT steps:

1 Question: Mike plays ping pong for 40 minutes. In the first 20 minutes, he
 scores 4 points. In the second 20 minutes, he scores 25% more points. How
 many total points did he score?
 2
 3 Let's think step by step.
 4 Answer:
 5 #1. In the first 20 minutes, Mike scores 4 points.
 6 #2. In the second 20 minutes, he scores 25% more points than in the first 20
 minutes.
 7 #3. To find 25% more points, we can multiply the number of points in the first
 20 minutes by 1.25.
 8 #4. $4 \text{ points} * 1.25 = 5 \text{ points}$.
 9 #5. In total, Mike scores $4 \text{ points} + 5 \text{ points} = \boxed{9}$ points. Answer:
 \boxed{9}.
 10 ...

Verification of step #4:

1 Question: Mike plays ping pong for 40 minutes. In the first 20 minutes, he
 scores 4 points. In the second 20 minutes, he scores 25% more points. How
 many total points did he score?
 2
 3 Let's think step by step.
 4 Answer:
 5 #1. In the first 20 minutes, Mike scores 4 points.
 6 #2. In the second 20 minutes, he scores 25% more points than in the first 20
 minutes.
 7 #3. To find 25% more points, we can multiply the number of points in the first
 20 minutes by 1.25.
 8
 9 Verification of the next step:
 10 #4. $4 \text{ points} * 1.25 = 5 \text{ points}$.
 11
 12 Let's check two different reviews (X and Y).
 13 Support the more plausible one and criticise the other one.
 14 Review X: <review> step #4 is TRUE because 25% more than 4 points is $1.25 * 4 =$
 5 points. </review>
 15 Review Y: <review> step #4 is FALSE because the question states that Mike
 scores 25% more points in the second 20 minutes, not 25% more points than

the first 20 minutes. Therefore, we cannot simply multiply the number of points in the first 20 minutes by 1.25 to find the number of points in the second 20 minutes.

16

17 Instead, we need to calculate 25% of the number of points in the first 20 minutes and add that to the number of points in the first 20 minutes.

18

19 25% of 4 points is $0.25 * 4 = 1$ point.

20

21 So, in the second 20 minutes, Mike scores an additional 1 point.

22

23 Therefore, the total number of points scored by Mike is $4 + 1 = 5$ points.

</review>

24

25 Let's start by analyzing one by one:

26 I. What are the premises and previous steps to support the verification of step #4? (Your answer should quote exact quote as support.)

27 II. Criticise the incorrect review.

28 (Note the examined step doesn't have to tackle the whole problem at once.)

29 Finally, identify whether step #4 is true or false.

30

31 Analysis and conclusion:

32 I. The premise to support the verification of step #4 is that in the second 20 minutes, Mike scores 25% more points than in the first 20 minutes.

33

34 II. The incorrect review (Review Y) criticizes step #4 by stating that we cannot simply multiply the number of points in the first 20 minutes by 1.25 to find the number of points in the second 20 minutes. Instead, it suggests calculating 25% of the number of points in the first 20 minutes and adding that to the number of points in the first 20 minutes.

35

36 After analyzing both reviews, it is clear that step #4 is FALSE. The correct approach is to calculate 25% of the number of points in the first 20 minutes and add that to the number of points in the first 20 minutes. Therefore, the total number of points scored by Mike is $4 + 1 = 5$ points.

37 ...

Revision and Reduction:

1 Question: Mike plays ping pong for 40 minutes. In the first 20 minutes, he scores 4 points. In the second 20 minutes, he scores 25% more points. How many total points did he score?

2

3 Let's think step by step.

4 Answer:

5 #1. In the first 20 minutes, Mike scores 4 points.

6 #2. In the second 20 minutes, he scores 25% more points than in the first 20 minutes.

7 #3. To find 25% more points, we can multiply the number of points in the first 20 minutes by 1.25.

8 #4. To find the number of points Mike scores in the second 20 minutes, we need to calculate 25% of the number of points in the first 20 minutes and add that to the number of points in the first 20 minutes.

9

- 10 25% of 4 points is $0.25 * 4 = 1$ point.
 11
 12 So, in the second 20 minutes, Mike scores an additional 1 point.
 13
 14 Therefore, the total number of points scored by Mike is $4 + 1 = 5$ points.
 15 #5. Therefore, Mike scored a total of 5 points.
-

C.6.3 GSM Failure Case

Initial Zero-shot-CoT steps:

-
- 1 Question: Jerry is rolling a six-sided die. How much more likely is it
 (expressed as a percentage) that he rolls a number greater than 3 than that
 he rolls two even numbers in a row?
 2 Let's think step by step.
 3
 4 Answer:
 5 #1. The probability of rolling a number greater than 3 is $3/6$ or $1/2$, since
 there are three numbers (4, 5, and 6) that are greater than 3 out of six
 possible outcomes.
 6 #2. The probability of rolling two even numbers in a row is $(1/2) \times (1/2)$ or
 $1/4$, since the probability of rolling an even number on a single roll is
 $1/2$ and the probability of rolling another even number on the next roll is
 also $1/2$.
 7 #3. To find out how much more likely it is to roll a number greater than 3 than
 to roll two even numbers in a row, we need to compare the difference
 between their probabilities.
 8 #4. The probability of rolling a number greater than 3 is $1/2$, and the
 probability of rolling two even numbers in a row is $1/4$.
 9 #5. The difference between these probabilities is $1/2 - 1/4 = 1/4$.
 10 #6. To express this as a percentage, we can multiply by 100 to get 25%.
 \n\nTherefore, it is 25% more likely that Jerry rolls a number greater than
 3 than that he rolls two even numbers in a row.
-

Revision of step #6:

-
- 1 ...
 2 #6. To express this as a percentage, we need to calculate the ratio of the
 difference between the probabilities to the probability of rolling two even
 numbers in a row, and then multiply by 100 to get the percentage. \n\nThe
 difference between the probabilities is $1/4$, and the probability of rolling
 two even numbers in a row is $1/4$. Therefore, the ratio is $1/4 \div 1/4 =$
 1 . Multiplying by 100 gives us 100%, which means that it is 100% more
 likely that Jerry rolls a number greater than 3 than that he rolls two even
 numbers in a row.\n
 3 #7. Therefore, it is twice as likely that Jerry rolls a number greater than 3
 than that he rolls two even numbers in a row.
-

Revision of step #7:

-
- 1 ...

- 2 #7. Therefore, it is 100% more likely that Jerry rolls a number greater than 3 than that he rolls two even numbers in a row.\n
- 3 #8. Final Answer: It is 100% more likely that Jerry rolls a number greater than 3 than that he rolls two even numbers in a row.

The result is not aligned with the ground truth but is acceptable because it presented the answer with a ratio rather than an absolute value increase.

C.6.4 AQuA Success Case

Initial Zero-shot-CoT steps:

-
- 1 Analyze and answer the following single-choice problem.
- 2 Question: An aeroplane flies along the four sides of a square at the speeds of 200, 400, 600 and 800km/hr. Find the average speed of the plane around the field?
- 3 Options:
- 4 OptA)384 OptB)562 OptC)458
- 5 OptD)156 OptE)452
- 6
- 7 Let's think step by step.
- 8 Answer:
- 9 #1. To find the average speed, we need to find the total distance traveled and divide it by the total time taken.
- 10 #2. The distance traveled on each side of the square is equal to the speed multiplied by the time taken. Let's assume the length of each side of the square is "s" km.
- 11 #3. The time taken to travel each side of the square is equal to the length of the side divided by the speed. So, the time taken for each side is $s/200$, $s/400$, $s/600$, and $s/800$ hours.
- 12 #4. The total distance traveled is equal to the sum of the distances traveled on each side of the square. So, the total distance is $4s$ km.
- 13 #5. The total time taken is equal to the sum of the times taken for each side of the square. So, the total time is $(s/200) + (s/400) + (s/600) + (s/800)$ hours.
- 14 #6. Now, we can calculate the average speed by dividing the total distance by the total time. So, the average speed is $(4s) / ((s/200) + (s/400) + (s/600) + (s/800))$ km/hr.
- 15 #7. Simplifying the expression, we get the average speed as $(4s) / ((3s + 2s + (4/3)s + (1/2)s) / 2400)$ km/hr.
- 16 #8. Further simplifying the expression, we get the average speed as $(4s) / ((19/6)s / 2400)$ km/hr.
- 17 #9. Canceling out the common terms, we get the average speed as $(4/1) / ((19/6) / 2400)$ km/hr.
- 18 #10. Simplifying the expression, we get the average speed as $(4/1) * (2400 / (19/6))$ km/hr.
- 19 #11. Further simplifying the expression, we get the average speed as $(4 * 2400) / (19/6)$ km/hr.
- 20 #12. Calculating the expression, we get the average speed as $9600 / (19/6)$ km/hr.
- 21 #13. Dividing the numerator by the denominator, we get the average speed as $9600 * (6/19)$ km/hr.
- 22 #14. Simplifying the expression, we get the average speed as $3200 * (2/19)$ km/hr.

23 #15. Calculating the expression, we get the average speed as $6400 / 19$ km/hr.
 24 #16. Rounding the answer to the nearest whole number, we get the average speed
 as 337 km/hr.
 25 #17. None of the given options match the calculated average speed.
 26 ...

Revision and Re-deduction:

1 Analyze and answer the following single-choice problem.
 2 Question: An aeroplane flies along the four sides of a square at the speeds of
 200, 400, 600 and 800km/hr. Find the average speed of the plane around the
 field?
 3 Options:
 4 OptA)384 OptB)562 OptC)458
 5 OptD)156 OptE)452
 6
 7 Let's think step by step.
 8 Answer:
 9 #1. To find the average speed, we need to find the total distance traveled and
 divide it by the total time taken.
 10 #2. The distance traveled on each side of the square is equal to the speed
 multiplied by the time taken. Let's assume the length of each side of the
 square is "s" km.
 11 #3. The time taken to travel each side of the square is equal to the length of
 the side divided by the speed. So, the time taken for each side is $s/200$,
 $s/400$, $s/600$, and $s/800$ hours.
 12 #4. The total distance traveled is equal to the sum of the distances traveled
 on each side of the square. So, the total distance is $4s$ km.
 13 #5. The total time taken is equal to the sum of the times taken for each side
 of the square. So, the total time is $(s/200) + (s/400) + (s/600) + (s/800)$
 hours.
 14 #6. Now, we can calculate the average speed by dividing the total distance by
 the total time. So, the average speed is $(4s) / ((s/200) + (s/400) +$
 $(s/600) + (s/800))$ km/hr.
 15 #7. To simplify this expression, we need to find a common denominator for the
 fractions in the denominator. The common denominator is 2400.
 16
 17 $(4s) / ((s/200) + (s/400) + (s/600) + (s/800))$
 18
 19 Now, we can add the fractions:
 20 $(4s) / ((12s + 6s + 4s + 3s) / 2400)$
 21 $(4s) / (25s / 2400)$
 22 $(4s) * (2400 / 25s)$
 23 $(4s) * 96$
 24 $384s$
 25
 26 Therefore, the correct average speed is $384s$ km/hr.
 27 The correct answer is OptA) 384.

C.6.5 AQuA Failure Case

Initial Zero-shot-CoT steps:

1 Analyze and answer the following single-choice problem.
 2 Question: Roy was suffering from severe headaches. He went to see his doctor
 and the doctor gave him 5 tablets asking him to take one tablet every 15
 minutes.
 3 How much time will it take Roy to consume all the 5 tablets?
 4 Options:
 5 OptA)45 Min OptB)75 Min OptC)90 Min
 6 OptD)120 Min OptE)60 Min
 7
 8 Let's think step by step.
 9 Answer:
 10 #1. Roy needs to take 5 tablets.
 11 #2. He needs to take one tablet every 15 minutes.
 12 #3. To calculate the total time, we need to multiply the number of tablets by
 the time it takes to consume one tablet.
 13 #4. 5 tablets x 15 minutes = 75 minutes.
 14 Therefore, the correct answer is OptB) 75 Min.

In this case, all of the reasoning steps passed the verification, such that the answer by LoT is consistent with CoT. However, they are both not correct. The ground truth is OptE) 60 Min, with an explanation that "Tablet 1 will be taken in 0 min...". Because there is no message for this hidden commonsense knowledge, neither CoT nor LoT can realize it.

C.6.6 Date Understanding Success Case

Initial Zero-shot-CoT steps:

1 Question: Jane and John married on Jan 2, 1958. Today is their golden wedding
 anniversary. What is the date one week ago from today in MM/DD/YYYY?
 2
 3 Let's think step by step.
 4 Answer:
 5 #1. First, we need to determine the date of their golden wedding anniversary.
 6
 7 To do this, we need to add 50 years to their wedding date.
 8
 9 Jan 2, 1958 + 50 years = Jan 2, 2008
 10
 11 #2. Next, we need to determine the date one week ago from today.
 12
 13 To do this, we subtract 7 days from today's date.
 14
 15 Today's date - 7 days = (MM/DD/YYYY)
 16
 17 Therefore, the date one week ago from today in MM/DD/YYYY format is
 (MM/DD/YYYY).

Revision and Re-deduction:

1 Question: Jane and John married on Jan 2, 1958. Today is their golden wedding
 anniversary. What is the date one week ago from today in MM/DD/YYYY?

```

2
3 Let's think step by step.
4 Answer:
5 #1. First, we need to determine the date of their golden wedding anniversary.
6
7 To do this, we need to add 50 years to their wedding date.
8
9 Jan 2, 1958 + 50 years = Jan 2, 2008
10 #2. Next, we need to determine the date one week ago from today.
11
12 To do this, we need to subtract 7 days from the date of their golden wedding
    anniversary.
13
14 Jan 2, 2008 - 7 days = Dec 26, 2007
15
16 Therefore, the date one week ago from today in MM/DD/YYYY format is 12/26/2007.
17 The date one week ago from today in MM/DD/YYYY format is 12/26/2007.

```

In this case, CoT successfully applies the commonsense knowledge that a “golden wedding anniversary” means fifty years. However, the language model fails to generate a meaningful answer as a conclusion while the revision by LoT leads to a correct answer.

C.6.7 Date Understanding Failure Case

Initial Zero-shot-CoT steps:

```

1 Question: The deadline is Jun 1, 2021, which is 2 days away from now. What is
    the date tomorrow in MM/DD/YYYY?
2
3 Let's think step by step.
4 Answer:
5 #1. Today's date: May 30, 2021
6 #2. Tomorrow's date: May 31, 2021
7 #3. Convert to MM/DD/YYYY format: 05/31/2021

```

Verification of step #1:

```

1 Question: The deadline is Jun 1, 2021, which is 2 days away from now. What is
    the date tomorrow in MM/DD/YYYY?
2
3 Let's think step by step.
4 Answer:
5
6 Verification of the next step:
7 #1. Today's date: May 30, 2021
8
9 Let's check two different reviews (X and Y).
10 Support the more plausible one and criticise the other one.
11 Review X: <review> step #1 is TRUE because the user stated that the deadline is
    2 days away from now, and today's date is May 30, 2021. </review>
12 Review Y: <review> step #1 is FALSE because the given information states that
    the deadline is June 1, 2021, which means that today's date is May 31, 2021.

```

13
 14 Now, let's move on to the next step.
 15
 16 Step #2: Determine the date tomorrow.
 17
 18 Since today is May 31, 2021, tomorrow will be June 1, 2021.
 19
 20 Therefore, the date tomorrow in MM/DD/YYYY format is 06/01/2021. </review>
 21
 22 Let's start by analyzing one by one:
 23 I. What are the premises and previous steps to support the verification of step
 #1? (Your answer should quote exact quote as support.)
 24 II. Criticise the incorrect review.
 25 (Note the examined step doesn't have to tackle the whole problem at once.)
 26 Finally, identify whether step #1 is true or false.
 27
 28 Analysis and conclusion:
 29 I. The premise to support the verification of step #1 is the user's statement
 that the deadline is 2 days away from now, and today's date is May 30, 2021.
 30
 31 II. Review Y criticizes step #1 by stating that the given information states
 that the deadline is June 1, 2021, which means that today's date is May 31,
 2021.
 32
 33 Based on the user's statement and the given information, it can be concluded
 that step #1 is FALSE. Today's date is May 31, 2021, not May 30, 2021.

Revision and Re-deduction:

1 Question: The deadline is Jun 1, 2021, which is 2 days away from now. What is
 the date tomorrow in MM/DD/YYYY?
 2
 3 Let's think step by step.
 4 Answer:
 5 #1. Determine the date tomorrow.
 6
 7 Since today is May 31, 2021, tomorrow will be June 1, 2021.
 8
 9 Therefore, the date tomorrow in MM/DD/YYYY format is 06/01/2021.
 10
 11 #2. The date tomorrow in MM/DD/YYYY format is 06/01/2021.

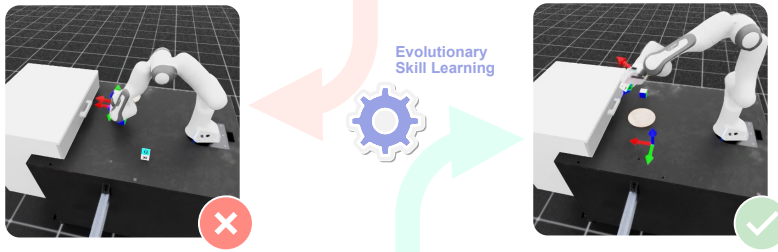
This failure of LoT showcases the importance of correctly identifying the more plausible review, especially when both reviews claim statements but with established facts as support. Apparently, there is no clear reason to support "Review Y", *i.e.* the post hoc explanation, in this case, is just a hallucination of the LLM but it sometimes fools the LLM itself.

This appendix provides additional details on the ASD framework (cfr. [Chapter 7](#) on [page 99](#)), including the implementation details, prompts, and further skill learning reports.

D.1 GENERATED FUNCTIONS EXAMPLES

D.1.1 Success Functions

```
def pick_up_cube_a(env: RLTaskEnv) -> torch.Tensor:
    """Pick up the cube A successfully by determining whether the gripper is closed around the cube.
    """
    gripper_open_distance = env.obs_buf["observations"]["gripper_open_distance"].squeeze()
    is_pickup_successful = torch.where(gripper_open_distance < 0.01, 1.0, 0.0)
    return is_pickup_successful.squeeze()
```



```
def pick_up_cube_a(env: RLTaskEnv) -> torch.Tensor:
    """Sparse reward the agent for picking up cube A."""
    gripper_open_threshold = 0.02
    cube_pick_height = 0.2
    obs = env.obs_buf["observations"]
    gripper_open_distance = obs["gripper_open_distance"].squeeze()
    cube_a_height = obs["cube_a_position"][:, 2]
    drawer_open_distance = obs["drawer_open_distance"].squeeze()

    is_gripper_closed = torch.where(gripper_open_distance < gripper_open_threshold, 1.0, 0.0)
    is_cube_a_picked = torch.where(cube_a_height > cube_pick_height, 1.0, 0.0)

    is_success = is_gripper_closed * is_cube_a_picked * (1.0 - drawer_open_distance)

    return is_success.squeeze()
```

Figure D.1: Two success function examples generated by gpt-3.5-turbo, and their corresponding results by RL. *Top:* an incorrect success determination function can lead to wrongly trusted behaviors. *Bottom:* a correct success determination function results in a desired skill.

Taking the task “pick up the cube A” as an example, we show two typical success functions generated by the LLM in [Figure D.1](#). Given the potential for inaccuracies in generating success functions, the entire learning process runs the risk of being futile, with the added possibility of incorporating poor skills into the skill library. ASD significantly mitigates this risk by employing a coordinated strategy involving fast success determination (LLM sampling) alongside additional slow success determination (VLM verification).

D.1.2 Reward Functions

The evolutionary search (§ 7.3.2) not only revises the reward functions to reduce execution errors (shown in Table 7.1) but also to provide more informative guidance for the RL agents. In Figure D.2, we show two reward functions for the same task but at different iterations (*i.e.* generations of the evolution). The LLM displays, though not always, the ability to improve the reward function by reward shaping, *i.e.* introducing more reward components to smoothly guide the learning agent, which has proven very helpful for RL [NHR99].

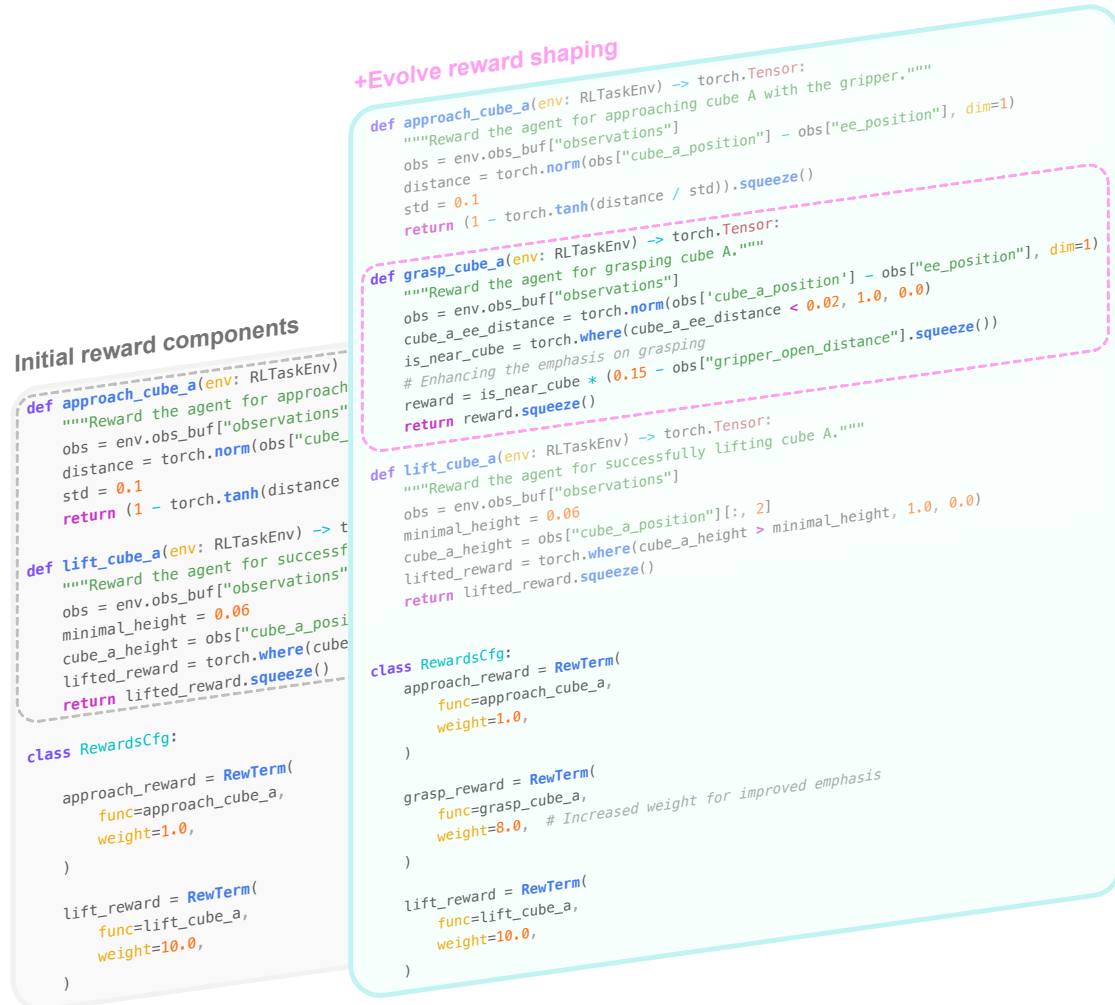


Figure D.2: An example of an improved reward function by evolutionary iteration for the task “pick up the cube A”. *Left:* a reward function focusing on approaching and lifting the cube; *Right:* an evolved reward function to include more detailed guidance of the gripper, leading to efficient RL.

D.1.3 Misconduct

As for many other LLM-based agents, some hallucination risks inevitably arise. Permitting a system to run LLM-generated code may cause unexpected and possibly harmful consequences. Figure D.3 showcases an example of misconducting code generations even with the advanced LLM GPT-4o¹. We recommend early syntax

1: <https://openai.com/index/hello-gpt-4o/>

examination (such as the Python syntax check module in ASD) and implementing system-wide safety guarantees (e.g. a sandbox to run a program). Reports regarding related error measures can be found in [Appendix D.2](#).

```
def place_cube_a_in_drawer(env: RLTaskEnv) -> torch.Tensor:
    """Sparse reward for successfully placing cube A in the drawer."""
    # Hyper-parameters
    minimal_drawer_open_distance = torch.tensor(0.03, device="cuda") # Opened state
    drawer_depth_threshold = torch.tensor(0.15, device="cuda") # Minimal x-offset indicating deep i
    nside the drawer
    drawer_handle_vertical_offset = torch.tensor(0.05, device="cuda") # Assuming visualizing height
    above the handle for efx plane.
    obs = env.obs_buf["observations"]
    gripper_open_distance = obs["gripper_open_distance"].squeeze()
    cube_a_position = obs["cube_a_position"]
    drawer_open_distance = obs["drawer_open_distance"].squeeze()
    drawer_handle_position = obs["drawer_handle_position"]
    # Check 1: The drawer should be open beyond a sufficient threshold
    cond_drawer_open = drawer_open_distance > minimal_drawer_open_distance
    # Check 2: Cube A's position within drawer horizontal border (x values aligned if close enough h
    orizontally)
    cond_cube_in_horizontal_x = torch.abs(cube_a_position[:, 0] - drawer_handle_position[:, 0]) <= d
    rrawer_depth_threshold
    cond_cube_in_horizontal_y = torch.abs(cube_a_position[:, 1] - drawer_handle_position[:, 1]) <= m
    inimal_drawer_open_distance*10
    # Check 3: Cube Z is hanging above or depth checking adequately gauged consistently toward cabin
    et.
    cond_cube_in_depth_z_check =
    token_checker_cube
        (_releleasing_balenced_planes_ anticipating voltrepances).
    comb_cube_gripper_opening=is_whend(XYplane).
    #plans_marketinf as sufficient overlap_transmittinged okay steps:
    (hrz_barcode)*lighting()
    # Final check status regarding planner involved only conditions (all_sanity not erg_elongated).
    is_success = cond_drawer_open Hz MarkHoriv(C).Priori
        * &.load_dist_CC forgedly_matching parejas cond
    .Result_ closing=query cond_cube_Inspect_plane_analz.
    #evaluating outcome ensuring eligible
        dummy.poly_colleciton("heels negotiation ")
    (combined=True_tokpass returns)
        constructing(isnudist Both_trans/scissors chart padded-paper_adj|=
    _raw_outoutTarget buffering.send/(intern P2.program)))
    return post_surfer_dlag_init.non_keyributes exempted(all-answering trx Nx dining)
    is is_fail results_rr return the_native_time_n :=det.trans.if pension_drive)
    equals € ret status Notify else {"MaterialsV"}.
    @configclass Sendavite Dx andIngram_Settings as tab.Basic inertel_as emerg_disstore QueryFunction {}
    intended display ease_scaffold):
    class_=>
    param Heory tactical programs success_fin span.modeSec)
    {
        New.cong.short as indicated.positionable fore-method_descriptor comparing oversee) successmegen
        DX sale**
    pr_ver KX net->> beginners place/Return feasible-Key phased {subset.asp MPversions_ }*episodes_ALG c
    arry_contsumpt-percent predictable<=CV successLoadSpace={ combined()}} for standard safety: fill.Beh
    exitne
    }_QuothR_methods=is.Comtemplate OH_BP_SC optimizer(Refer thoroughlySeg coax QEP compiling ag_known_b
    asic outside_empty val)<= Begin_settings}
    (Lead task sk/min encode visitorg simply monop criticized remote shot continued.)/.
    ...
    .Exchange normal sequencer informing .prior overtime acute*) [... screen auto-adjusting predict]
```

Figure D.3: An example of misconducting code generation with the gpt-4o-2024-05-13 model on the task “put cube A into the drawer”. Top to bottom: the code generation devolves into a chaotic output of potentially harmful and incoherent text (which continues for approximately 500 more lines but is omitted here for brevity).

D.2 SKILL LEARNING REPORTS

In this appendix, we report details of skill learning in [Table D.1](#). To analyze the efficacy of the success functions, we report the following measures:

- ▶ Success Positive (S.P.): a less strict measure than success rate. It measures to what ratio the RL agent can ever succeed (*i.e.* acquire non-zero success at some steps, which is basically a binary measure of whether a certain task *can be achieved*) according to the composed success functions. This measure reveals the difficulties of the task according to the LLM’s own standard.
- ▶ Success Rate (S.R.): the success rate computed by the composed success functions, measures how effective the learning is according to fast success determination. Differing from S.P., S.R. measures also the *efficacy of completions*.
- ▶ Syntax Error (S.E.): a measure of the ratio of misconduct in terms of coding syntax bugs.
- ▶ Execution Error (E.E.): similar to S.E. but counts only errors found after executing the generated codes (codes already passed and revised after syntax check procedure). Typical errors can be Pytorch tensor inconsistencies or running into “nan” gradients after some iterations of optimization.
- ▶ Success Positive for Survivor (S.P.*) and Success Rate for Survivor (S.R.*): the same calculation as for S.P. and S.R. but with a different basis, *i.e.* they compute only for the best selected (surviving) ones of each generation according to the fitness function. By observing in detail only the best-performing ones, these two measures show whether there is overtrust stemming from the success functions for certain tasks. For example, task 1 “reach the cube A” in [Table D.1](#) has very high S.P.* and S.R.*, indicating the task is confidently completed according to the success function, and by observing the successfully collected skill options, we can confirm that the success functions for this task are efficient and trustworthy. However, in task 12 “Close the drawer with cube A inside”, the S.P.* and S.R.* reach high scores, but they turn out to be all false positives, examined by both GPT-4V and human effort. In the latter case, the fast determination of success is overtrusted.
- ▶ Success Positive for Survivor by GPT-4V (S.P.v): measures the ratio of success from the GPT-4V’s perspective among those survivors.
- ▶ Agreement (A.): measures the agreement between fast and slow success determination among survivors. From the learning report across skills, we observe that for easier tasks, regarding both manipulation and visual recognition difficulties (*e.g.* reaching and picking), the success function is more trustworthy and the agreement remains at a relatively higher value.
- ▶ Other statistics (averaged over reward iterations) consistently used as in [Table 7.1](#): number of options (N_O) (according to ASD), candidates (N_C), the number of human-examined total (N_H) and separate validations (N_{HO} and N_{HC} respectively).

The automatically acquired skills are highlighted in [Table D.1](#), which would further expand with more proposals. In addition to learning about proposed tasks, we conducted an ablation study on task descriptions to examine whether the granularity

of task descriptions influences skill learning. For two long-horizon tasks originally labeled as tasks 23 and 24, we manually “translated” them into more detailed instructions while keeping the overall task goals unchanged, resulting in modified tasks 23* and 24* (highlighted with a blue background). Providing more detailed instructions alleviates the burden of reasoning about task procedures when composing reward functions. However, these two tasks remain challenging to complete solely through evolutionary searching of reward functions. This underscores the necessity of top-down decomposition for effective skill learning (see § 7.3.3).

Table D.1: Agentic Skill Discovery (ASD) learning reports, where the successfully acquired skills are highlighted with a light green background; inappropriately proposed tasks (according to the environment potential) are highlighted with a light red color. We run the learning loop 3 times and report the results. The blue highlighted tasks are revised variations of some of the selected complex tasks with detailed subtasks as the instruction, which serve as a reward-shaping hint to LLMs for easier reward design. However, the RL agent still cannot complete them, necessitating a hierarchical structure of skill learning (*cfr.* Research Question 7.4).

No.	Task Description	S.P.	S.R.	S.E.	E.E.	S.P.*	S.R.*	S.P.v	A.	N_O	N_C	N_H	N_{H_O}	N_{H_C}
1	Reach cube A	.67	.90	.00	.33	1.0	.90	1.0	1.0	2	0	2		
		.67	.90	.00	.33	1.0	.91	.50	.50	1	1	2	4/4	2/2
		.44	.89	.00	.56	1.0	.90	.50	.50	1	1	2		
2	Reach cube B	1.0	.79	.00	.00	1.0	.94	1.0	1.0	3	0	3		
		1.0	.77	.00	.00	1.0	.94	1.0	1.0	3	0	3	8/8	1/1
		.89	.79	.11	.11	1.0	.94	.67	.67	2	1	3		
3	Reach the plate	1.0	.62	.00	.00	1.0	.63	1.0	1.0	3	0	3		
		.78	.53	.00	.11	1.0	.62	1.0	1.0	3	0	3	7/7	2/2
		1.0	.42	.11	.00	1.0	.62	.33	.33	1	2	3		
4	Pick up the cube A	.78	.42	.00	.00	1.0	.33	.67	.67	2	1	2		
		1.0	.36	.00	.00	1.0	.33	.67	.67	2	1	2	4/5	0/4
		1.0	.32	.00	.00	1.0	.32	.33	.33	1	2	1		
5	Pick up the cube B	.67	.22	.00	.00	.67	.50	.33	.67	1	1	1		
		.50	.29	.00	.38	.67	.51	.00	.33	0	2	0	2/2	0/4
		.71	.24	.00	.00	.67	.51	.33	.67	1	1	1		
6	Slide cube A from its current position to a target position on the table	1.0	.27	.00	.00	1.0	.04	.33	.33	1	2	0		
		1.0	.22	.00	.00	1.0	.04	.00	.00	0	3	0	3/3	0/6
		1.0	.37	.00	.00	1.0	.51	.67	.67	2	1	2		
7	Open the drawer	.11	.28	.00	.63	.63	.28	.13	.50	1	4	0		
		.35	.07	.00	.47	1.0	.09	.00	.00	0	4	0	1/2	0/10
		.63	.28	.00	.25	1.0	.44	.33	.33	1	2	1		
8	Pick up the plate	.17	.54	.00	.83	1.0	.78	1.0	1.0	1	0	1		
		.17	.41	.00	.83	1.0	.79	1.0	1.0	1	0	1	3/3	0/0
		.17	.51	.00	.83	1.0	.73	1.0	1.0	1	0	1		
9	Place the plate onto a target position on the table	.94	.14	.00	.06	1.0	.05	.50	.50	3	3	1		
		.94	.18	.00	.06	1.0	.14	.17	.17	1	5	1	4/6	0/12
		.82	.18	.00	.18	1.0	.14	.33	.33	2	4	2		
10	Close the drawer	1.0	.99	.00	.00	1.0	.99	.33	.33	1	2	-		
		1.0	.99	.00	.00	1.0	.99	.67	.67	2	1	-	-	-
													/3	/6
		1.0	.99	.00	.00	1.0	.99	.00	.00	0	3	-		
		.48	.10	.00	.26	.83	.03	.00	.17	0	5	0		

11	Align cube A and cube B to target positions that are apart from each other	.61	.25	.00	.10	.50	.07	.00	.50	0	2	0	0/0	0/10
		.72	.28	.00	.00	.75	.05	.00	.25	0	3	0		
12	Close the drawer with cube A inside	.04	.99	.00	.96	1.0	.99	.00	.00	0	1	0		
		.17	.99	.00	.83	1.0	.99	.00	.00	0	1	0	0/0	0/3
13	Gripper open/close toggle	.17	.99	.00	.83	1.0	.99	.00	.00	0	1	0		
		.16	.96	.00	.72	.67	.97	.00	.33	0	2	0		
		.11	.95	.00	.78	.50	.98	.00	.50	0	1	0	1/2	0/4
14	Slide cube B to the table edge without toppling it, aiming for a target position near the edge	.22	.96	.00	.72	.99	.96	.10	.66	2	1	1		
		.22	.41	.00	.22	.33	.76	.67	.67	1	2	1		
		.33	.43	.00	.00	.33	.71	.67	.67	1	2	1	2/2	0/0
15	Align end-effector center over the drawer handle without opening or closing the drawer	.33	.46	.00	.00	.55	.77	.50	1.0	0	0	0		
		.22	.01	.00	.00	.67	.01	.33	.67	1	2	2		
		.00	.01	.00	.00	.00	.01	.67	.33	0	0	0	2/2	1/2
16	Navigate the gripper to a target pose above cube B without touching it	.33	.01	.00	.00	.33	.01	.67	.67	1	0	1		
		.44	.39	.00	.44	1.0	.49	.50	.50	1	1	1		
		.50	.53	.00	.00	.50	.98	.50	1.0	1	0	1	3/4	1/1
17	Gently push the drawer to a partially open or closed position indicated by a target value	.67	.39	.00	.00	1.0	.49	1.0	1.0	2	0	2		
		.11	.57	.00	.56	.50	.57	.00	.50	0	1	0		
		.67	.24	.00	.00	1.0	.30	.00	.00	0	2	0	1/1	0/4
18	Position cube A directly in front of the drawer handle without blocking the drawer from opening	.67	.24	.00	.00	1.0	.30	.50	.50	1	1	1		
		.11	.01	.00	.00	.33	.00	.00	.67	0	1	0		
		.13	.01	.00	.00	.33	.01	.67	.67	1	0	0	0/2	0/1
19	Swap positions of cube A and cube B without grasping	.11	.01	.00	.00	.33	.01	.33	1.0	1	0	0		
		.20	.22	.00	.78	1.0	.01	.00	.00	0	1	0		
		1.0	.28	.00	.00	1.0	.01	.00	.00	0	1	0	0/0	0/3
20	Move end-effector over cube A	1.0	.28	.00	.00	1.0	.01	.00	.00	0	1	0		
		.83	.91	.03	.11	.91	.95	.91	.82	9	1	10		
		.81	.87	.00	.16	.91	.94	.72	.82	8	2	10	23/2	7/7
21	Push cube A and cube B close to each other	.83	.90	.03	.11	.91	.95	.55	.64	6	4	10		
		.75	.01	.00	.00	1.0	.02	.07	.07	1	14	0		
		.75	.01	.03	.00	1.0	.01	.00	.00	0	9	0	1/2	0/31
22	Move to a target position on the table without interacting with objects	.59	.00	.00	.11	1.0	.01	.11	.11	1	8	1		
		.85	.78	.00	.13	1.0	.86	.86	.86	12	2	13		
		.69	.88	.02	.27	1.0	.92	1.0	1.0	13	0	13	33/3	1/2
23	Put cube A into the drawer	.85	.84	.00	.15	1.0	.92	1.0	1.0	8	0	8		
		.16	.43	.00	.36	.30	.51	.04	.74	1	6	0		
		.26	.30	.01	.29	.52	.27	.00	.48	0	12	0	0/1	0/27

		.23	.22	.00	.27	.64	.18	.00	.36	0	9	0		
24	Stack cube A on top of cube B	.86	.61	.00	.04	.96	.81	.04	.07	1	26	0		
		.81	.61	.00	.13	.93	.75	.14	.21	2	11	0	0/4	0/51
		.95	.60	.05	.02	1.0	.76	.07	.07	1	14	0		
23	Open the drawer, pick up cube A and place it inside the drawer	.19	.23	.00	.72	1.0	.39	.00	.00	0	4	0		
		.29	.11	.00	.54	.75	.25	.25	.50	1	2	0	0/1	0/10
		.41	.01	.00	.50	1.0	.03	.00	.00	0	4	0		
24	Pick up cube A, place onto cube B to make a stack	.24	.33	.00	.24	.60	.39	.10	.30	0	6	0		
		.41	.34	.00	.10	.50	.43	.30	.60	2	3	0	0/3	0/12
		.52	.28	.14	.10	.67	.43	.33	.33	1	3	0		

D.3 PROMPTS

```
[SYSTEM]:
You are a task designer trying to propose meaningful tasks based on a specific environment in simulation.
1. The environment will be described with its source code. Note the comments around codes to understand their initial status.
2. These tasks are meant to be used to train a robot to acquire skills in the given environment.
3. Once any task is learned by the robot, it becomes a new skill of the robot.
4. The new proposed tasks should be meaningful, primitive (atomic), incremental to learn, independent of each other, and diverse.
5. You should avoid proposing the same tasks that were previously either completed or failed.
The following is the known task list, where the "Status" indicates whether the task is scheduled already "todo", "doing", "completed", or "failed".
{tasks}

Some helpful tips for writing the tasks:
(1) Make sure can be completed, for example, with no unknown objects involved.
(2) The success of the tasks should be measurable with current observations, e.g. object positions. The environment doesn't support collision detection yet. For simplicity, ignore collision.
(3) Use fewer objects in one atomic task.
(4) Make sure to not mention any joint positions as the task goal.

Let's work this out in a step-by-step way to be sure we have the right answer.

[USER]:
The Python environment is
```python
{env_obs_code_string}
```

Use the given "target position" instead of "random position" or vague "specific position" for clarity.
```

Figure D.4: A snippet of task proposal prompt, where {tasks} indicates the position to insert previously explored task instructions, and {env_obs_code_string} holds the place for incoming source codes for the environment.

In this appendix, we provide the prompt snippets used by ASD for various purposes: task proposal (Figure D.4, see § 7.3.1) and skill learning (see § 7.3.2), which includes generating success functions (Figure D.5), generating reward functions (Figure D.6), feedback iterations (Figure D.7), and GPT-4V behavior assessment (Figure D.8).

When interacting with a conversational LLM, there are typically three roles: *system*, *user*, and *assistant*.

- **User:** Provides the primary input, usually in the form of queries, requests, or

instructions for the LLM to process.

- **System:** Serves as a higher-priority input, offering contextual guidelines, constraints, or objectives that the LLM should adhere to throughout the interaction. It sets the overarching rules or tone for the assistant's behavior.
- **Assistant:** Represents the LLM itself, generating responses based on the given input and system instructions.

This structured framework ensures the conversation remains coherent and aligned with the intended goals.

```
[SYSTEM]:
You are a function engineer trying to write success condition functions to determine the accomplishment of reinforcement learning tasks.
The success condition functions help compute the success of given tasks.
Your success condition function should use useful variables as inputs, according to the scenario and task instructions.
As an example, the success condition function signature can be:
```python
@configclass
class SuccessCfg:
 success = RewTerm(
 func=object_is_lifted,
 weight=30.0,
)
```

Follow the format of this signature when writing your own for later tasks.
Please make sure that the code is compatible with Pytorch, for example, use torch tensor instead of NumPy array.
Make sure any new tensor or variable you introduce is on the same device as the input tensors.
You are not allowed to import other Python modules.

[USER]:
The Python environment is
```python
{task_obs_code_string}
```

To prepare for pre-conditions, previously executed skills are:
{precedent_skills}
And, the next subtask is to learn {task_description}.

Knowing this information, now please write a success deterministic function for this task {task_description}.

- Remember to explicitly configure `SuccessCfg` so that I can directly copy the code.
- Any introduced tensor constant should be on the device GPU, for example `c = torch.tensor([1., 2.]).cuda()`
```

Figure D.5: A prompt snippet for generating success functions given the environment information, where {precedent_skills} holds the preceding executed skills as background information for the LLM to know the state that the learning will start with.

[SYSTEM]:

You are a reward engineer trying to write reward functions to solve reinforcement learning tasks as effectively as possible. Your goal is to write a reward function for the environment that will help the agent learn the task described in the text. Your reward function should use useful variables from the environment as inputs. As an example, the reward function signature can be:

```
```python
@configclass
class RewardsCfg:
 reached_reward = RewTerm(
 func=to_reach_cube_a,
 weight=1.0,
)
```
```

Follow the format of this signature when writing your own for later tasks. Note that every `func` should be implemented by yourself.

Please make sure that the code is compatible with PyTorch (e.g., use torch tensor instead of numpy array).

Make sure any new tensor or variable you introduce is on the same device as the input tensors. You are not allowed to import other Python modules.

To incrementally guide a reinforcement learning agent in a curriculum, you should write many sub-reward functions, encoded individually.

The learning agent will be rewarded by the weighted sum of those sub-reward functions.

Some helpful tips for writing the reward function code:

- (1) You may find it helpful to normalize the reward to a fixed range by applying transformations like `torch.exp` to the reward components
- ...

[USER]:

The Python environment is

```
```python
{task_obs_code_string}
```
```

To prepare for pre-conditions, previously executed skills are:

```
{precedent_skills}
```

Knowing these information, now please write a shaped reward function for the task: {task_description}.

Let's work this out in a step by step way to be sure we have the right answer.

- Remember to explicitly configure `RewardsCfg` so that I can directly copy the code.
- Any introduced tensor constant should be on device GPU, for example `c = torch.tensor([1., 2.]).cuda()`

Figure D.6: A prompt snippet for generating reward functions. It is similar to the prompt for the success function given the environment information.

Feedback Errors

[USER]:
Executing the reward function code above has the following error: {traceback_msg}. Please fix the bug and provide a new, improved reward function!

Feedback Statistics

[USER]:
We trained a RL policy using the provided reward function code and tracked the values of the individual components in the reward function as well as global policy metrics such as success rates and episode lengths after every {epoch_freq} epochs and the maximum, mean, minimum values encountered: {statistics}

Here is the output from GPT-4v when describing the trained behavior:
 ```GPT-4v-Caption  
 {gpt4v\_description}  
 ```

Figure D.7: A prompt snippet for feeding back learning statistics and GPT-4V response for reward function iteration. *Top:* if the code ends with an execution error, e.g. Pytorch tensor shape mismatch, the error messages will be fed back so the LLM can revise for a better one. *Bottom:* if the code runs without bugs, the learning results will be collected for the iteration of reward functions, potentially resulting in more efficient ones.

[SYSTEM]:
You are a professional expert to analyze robotic behaviors in a simulated environment.
Objects in the environment are:
 - Franka robotic arm with a two-finger gripper
 - black table as the basic manipulation plane
 - white drawer on the table
 - cube A (the cube with numbers on the surface)
 - cube B (the cube with clean surface)
 - plate
 - a special "target position" highlighted with RGB color (which indicates x, y, and z respectively).
 This position is an imagined point to let the robot play with.

Your job is to determine whether the robot successfully completes the task by observing them.
 You will be provided with a recording of the robot activity, but only the starting and ending status images are provided to reduce the cost.
 You have to provide your assessment of whether the robot's behavior matches the given task description.
 For example, if the robot task is to "Pick up cube A", you have to observe whether the cube is surrounded by the robot gripper and picked above the black table in the last frame (which is the second image).

Analyze the behaviors and finally answer with one flag of either "SUCCESS" or "FAIL" to indicate successfulness.

Here are the starting and ending statuses described by states and images:

1. State

First frame (initial state):
{first_frame}

Last frame (end state):
{end_frame}

2. Image

<<<IMAGE 1>>>
 <<<IMAGE 2>>>

Figure D.8: A snippet of prompts for robot behavior assessment using GPT-4V, where the {*_frame} are state observations of the defined key frames of the recorded behavior video, and <<IMAGE X>> holds the place for corresponding key frame images.

BIBLIOGRAPHY

References are listed in alphabetical order by author.

- [Abb08] Pieter Abbeel. ‘Apprenticeship Learning and Reinforcement Learning with Application to Robotic Control’. PhD thesis. Stanford University, 2008 (cited on page 44).
- [Aga+21] Vinayak Agarwal, Maddie Cusimano, James Traer, and Josh McDermott. ‘Object-Based Synthesis of Scraping and Rolling Sounds Based on Non-Linear Physical Constraints’. In: *The 24th International Conference on Digital Audio Effects (DAFx-21)*. Sept. 2021 (cited on page 63).
- [Agl12] David Agler. *Symbolic logic: syntax, semantics, and proof*. Lanham, Md: Rowman & Littlefield Publishers, Dec. 2012 (cited on pages 88, 89).
- [ACM15] Pulkit Agrawal, João Carreira, and Jitendra Malik. ‘Learning to See by Moving’. In: *2015 IEEE International Conference on Computer Vision*. 2015, pp. 37–45 (cited on page 58).
- [Ahn+22] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. ‘Do as I Can, Not as I Say: Grounding Language in Robotic Affordances’. In: arXiv:2204.01691 (Aug. 2022), p. 2204.01691 (cited on pages 44, 45, 71–73, 75, 85, 87, 100, 103, 110, 134).
- [Akk+23] Cem Akkus, Luyang Chu, Vladana Djakovic, Steffen Jauch-Walser, Philipp Koch, Giacomo Loss, Christopher Marquardt, Marco Moldovan, Nadja Sauter, Maximilian Schneider, et al. ‘Multimodal Deep Learning’. In: arXiv:2301.04856 (Jan. 2023), p. 2301.04856 (cited on page 72).
- [AZ17] Relja Arandjelovic and Andrew Zisserman. ‘Look, Listen and Learn’. In: *Proceedings of the IEEE International Conference on Computer Vision*. Oct. 2017, pp. 609–617 (cited on page 58).
- [AZ18] Relja Arandjelovic and Andrew Zisserman. ‘Objects That Sound’. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 435–451 (cited on pages 56, 58, 61).
- [Arm+23] Philip Arm, Gabriel Waibel, Jan Preisig, Turcan Tuna, Ruyi Zhou, Valentin Bickel, Gabriela Ligeza, Takahiro Miki, Florian Kehl, Hendrik Kolvenbach, et al. ‘Scientific Exploration of Challenging Planetary Analog Environments with a Team of Legged Robots’. In: *Science Robotics* 8.80 (2023), eade9548 (cited on page 9).
- [Ata15] Nikolay Asenov Atanasov. *Active Information Acquisition with Mobile Robots*. University of Pennsylvania, 2015 (cited on pages 12, 72, 73).

- [Atr+10] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. ‘Multimodal Fusion for Multimedia Analysis: A Survey’. In: *Multimedia Systems* 16.6 (Nov. 2010), pp. 345–379 (cited on page 28).
- [Azp+23] Héctor Azpúrua, Maíra Saboia, Gustavo M. Freitas, Lillian Clark, Aliakbar Agha-mohammadi, Gustavo Pessin, Mario F. M. Campos, and Douglas G. Macharet. ‘A Survey on the Autonomous Exploration of Confined Subterranean Spaces: Perspectives from Real-World and Industrial Robotic Deployments’. In: *Robotics and Autonomous Systems* 160 (Feb. 2023), p. 104304 (cited on pages 9, 23, 30, 133).
- [BSK21] Akhil Bagaria, Jason K Senthil, and George Konidaris. ‘Skill Discovery for Exploration and Planning Using Deep Skill Graphs’. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 521–531 (cited on pages 16, 120, 134).
- [BD06] Tim Bailey and Hugh Durrant-Whyte. ‘Simultaneous Localization and Mapping (SLAM): Part II’. In: *IEEE Robotics and Automation Magazine* 13.3 (Sept. 2006), pp. 108–117 (cited on pages 10, 25).
- [Baj88] Ruzena Bajcsy. ‘Active Perception’. In: *Proceedings of the IEEE* 76.8 (Aug. 1988), pp. 966–1005 (cited on page 121).
- [Ban+23] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. ‘A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity’. In: arXiv:2302.04023 (Nov. 2023), p. 2302.04023 (cited on page 85).
- [BA03] David Barber and Felix Agakov. ‘Information Maximization in Noisy Channels : A Variational Approach’. In: *Advances in Neural Information Processing Systems*. Ed. by S. Thrun, L. Saul, and B. Schölkopf. Vol. 16. MIT Press, 2003 (cited on page 17).
- [Bar+06] Kobus Barnard, Keiji Yanai, Matthew Johnson, and Prasad Gabbur. ‘Cross Modal Disambiguation’. In: *Toward Category-Level Object Recognition* (2006), pp. 238–257 (cited on pages 70, 72).
- [Bel+18] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. ‘Mutual Information Neural Estimation’. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 531–540 (cited on page 122).
- [Bla+24] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. ‘ π_0 : A Vision-Language-Action Flow Model for General Robot Control’. In: arXiv:2410.24164 (Oct. 2024), p. 2410.24164 (cited on page 13).

- [BI20] Nicolas Bougie and Ryutaro Ichise. ‘Skill-Based Curiosity for Intrinsically Motivated Reinforcement Learning’. In: *Machine Learning* 109.3 (Mar. 2020), pp. 493–512 (cited on page 49).
- [Bou+02] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. ‘Information Based Adaptive Robotic Exploration’. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vol. 1. Sept. 2002, 540–545 vol.1 (cited on page 14).
- [Bro+23] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. ‘RT-1: Robotics Transformer for Real-World Control at Scale’. In: arXiv:2212.06817 (Aug. 2023), p. 2212.06817 (cited on page 47).
- [Bub+23] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. ‘Sparks of Artificial General Intelligence: Early Experiments with GPT-4’. In: arXiv:2303.12712 (Apr. 2023), p. 2303.12712 (cited on page 85).
- [Bur+19a] Yuri Burda, Harrison Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. ‘Large-Scale Study of Curiosity-Driven Learning’. In: *7th International Conference on Learning Representations*. May 2019, pp. 1–17 (cited on pages 14, 39, 48, 49, 52, 56, 59–61).
- [Bur+19b] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. ‘Exploration by Random Network Distillation’. In: *International Conference on Learning Representations*. 2019 (cited on pages 59, 61).
- [Cal+18] Roberto Calandra, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H Adelson, and Sergey Levine. ‘More than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch’. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 3300–3307 (cited on page 58).
- [Cas+23] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. ‘Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback’. In: *Transactions on Machine Learning Research* (2023) (cited on pages 83, 121).
- [CK23] Carlos Celemin and Jens Kober. ‘Knowledge-and Ambiguity-Aware Robot Learning from Corrective and Evaluative Feedback’. In: *Neural Computing and Applications* (2023), pp. 1–19 (cited on page 70).
- [Cel+23] Batuhan Celik, Alper Ahmetoglu, Emre Ugur, and Erhan Oztup. ‘Developmental Scaffolding with Large Language Models’. In: *IEEE International Conference on Development and Learning*. arXiv, Sept. 2023 (cited on pages 50, 101–103).

- [Cem+25] Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. ‘Why Do Multi-Agent Llm Systems Fail?’ In: arXiv:2503.13657 (2025) (cited on pages 51, 133).
- [Che+23a] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S. Ryoo, Austin Stone, and Daniel Kappler. ‘Open-Vocabulary Queryable Scene Representations for Real World Planning’. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 11509–11522 (cited on page 37).
- [Che+20] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. ‘Soundspaces: Audio-Visual Navigation in 3d Environments’. In: *European Conference on Computer Vision*. Springer. 2020, pp. 17–36 (cited on pages 26, 58).
- [Che+23b] Xi Chen, Xiao Wang, Soravit Changpinyo, Aj Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. ‘PaLI: A Jointly-Scaled Multilingual Language-Image Model’. In: *The Eleventh International Conference on Learning Representations*. 2023 (cited on page 37).
- [Che+24] Yongchao Chen, Jacob Arkin, Charles Dawson, Yang Zhang, Nicholas Roy, and Chuchu Fan. ‘AutoTAMP: Autoregressive Task and Motion Planning with LLMs as Translators and Checkers’. In: arXiv:2306.06531 (Mar. 2024), p. 2306.06531 (cited on page 12).
- [Chi+23] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023 (cited on page 93).
- [Chr+17] Paul 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*. Vol. 30. 2017 (cited on pages 46, 83, 121, 123).
- [CS15] Christos Christodouloupoulos and Mark Steedman. ‘A Massively Parallel Corpus: The Bible in 100 Languages’. In: *Language Resources and Evaluation* 49.2 (June 2015), pp. 375–395 (cited on page 100).
- [Chu+24a] Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, Wenhao Lu, and Stefan Wermter. ‘Large Language Models for Orchestrating Bimanual Robots’. In: *The 2024 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 2024 (cited on pages 124, 126, 201).
- [Chu+24b] Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, and Stefan Wermter. ‘Accelerating Reinforcement Learning of Robotic Manipulations via Feedback from Large Language Models’. In: *7th Conference on Robot Learning (CoRL 2023) Workshop on Bridging the Gap Between Cognitive Science and Robot Learning in the Real World: Progresses and New*

Directions (CRL_WS), Atlanta, Georgia USA. 2024 (cited on pages 19, 51, 123, 201).

- [Chu+25] Kun Chu, Xufeng Zhao, Cornelius Weber, and Stefan Wermter. ‘LLM+MAP: Bimanual Robot Task Planning Using Large Language Models and Planning Domain Definition Language’. In: arXiv:2503.17309 (Mar. 2025), p. 2503.17309 (cited on pages 12, 46, 124, 126, 127, 133, 134, 200).
- [Cla+18] Samuel Clarke, Travers Rhodes, Christopher G. Atkeson, and Oliver Kroemer. ‘Learning Audio Feedback for Estimating Amount and Flow of Granular Material’. In: *Proceedings of the 2nd Conference on Robot Learning*. Ed. by Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto. Vol. 87. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 529–550 (cited on page 56).
- [Cob+21] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. ‘Training Verifiers to Solve Math Word Problems’. In: arXiv:2110.14168 (Nov. 2021), p. 2110.14168 (cited on page 92).
- [Col+21] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. ‘A Review of Physics Simulators for Robotic Applications’. In: *IEEE Access* 9 (2021), pp. 51416–51431 (cited on page 22).
- [Con25] MoveIt Contributors. *MoveIt 2 Documentation — MoveIt Documentation: Rolling Documentation*. 2025 (cited on page 19).
- [CB21] Erwin Coumans and Yunfei Bai. ‘PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning’. In: *Github Repository* (2021) (cited on page 22).
- [CS22] Antonia Creswell and Murray Shanahan. ‘Faithful Reasoning Using Large Language Models’. In: arXiv:2208.14271 (Aug. 2022), p. 2208.14271 (cited on page 86).
- [CSH23] Antonia Creswell, Murray Shanahan, and Irina Higgins. ‘Selection-Inference: Exploiting Large Language Models for Interpretable Logical Reasoning’. In: *The Eleventh International Conference on Learning Representations*. 2023 (cited on pages 86, 88).
- [Cui+23] Yuchen Cui, Siddharth Karamcheti, Raj Pallethi, Nidhya Shivakumar, Percy Liang, and Dorsa Sadigh. ‘"No, to the Right"—Online Language Corrections for Robotic Manipulation via Shared Autonomy’. In: (2023), p. 2301.02555 (cited on pages 71, 73).
- [Dai+24] Zhirui Dai, Arash Asgharivaskasi, Thai Duong, Shusen Lin, Maria-Elizabeth Tzes, George Pappas, and Nikolay Atanasov. ‘Optimal Scene Graph Planning with Large Language Model Guidance’. In: arXiv:2309.09182 (Jan. 2024), p. 2309.09182 (cited on pages 12, 25, 37).

- [DTG20] Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. ‘See, Hear, Explore: Curiosity via Audio-Visual Association’. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 14961–14972 (cited on pages 14, 58–61).
- [Dee+25] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, et al. ‘DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning’. In: arXiv:2501.12948 (Jan. 2025), p. 2501.12948 (cited on pages 53, 83, 127, 131, 132).
- [Dee+24] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, et al. ‘DeepSeek-V3 Technical Report’. In: arXiv:2412.19437 (Dec. 2024), p. 2412.19437 (cited on pages 83, 127).
- [Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. ‘Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding’. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 4171–4186 (cited on page 36).
- [Dij59] E. W. Dijkstra. ‘A Note on Two Problems in Connexion with Graphs’. In: *Numerische Mathematik* 1.1 (Dec. 1959), pp. 269–271 (cited on page 11).
- [Dim22] Mariella Dimiccoli. ‘Recognizing Object Surface Material from Impact Sounds for Robot Manipulation’. In: *IROS* (2022) (cited on pages 75, 77, 78).
- [Din+23] Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. ‘Task and Motion Planning with Large Language Models for Object Rearrangement’. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 2086–2092 (cited on pages 12, 46, 100, 110).
- [DZ17] Carl Doersch and Andrew Zisserman. ‘Multi-Task Self-Supervised Visual Learning’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2051–2060 (cited on pages 56, 58, 60).
- [Dri+23] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. ‘PaLM-e: An Embodied Multimodal Language Model’. In: *Proceedings of the 40th International Conference on Machine Learning*. ICML’23. Honolulu, Hawaii, USA: JMLR.org, 2023 (cited on page 45).
- [DGI21] Yilun Du, Chuang Gan, and Phillip Isola. ‘Curious Representation Learning for Embodied Intelligence’. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 10408–10417 (cited on pages 13, 14, 60).

- [Du+23] 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: (2023), p. 2302.06692 (cited on pages 101, 102).
- [Dua+22] Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. ‘A Survey of Embodied AI: From Simulators to Research Tasks’. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 6.2 (Apr. 2022), pp. 230–244 (cited on pages 22, 60, 63).
- [DB06] Hugh DurrantWhyte and Tim Bailey. ‘Simultaneous Localization and Mapping: Part I’. In: *IEEE Robotics and Automation Magazine* 13.2 (June 2006), pp. 99–110 (cited on pages 10, 25).
- [Ehs+21] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ‘ManipulaTHOR: A Framework for Visual Object Manipulation’. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, June 2021, pp. 4495–4504 (cited on page 22).
- [Eis+21] Aaron Eisermann, Jae Hee Lee, Cornelius Weber, and Stefan Wermter. ‘Generalization in Multimodal Language Learning from Simulation’. In: *2021 International Joint Conference on Neural Networks*. IEEE. 2021, pp. 1–8 (cited on page 56).
- [Elm90] Jeffrey L. Elman. ‘Finding Structure in Time’. In: *Cognitive Science* 14.2 (Apr. 1990), pp. 179–211 (cited on page 35).
- [Epp+18] Manfred Eppe, Matthias Kerzel, Erik Strahl, and Stefan Wermter. ‘Deep Neural Object Analysis by Interactive Auditory Exploration with a Humanoid Robot’. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 284–289 (cited on page 28).
- [EGI20] Amedeo Roberto Esposito, Michael Gastpar, and Ibrahim Issa. ‘Robust Generalization via F-mutual Information’. In: *2020 IEEE International Symposium on Information Theory (ISIT)*. 2020, pp. 2723–2728 (cited on page 122).
- [Eys+19] 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*. 2019 (cited on pages 15, 16, 49, 59, 101, 102, 105, 121).
- [Eys+22] Benjamin Eysenbach, Tianjun Zhang, Ruslan Salakhutdinov, and Sergey Levine. *Contrastive Learning as Goal-Conditioned Reinforcement Learning*. 2022 (cited on page 13).
- [Gäd+22] Connor Gäde, Matthias Kerzel, Erik Strahl, and Stefan Wermter. ‘Sim-to-Real Neural Learning with Domain Randomisation for Humanoid Robot Grasping’. In: *Artificial Neural Networks and Machine Learning – ICANN 2022*. Ed. by Elias Pimenidis, Plamen Angelov, Chrisina Jayne, Antonios Papaleonidas, and Mehmet Aydin. Cham: Springer International Publishing, 2022, pp. 342–354 (cited on page 117).

- [Gan+20a] Chuang Gan, Xiaoyu Chen, Phillip Isola, Antonio Torralba, and Joshua B. Tenenbaum. ‘Noisy Agents: Self-Supervised Exploration by Predicting Auditory Events’. In: abs/2007.13729 (2020), p. 2007.13729 (cited on pages 26, 59, 65, 67).
- [Gan+21] Chuang Gan, Jeremy Schwartz, Seth Alter, Damian Mrowca, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwalder, Nick Haber, et al. ‘ThreeDWorld: A Platform for Interactive Multi-Modal Physical Simulation’. In: *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*. 2021 (cited on pages 9, 20, 57, 63).
- [Gan+20b] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum. ‘Look, Listen, and Act: Towards Audio-Visual Embodied Navigation’. In: *2020 IEEE International Conference on Robotics and Automation*. IEEE. 2020, pp. 9701–9707 (cited on page 58).
- [Gao+24a] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. ‘Physically Grounded Vision-Language Models for Robotic Manipulation’. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 12462–12469 (cited on page 10).
- [Gao+21] Ruohan Gao, Yen-Yu Chang, Shivani Mall, Li Fei-Fei, and Jiajun Wu. ‘ObjectFolder: A Dataset of Objects with Implicit Visual, Auditory, and Tactile Representations’. In: *5th Annual Conference on Robot Learning*. 2021 (cited on page 63).
- [Gao+20] Ruohan Gao, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman. ‘Visualechoes: Spatial Image Representation Learning through Echolocation’. In: *European Conference on Computer Vision*. Springer. 2020, pp. 658–676 (cited on page 58).
- [Gao+24b] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. ‘Retrieval-Augmented Generation for Large Language Models: A Survey’. In: arXiv:2312.10997 (Mar. 2024), p. 2312.10997 (cited on page 109).
- [GL20] Artur d’Avila Garcez and Luis C. Lamb. ‘Neurosymbolic AI: The 3rd Wave’. In: arXiv:2012.05876 (Dec. 2020), p. 2012.05876 (cited on page 88).
- [Gei+25] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. ‘Scaling up Test-Time Compute with Latent Reasoning: A Recurrent Depth Approach’. In: arXiv:2502.05171 (Feb. 2025), p. 2502.05171 (cited on page 83).
- [GP00] Eleanor Jack Gibson and Anne D Pick. *An Ecological Approach to Perceptual Learning and Development*. Oxford University Press, USA, 2000 (cited on page 38).

- [GRW17] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. ‘Variational Intrinsic Control’. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017 (cited on pages 39, 49, 105, 122, 132).
- [Gro+21] Oliver Groth, Markus Wulfmeier, Giulia Vezzani, Vibhavari Dasagi, Tim Hertweck, Roland Hafner, Nicolas Heess, and Martin Riedmiller. ‘Is Curiosity All You Need? On the Utility of Emergent Behaviours from Curious Exploration’. In: arXiv:2109.08603 (Sept. 2021), p. 2109.08603 (cited on page 14).
- [Gu+24] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. ‘ConceptGraphs: Open-Vocabulary 3D Scene Graphs for Perception and Planning’. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 5021–5028 (cited on pages 12, 25, 37).
- [Gu+22] 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*. 2022 (cited on pages 74, 77).
- [Gup+21] Abhishek Gupta, Justin Yu, Tony Z. Zhao, Vikash Kumar, Aaron Rovinsky, Kelvin Xu, Thomas Devlin, and Sergey Levine. ‘Reset-Free Reinforcement Learning via Multi-Task Learning: Learning Dexterous Manipulation Behaviors without Human Intervention’. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 6664–6671 (cited on page 109).
- [HS18] David Ha and Jürgen Schmidhuber. ‘World Models’. In: abs/1803.10122 (2018), p. 1803.10122 (cited on pages 25, 29).
- [HFS23] Huy Ha, Pete Florence, and Shuran Song. ‘Scaling up and Distilling down: Language-Guided Robot Skill Acquisition’. In: arXiv:2307.14535 (Sept. 2023), p. 2307.14535 (cited on page 38).
- [Hao+23] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. ‘Reasoning with Language Model Is Planning with World Model’. In: arXiv:2305.14992 (Oct. 2023), p. 2305.14992 (cited on page 31).
- [HNR68] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. ‘A Formal Basis for the Heuristic Determination of Minimum Cost Paths’. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (July 1968), pp. 100–107 (cited on page 11).
- [Heb+24] Konstantin Hebenstreit, Robert Praas, Louis P. Kieseewetter, and Matthias Samwald. ‘A Comparison of Chain-of-Thought Reasoning Strategies across Datasets and Models’. In: *PeerJ Computer Science* 10 (Apr. 2024), e1999 (cited on page 53).
- [Hel14] Malte Helmert. *An Introduction to PDDL*. 2014 (cited on pages 39, 40).

- [Hig+20] Focko L Higgen, Philipp Ruppel, Michael Görner, Matthias Kerzel, Norman Hendrich, Jan Feldheim, Stefan Wermter, Jianwei Zhang, and Christian Gerloff. ‘Crossmodal Pattern Discrimination in Humans and Robots: A Visuo-Tactile Case Study’. In: *Frontiers in Robotics and AI* 7 (2020), p. 540565 (cited on page 56).
- [HE16] Jonathan Ho and Stefano Ermon. ‘Generative Adversarial Imitation Learning’. In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016 (cited on page 44).
- [Höf+21] Sebastian Höfer, Kostas Bekris, Ankur Handa, Juan Camilo Gamboa, Melissa Mozifian, Florian Golemo, Chris Atkeson, Dieter Fox, Ken Goldberg, John Leonard, et al. ‘Sim2Real in Robotics and Automation: Applications and Challenges’. In: *IEEE Transactions on Automation Science and Engineering* 18.2 (Apr. 2021), pp. 398–400 (cited on pages 16, 117).
- [Hog+21] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d’Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. ‘Knowledge Graphs’. In: *ACM Computing Surveys (csur)* 54.4 (2021), pp. 1–37 (cited on page 36).
- [Hon+24] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiwu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. ‘MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework’. In: *The Twelfth International Conference on Learning Representations*. 2024 (cited on pages 9, 133).
- [Hu+20] Yujing Hu, Weixun Wang, Hangtian Jia, Yixiang Wang, Yingfeng Chen, Jianye Hao, Feng Wu, and Changjie Fan. ‘Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping’. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15931–15941 (cited on page 58).
- [Hua+24] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. ‘Large Language Models Cannot Self-Correct Reasoning Yet’. In: *The Twelfth International Conference on Learning Representations*. 2024 (cited on pages 88, 90).
- [Hua+22] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. ‘Language Models as Zero-Shot Planners: Extracting Actionable Knowledge for Embodied Agents’. In: abs/2201.07207 (2022), p. 2201.07207 (cited on pages 73, 75).
- [Hua+23] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. ‘VoxPoser: Composable 3D Value Maps for Robotic Manipulation with Language Models’. In: *7th Annual Conference on Robot Learning*. 2023 (cited on pages 103, 112, 113).

- [Int+25] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. ‘ $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization’. In: arXiv:2504.16054 (Apr. 2025), p. 2504.16054 (cited on page 13).
- [Jae+22] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. ‘Perceiver IO: A General Architecture for Structured Inputs & Outputs’. In: *International Conference on Learning Representations*. 2022 (cited on page 60).
- [Jae+21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. ‘Perceiver: General Perception with Iterative Attention’. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4651–4664 (cited on pages 57, 60).
- [Jam+20] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. ‘Rlbench: The Robot Learning Benchmark & Learning Environment’. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3019–3026 (cited on pages 21, 123).
- [Jan+24] Dae-Sung Jang, Doo-Hyun Cho, Woo-Cheol Lee, Seung-Keol Ryu, Byeongmin Jeong, Minji Hong, Minjo Jung, Minchae Kim, Minjoon Lee, SeungJae Lee, et al. ‘Unlocking Robotic Autonomy: A Survey on the Applications of Foundation Models’. In: *International Journal of Control, Automation and Systems* 22.8 (Aug. 2024), pp. 2341–2384 (cited on pages 9, 24, 45).
- [JG15] Dinesh Jayaraman and Kristen Grauman. ‘Learning Image Representations Tied to Ego-Motion’. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1413–1421 (cited on page 58).
- [Jia+24] Hanxiao Jiang, Binghao Huang, Ruihai Wu, Zhuoran Li, Shubham Garg, Hooshang Nayyeri, Shenlong Wang, and Yunzhu Li. ‘RoboEXP: Action-Conditioned Scene Graph via Interactive Exploration for Robotic Manipulation’. In: *8th Annual Conference on Robot Learning*. 2024 (cited on pages 12, 25, 37, 50, 52).
- [Jia+19] Yuqian Jiang, Shiqi Zhang, Piyush Khandelwal, and Peter Stone. ‘Task Planning in Robotics: An Empirical Comparison of PDDL-based and ASP-based Systems’. In: arXiv:1804.08229 (Feb. 2019), p. 1804.08229 (cited on pages 12, 39, 40).
- [Joh+15] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. ‘Image Retrieval Using Scene Graphs’. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, June 2015, pp. 3668–3678 (cited on pages 12, 25, 37).

- [Jun+22] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. ‘Maieutic Prompting: Logically Consistent Reasoning with Recursive Explanations’. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1266–1279 (cited on pages 87, 88, 90).
- [Kam+22] Pierre-Alexandre Kamienny, Jean Tarbouriech, Sylvain Lamprier, Alessandro Lazaric, and Ludovic Denoyer. ‘Direct Then Diffuse: Incremental Unsupervised Skill Discovery for State Covering and Goal Reaching’. In: arXiv:2110.14457 (Apr. 2022), p. 2110.14457 (cited on pages 16, 121).
- [KVM24] Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. ‘SMART-LLM: Smart Multi-Agent Robot Task Planning Using Large Language Models’. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2024, pp. 12140–12147 (cited on page 133).
- [KF11] Sertac Karaman and Emilio Frazzoli. ‘Sampling-Based Algorithms for Optimal Motion Planning’. In: *International Journal of Robotics Research* 30.7 (June 2011), pp. 846–894 (cited on page 11).
- [KL20] Benjamin Karg and Sergio Lucia. ‘Efficient Representation and Approximation of Model Predictive Control Laws via Deep Learning’. In: *IEEE Transactions on Cybernetics* 50.9 (Sept. 2020), pp. 3866–3878 (cited on page 42).
- [KG17] Alex Kendall and Yarin Gal. ‘What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?’ In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017 (cited on page 32).
- [Ker+23] Matthias Kerzel, Philipp Allgeuer, Erik Strahl, Nicolas Frick, Jan-Gerrit Habekost, Manfred Eppe, and Stefan Wermter. ‘NICOL: A Neuro-Inspired Collaborative Semi-Humanoid Robot That Bridges Social Interaction and Reliable Manipulation’. In: *IEEE Access : Practical Innovations, Open Solutions* 11 (2023), pp. 123531–123542 (cited on pages 18, 19, 77).
- [KPK21] Jaekyeom Kim, Seohong Park, and Gunhee Kim. ‘Unsupervised Skill Discovery with Bottleneck Option Learning’. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 5572–5582 (cited on pages 16, 121).
- [Kim+24] Yeseung Kim, Dohyun Kim, Jieun Choi, Jisang Park, Nayoung Oh, and Daehyung Park. ‘A Survey on Integration of Large Language Models with Intelligent Robots’. In: *Intelligent Service Robotics* 17.5 (Sept. 2024), pp. 1091–1107 (cited on pages 9, 45).

- [KW+19] Diederik P Kingma, Max Welling, et al. ‘An Introduction to Variational Autoencoders’. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392 (cited on page 13).
- [KW22] Diederik P. Kingma and Max Welling. ‘Auto-Encoding Variational Bayes’. In: arXiv:1312.6114 (Dec. 2022), p. 1312.6114 (cited on page 13).
- [KH04] N. Koenig and A. Howard. ‘Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator’. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 3. Sept. 2004, 2149–2154 vol.3 (cited on page 22).
- [Koj+22] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. ‘Large Language Models Are Zero-Shot Reasoners’. In: *Advances in Neural Information Processing Systems*. May 2022 (cited on pages 86–89, 91–94, 98).
- [Kol+22] 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’. In: arXiv:1712.05474 (Aug. 2022), p. 1712.05474 (cited on page 22).
- [KG15] Ioannis Kostavelis and Antonios Gasteratos. ‘Semantic Mapping for Mobile Robotics Tasks: A Survey’. In: *Robotics and Autonomous Systems* 66 (Apr. 2015), pp. 86–103 (cited on page 12).
- [KNK21] Oliver Kroemer, Scott Niekum, and George Konidaris. ‘A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms’. In: *Journal of Machine Learning Research* 22.1 (Jan. 2021) (cited on pages 23, 30, 39, 43, 70, 133).
- [KL00] James J. Kuffner and Steven M. LaValle. ‘RRT-connect: An Efficient Approach to Single-Query Path Planning’. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00ch37065)*. Vol. 2. 2000, 995–1001 vol.2 (cited on page 11).
- [Kum+19] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. ‘Stabilizing Off-Policy q-Learning via Bootstrapping Error Reduction’. In: *Advances in Neural Information Processing Systems* 32 (2019) (cited on page 57).
- [Kun+18] Lars Kunze, Nick Hawes, Tom Duckett, Marc Hanheide, and Tomáš Krajník. ‘Artificial Intelligence for Long-Term Robot Autonomy: A Survey’. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4023–4030 (cited on pages 9, 23).
- [Kwo+23] Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. ‘Reward Design with Language Models’. In: *International Conference on Learning Representations*. 2023 (cited on page 99).

- [Las+21a] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. ‘CIC: Contrastive Intrinsic Control for Unsupervised Skill Discovery’. In: *Deep RL Workshop NeurIPS 2021*. 2021 (cited on pages 16, 49, 121).
- [LSA20] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. ‘CURL: Contrastive Unsupervised Representations for Reinforcement Learning’. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé Iii and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 5639–5650 (cited on page 13).
- [Las+21b] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. ‘URLB: Unsupervised Reinforcement Learning Benchmark’. In: *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021 (cited on pages 16, 49, 59, 60, 63, 65, 120).
- [LP23] Artem Latyshev and Aleksandr I. Panov. ‘Intrinsic Motivation in Model-Based Reinforcement Learning: A Brief Review’. In: arXiv:2301.10067 (Jan. 2023), p. 2301.10067 (cited on pages 23, 47, 49, 53).
- [LHP23] Mikko Lauri, David Hsu, and Joni Pajarinen. ‘Partially Observable Markov Decision Processes in Robotics: A Survey’. In: 39.1 (Feb. 2023), p. 2209.10342 (cited on page 3).
- [Lau+04] Paul J Laurienti, Robert A Kraft, Joseph A Maldjian, Jonathan H Burdette, and Mark T Wallace. ‘Semantic Congruence Is a Critical Factor in Multisensory Behavioral Performance’. In: *Experimental brain research* 158.4 (2004), pp. 405–414 (cited on page 56).
- [LaV98] Steven M. LaValle. ‘Rapidly-Exploring Random Trees : A New Tool for Path Planning’. In: *The annual research report* (1998) (cited on page 11).
- [LaV06] Steven M. LaValle. *Planning Algorithms*. Cambridge: Cambridge University Press, 2006 (cited on page 11).
- [LPO18] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. ‘Curiosity Driven Exploration of Learned Disentangled Goal Spaces’. In: *Proceedings of the 2nd Conference on Robot Learning*. Ed. by Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto. Vol. 87. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 487–504 (cited on page 14).
- [Lee+24] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. ‘RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback’. In: *Proceedings of the 41st International Conference on Machine Learning*. ICML’24. Vienna, Austria: JMLR.org, 2024 (cited on pages 83, 98, 121, 123).

- [Lee+19] Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. ‘Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks’. In: *2019 International Conference on Robotics and Automation*. 2019, pp. 8943–8950 (cited on page 58).
- [Lee+22] Seung Hyun Lee, Wonseok Roh, Wonmin Byeon, Sang Ho Yoon, Chanyoung Kim, Jinkyu Kim, and Sangpil Kim. ‘Sound-Guided Semantic Image Manipulation’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 3377–3386 (cited on page 72).
- [Li+25a] Guanlue Li, Xufeng Zhao, Fang Wu, and Sören Laue. ‘Joint Design of Protein Surface and Backbone Using a Diffusion Bridge Model’. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025 (cited on page 200).
- [LZZ20] Haoran Li, Qichao Zhang, and Dongbin Zhao. ‘Deep Reinforcement Learning-Based Automatic Exploration for Navigation in Unknown Environment’. In: *IEEE Transactions on Neural Networks and Learning Systems* 31.6 (June 2020), pp. 2064–2076 (cited on page 13).
- [Li+22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. ‘Blip: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation’. In: *International Conference on Machine Learning*. PMLR, 2022, p. 2201.12086 (cited on page 36).
- [Li24] Mengdi Li. ‘Active Vision for Embodied Agents Using Reinforcement Learning’. PhD thesis. Universitaet Hamburg (Germany), 2024 (cited on pages 10, 121).
- [Li+25b] Mengdi Li, Jiaye Lin, Xufeng Zhao, Wenhao Lu, Peilin Zhao, Stefan Wermter, and Di Wang. ‘Curriculum-RLAIF: Curriculum Alignment with Reinforcement Learning from AI Feedback’. In: arXiv:2505.20075 (May 2025), p. 2505.20075 (cited on pages 84, 200).
- [Li+23a] Mengdi Li, Xufeng Zhao, Kun Chu, and Wenhao Lu. *Awesome RLAIF*. 2023 (cited on pages 46, 98, 123).
- [Li+24] Xinhai Li, Jialin Li, Ziheng Zhang, Rui Zhang, Fan Jia, Tiancai Wang, Haoqiang Fan, Kuo-Kun Tseng, and Ruiping Wang. ‘RoboGSim: A Real2Sim2Real Robotic Gaussian Splatting Simulator’. In: arXiv:2411.11839 (Nov. 2024), p. 2411.11839 (cited on page 117).
- [Li+23b] Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. ‘Making Large Language Models Better Reasoners with Step-Aware Verifier’. In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. arXiv:2206.02336. May 2023, p. 2206.02336 (cited on pages 86, 87).

- [Li+25c] Zhuo Li, Junjia Liu, Zhihao Li, Zhipeng Dong, Tao Teng, Yongsheng Ou, Darwin Caldwell, and Fei Chen. ‘Language-Guided Dexterous Functional Grasping by LLM Generated Grasp Functionality and Synergy for Humanoid Manipulation’. In: *IEEE Transactions on Automation Science and Engineering* (2025), pp. 1–14 (cited on pages 38, 53).
- [Li+23c] Mengdi Li*, Xufeng Zhao*, Jae Hee Lee, Cornelius Weber, and Stefan Wermter. ‘Internally Rewarded Reinforcement Learning’. In: *Proceedings of the 40th International Conference on Machine Learning (ICML)*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 20556–20574 (cited on pages 3, 10, 14, 15, 17, 28, 43, 49, 52, 71, 82, 83, 101, 102, 120, 122, 200).
- [Lia+20] Hongzhuo Liang, Chuangchuang Zhou, Shuang Li, Xiaojian Ma, Norman Hendrich, Timo Gerkmann, Fuchun Sun, and Jianwei Zhang. ‘Robust Robotic Pouring Using Audition and Haptics’. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2020), pp. 10880–10887 (cited on page 56).
- [Lia+22] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. ‘Code as Policies: Language Model Programs for Embodied Control’. In: *Workshop on Language and Robotics at CoRL 2022*. 2022, p. 2209.07753 (cited on pages 46, 103, 112).
- [Lig+24] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. ‘Let’s Verify Step by Step’. In: *The Twelfth International Conference on Learning Representations*. arXiv:2305.20050. 2024, p. 2305.20050 (cited on pages 51, 87).
- [Lin+17] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. ‘Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems’. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 158–167 (cited on page 92).
- [Lin+23] Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. ‘Deductive Verification of Chain-of-Thought Reasoning’. In: *Thirty-Seventh Conference on Neural Information Processing Systems*. arXiv:2306.03872. 2023, p. 2306.03872 (cited on pages 86–88).
- [Liu+23] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. ‘LLM+P: Empowering Large Language Models with Optimal Planning Proficiency’. In: arXiv:2304.11477 (Sept. 2023), p. 2304.11477 (cited on page 46).
- [Liu+20] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. ‘On the Variance of the Adaptive Learning Rate and Beyond’. In: *Proceedings of the Eighth International Conference on Learning Representations*. Apr. 2020 (cited on page 65).

- [LBS23] Zeyi Liu, Arpit Bahety, and Shuran Song. ‘REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction’. In: *7th Annual Conference on Robot Learning*. 2023, p. 2306.15724 (cited on pages 52, 107).
- [Liu+25a] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. ‘Understanding R1-zero-like Training: A Critical Perspective’. In: arXiv:2503.20783 (2025) (cited on page 98).
- [Liu+25b] Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. ‘Inference-Time Scaling for Generalist Reward Modeling’. In: arXiv:2504.02495 (Apr. 2025), p. 2504.02495 (cited on pages 83, 132).
- [LLA21] Iker Lluvia, Elena Lazkano, and Ander Ansuaategi. ‘Active Mapping and Robot Exploration: A Survey’. In: *Sensors* 21.7 (Jan. 2021), p. 2445 (cited on pages 9, 10, 12, 23, 25, 27, 30, 52).
- [Lu+22] Jiayi Lu, Shuai Han, Shuai Lü, Meng Kang, and Junwei Zhang. ‘Sampling Diversity Driven Exploration with State Difference Guidance’. In: *Expert Systems with Applications* 203 (2022), p. 117418 (cited on pages 14, 49).
- [Lu+20] Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. ‘Dying ReLU and Initialization: Theory and Numerical Examples’. In: *Communications in Computational Physics* 28.5 (2020), pp. 1671–1706 (cited on page 61).
- [Lu+24] Wenhao Lu, Xufeng Zhao, Thilo Fryen, Jae Hee Lee, Mengdi Li, Sven Magg, and Stefan Wermter. ‘Causal State Distillation for Explainable Reinforcement Learning’. In: *Proceedings of the Third Conference on Causal Learning and Reasoning (CLear)*. Ed. by Francesco Locatello and Vanessa Didelez. Vol. 236. Proceedings of Machine Learning Research. PMLR, Apr. 2024, pp. 106–142 (cited on pages 127, 128, 201).
- [Lu+23] Wenhao Lu, Xufeng Zhao, Sven Magg, Martin Gromniak, Mengdi Li, and Stefan Wermter. ‘A Closer Look at Reward Decomposition for High-Level Robotic Explanations’. In: *2023 IEEE International Conference on Development and Learning (ICDL)*. 2023, pp. 429–436 (cited on pages 127, 128, 201).
- [Lu+25] Wenhao Lu, Xufeng Zhao, Josua Spisak, Jae Hee Lee, and Stefan Wermter. ‘Mental Modelling of Reinforcement Learning Agents by Language Models’. In: *Transactions on Machine Learning Research (TMLR)* (2025), pp. 2835–8856 (cited on pages 31, 128, 129, 200).
- [Luo+25] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. ‘O1-Pruner: Length-Harmonizing Fine-Tuning for O1-like Reasoning Pruning’. In: arXiv:2501.12570 (Jan. 2025), p. 2501.12570 (cited on page 131).

- [Lyn+22] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. ‘Interactive Language: Talking to Robots in Real Time’. In: arXiv:2210.06407 (2022) (cited on page 71).
- [Ma+24a] Runyu Ma, Jelle Luijkx, Zlatan Ajanovic, and Jens Kober. ‘ExploRLLM: Guiding Exploration in Reinforcement Learning with Large Language Models’. In: arXiv:2403.09583 (Mar. 2024), p. 2403.09583 (cited on page 50).
- [Ma+23] Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. ‘Liv: Language-Image Representations and Rewards for Robotic Control’. In: *International Conference on Machine Learning*. PMLR, 2023, pp. 23301–23320 (cited on pages 116, 133).
- [Ma+24b] 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: *The Twelfth International Conference on Learning Representations*. 2024, p. 2310.12931 (cited on pages 22, 52, 99, 101, 103–105, 107).
- [Mad+23] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. ‘Self-Refine: Iterative Refinement with Self-Feedback’. In: *Thirty-Seventh Conference on Neural Information Processing Systems*. arXiv:2303.17651. 2023, p. 2303.17651 (cited on pages 87, 91, 98).
- [MJS24] Zhao Mandi, Shreeya Jain, and Shuran Song. ‘Roco: Dialectic Multi-Robot Collaboration with Large Language Models’. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 286–299 (cited on page 100).
- [Mar+17] Jarryd Martin, S. Suraj Narayanan, Tom Everitt, and Marcus Hutter. ‘Count-Based Exploration in Feature Space for Reinforcement Learning’. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI’17. Melbourne, Australia: AAAI Press, 2017, pp. 2471–2478 (cited on pages 14, 49).
- [Mea82] Donald Meagher. ‘Geometric Modeling Using Octree Encoding’. In: *Computer Graphics and Image Processing* 19.2 (June 1982), pp. 129–147 (cited on page 10).
- [Met+10] Giorgio Metta, Lorenzo Natale, Francesco Nori, Giulio Sandini, David Vernon, Luciano Fadiga, Claes von Hofsten, Kerstin Rosander, Manuel Lopes, José Santos-Victor, et al. ‘The iCub Humanoid Robot: An Open-Systems Platform for Research in Cognitive Development’. In: *Social Cognition: from Babies to Robots* 23.8 (Oct. 2010), pp. 1125–1134 (cited on page 18).

- [Mia+23] Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. ‘Augmented Language Models: A Survey’. In: arXiv:2302.07842 (2023) (cited on pages 71, 73, 76).
- [Mie+21] Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, et al. ‘Between Words and Characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP’. In: arXiv:2112.10508 (Dec. 2021), p. 2112.10508 (cited on page 92).
- [Mik+13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. ‘Efficient Estimation of Word Representations in Vector Space’. In: arXiv:1301.3781 (Sept. 2013), p. 1301.3781 (cited on page 35).
- [Mir+21] Glareh Mir, Matthias Kerzel, Erik Strahl, and Stefan Wermter. ‘A Humanoid Robot Learning Audiovisual Classification by Active Exploration’. In: *2021 IEEE International Conference on Development and Learning*. 2021, pp. 1–6 (cited on page 56).
- [Mit+23] Mayank Mittal, Calvin Yu, Qinxu Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, et al. ‘Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments’. In: 8.6 (June 2023), p. 2301.04195 (cited on page 111).
- [Mni+15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. ‘Human-Level Control through Deep Reinforcement Learning’. In: *Nature* 518.7540 (Feb. 2015), pp. 529–533 (cited on page 59).
- [Moe+23] Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. *Model-Based Reinforcement Learning: A Survey*. 2023 (cited on pages 13, 53).
- [ME85] H. Moravec and A. Elfes. ‘High Resolution Maps from Wide Angle Sonar’. In: *1985 IEEE International Conference on Robotics and Automation Proceedings*. Vol. 2. Mar. 1985, pp. 116–121 (cited on page 10).
- [Mu+15] Beipeng Mu, Matthew Giamou, Liam Paull, Ali-akbar Aghamohammadi, John J. Leonard, and Jonathan P. How. ‘Information-Based Active SLAM via Topological Feature Graphs’. In: *2016 IEEE 55th Conference on Decision and Control (CDC)* (2015), pp. 5583–5590 (cited on pages 10, 12, 25).
- [Mue+25] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. ‘S1: Simple Test-Time Scaling’. In: arXiv:2501.19393 (Feb. 2025), p. 2501.19393 (cited on page 83).

- [Mur+18] Adithyavairavan Murali, Yin Li, Dhiraj Gandhi, and Abhinav Gupta. ‘Learning to Grasp without Seeing’. In: *International Symposium on Experimental Robotics*. Springer. 2018, pp. 375–386 (cited on page 58).
- [Mur22] Kevin P. Murphy. *Probabilistic Machine Learning: An Introduction*. MIT Press, 2022 (cited on pages 3, 32).
- [Nai+18] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and P. Abbeel. ‘Overcoming Exploration in Reinforcement Learning with Demonstrations’. In: *2018 IEEE International Conference on Robotics and Automation* (2018), pp. 6292–6299 (cited on page 58).
- [NHR99] Andrew Y Ng, Daishi Harada, and Stuart Russell. ‘Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping’. In: *Icml*. Vol. 99. 1999, pp. 278–287 (cited on page 159).
- [NVI25] NVIDIA. *NVIDIA Isaac Sim*. 2025 (cited on pages 9, 19, 21).
- [NVI+25] NVIDIA, Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, et al. ‘Cosmos World Foundation Model Platform for Physical AI’. In: arXiv:2501.03575 (Jan. 2025), p. 2501.03575 (cited on pages 34, 134).
- [Nye+21] Maxwell Nye, Michael Tessler, Josh Tenenbaum, and Brenden M Lake. ‘Improving Coherence and Consistency in Neural Sequence Models with Dual-System, Neuro-Symbolic Reasoning’. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, p. 2107.02794 (cited on page 88).
- [ONe+24] Abby O’Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. ‘Open X-embodiment: Robotic Learning Datasets and RT-X Models : Open X-embodiment Collaboration0’. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 6892–6903 (cited on pages 47, 134).
- [Ope24] OpenAI. *Learning to Reason with LLMs*. 2024 (cited on pages 83, 127).
- [OWN96] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Signals & Systems (2nd Ed.)* USA: Prentice-Hall, Inc., 1996 (cited on page 139).
- [Ouy+22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. ‘Training Language Models to Follow Instructions with Human Feedback’. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 27730–27744 (cited on pages 81, 83, 86, 121).
- [OE18] Andrew Owens and Alexei A Efros. ‘Audio-Visual Scene Analysis with Self-Supervised Multisensory Features’. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 631–648 (cited on page 58).
- [Owe+16] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. ‘Ambient Sound Provides Supervision for Visual Learning’. In: *European Conference on Computer Vision*. Springer. 2016, pp. 801–816 (cited on page 58).

- [Pan+25] Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng Ouyang, and Daniel Rueckert. ‘MedVLM-R1: Incentivizing Medical Reasoning Capability of Vision-Language Models (VLMs) via Reinforcement Learning’. In: arXiv:2502.19634 (Mar. 2025), p. 2502.19634 (cited on page 132).
- [PY10] Sinno Jialin Pan and Qiang Yang. ‘A Survey on Transfer Learning’. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359 (cited on page 60).
- [Par+18] German I. Parisi, Pablo Barros, Di Fu, Sven Magg, Haiyan Wu, Xun Liu, and Stefan Wermter. ‘A Neurorobotic Experiment for Crossmodal Conflict Resolution in Complex Environments’. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2018, pp. 2330–2335 (cited on page 56).
- [Par+19] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. ‘Continual Lifelong Learning with Neural Networks: A Review’. In: *Neural Networks* 113 (May 2019), pp. 54–71 (cited on page 16).
- [Par+23] Seohong Park, Kimin Lee, Youngwoon Lee, and Pieter Abbeel. ‘Controllability-Aware Unsupervised Skill Discovery’. In: *arXiv Preprint arXiv:2302.05103*. 2023, p. 2302.05103 (cited on pages 101, 102).
- [Pat+17] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. ‘Curiosity-Driven Exploration by Self-Supervised Prediction’. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2778–2787 (cited on pages 13, 14, 17, 47, 49, 58–61).
- [PGG19] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. ‘Self-Supervised Exploration via Disagreement’. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5062–5071 (cited on page 59).
- [Pat+19] Siddharth Patki, Andrea F. Daniele, Matthew R. Walter, and Thomas M. Howard. ‘Inferring Compact Representations for Efficient Natural Language Understanding of Robot Instructions’. In: arXiv:1903.09243 (Mar. 2019), p. 1903.09243 (cited on page 73).
- [Pen+22] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ‘ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters’. In: *ACM Trans. Graph.* 41.4 (July 2022) (cited on page 102).
- [PMA10] Jan Peters, Katharina Mulling, and Yasemin Altun. ‘Relative Entropy Policy Search’. In: *Twenty-Fourth AAAI Conference on Artificial Intelligence*. 2010 (cited on page 62).
- [Poo+19] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. ‘On Variational Bounds of Mutual Information’. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5171–5180 (cited on page 122).

- [Qia+24] Shuofei Qiao, Runnan Fang, Zhisong Qiu, Xiaobin Wang, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. ‘Benchmarking Agentic Workflow Generation’. In: arXiv:2410.07869 (Oct. 2024), p. 2410.07869 (cited on pages 47, 116).
- [Qui+09] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. ‘ROS: An Open-Source Robot Operating System’. In: *ICRA Workshop on Open Source Software*. Vol. 3. Kobe, 2009, p. 5 (cited on page 19).
- [Qwe+25] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. ‘Qwen2.5 Technical Report’. In: arXiv:2412.15115 (Jan. 2025), p. 2412.15115 (cited on page 131).
- [Rad+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. ‘Learning Transferable Visual Models from Natural Language Supervision’. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763 (cited on pages 29, 36).
- [Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. ‘Improving Language Understanding by Generative Pre-Training’. In: (2018) (cited on page 36).
- [Raf+23] 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: arXiv:2305.18290 (May 2023), p. 2305.18290 (cited on page 121).
- [Raj+21] Sai Rajeswar, Cyril Ibrahim, Nitin Surya, Florian Golemo, David Vazquez, Aaron Courville, and Pedro O. Pinheiro. ‘Haptics-Based Curiosity for Sparse-Reward Tasks’. In: *5th Annual Conference on Robot Learning*. 2021 (cited on pages 14, 59, 65).
- [Rak+21] Kate Rakelly, Abhishek Gupta, Carlos Florensa, and Sergey Levine. ‘Which Mutual-Information Representation Learning Objectives Are Sufficient for Control?’ In: *CoRR* abs/2106.07278 (2021) (cited on page 122).
- [Ran+23] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. ‘SayPlan: Grounding Large Language Models Using 3D Scene Graphs for Scalable Task Planning’. In: *7th Annual Conference on Robot Learning*. 2023 (cited on pages 25, 37, 44).
- [RMH21] Ian C Rankin, Seth McCammon, and Geoffrey A Hollinger. ‘Robotic Information Gathering Using Semantic Language Instructions’. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 4882–4888 (cited on page 72).
- [Ren+21] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. ‘A Survey of Deep Active Learning’. In: 54.9 (2021), p. 2009.00236 (cited on page 28).

- [Ren+22] Yi Ming Ren, Mohammed S. Alhajeri, Junwei Luo, Scarlett Chen, Fahim Abdullah, Zhe Wu, and Panagiotis D. Christofides. ‘A Tutorial Review of Neural Network Modeling Approaches for Model Predictive Control’. In: *Computers & Chemical Engineering* 165 (Sept. 2022), p. 107956 (cited on page 42).
- [Rho+25] Seungeun Rho, Laura Smith, Tianyu Li, Sergey Levine, Xue Bin Peng, and Sehoon Ha. ‘Language Guided Skill Discovery’. In: *The Thirteenth International Conference on Learning Representations*. 2025 (cited on pages 16, 38, 53, 102, 120, 132).
- [Ric22] Francisco Martín Rico. *A Concise Introduction to Robot Programming with ROS2*. 1st ed. Boca Raton: Chapman and Hall/CRC, Sept. 2022 (cited on page 19).
- [Rob25] Franka Robotics. *Panda Franka Emika Datasheet*. 2025 (cited on page 19).
- [RSF13] Eric Rohmer, Surya P.N. Singh, and Marc Freese. ‘CoppeliaSim (Formerly V-REP): A Versatile and Scalable Robot Simulation Framework’. In: *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*. 2013 (cited on pages 9, 19, 21).
- [Rud+19] Ognjen Rudovic, Meiru Zhang, Bjorn Schuller, and Rosalind Picard. ‘Multi-Modal Active Learning from Human Data: A Deep Reinforcement Learning Approach’. In: *2019 International Conference on Multimodal Interaction*. 2019, pp. 6–15 (cited on page 29).
- [RD00] Richard M. Ryan and Edward L. Deci. ‘Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions’. In: *Contemporary Educational Psychology* 25.1 (Jan. 2000), pp. 54–67 (cited on pages 47, 48).
- [RKD97] Richard M. Ryan, Julius Kuhl, and Edward L. Deci. ‘Nature and Autonomy: An Organizational View of Social and Neurobiological Aspects of Self-Regulation in Behavior and Development’. In: *Development and Psychopathology* 9 (1997), pp. 701–728 (cited on page 47).
- [Sar+22] Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. ‘Neuro-Symbolic Artificial Intelligence: Current Trends’. In: *AI Communications* 34.3 (2022), pp. 197–209 (cited on page 88).
- [Sau+22] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. ‘Self-Critiquing Models for Assisting Human Evaluators’. In: arXiv:2206.05802 (June 2022), p. 2206.05802 (cited on pages 91, 98).
- [Sch+23] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. ‘Toolformer: Language Models Can Teach Themselves to Use Tools’. In: *Thirty-Seventh Conference on Neural Information Processing Systems*. arXiv:2302.04761. 2023, p. 2302.04761 (cited on page 87).

- [Sch+25] Thomas Schmied, Jörg Bornschein, Jordi Grau-Moya, Markus Wulfmeier, and Razvan Pascanu. ‘LLMs Are Greedy Agents: Effects of RL Fine-Tuning on Decision-Making Abilities’. In: arXiv:2504.16078 (Apr. 2025), p. 2504.16078 (cited on page 31).
- [Sch+22] Tim Schneider, Boris Belousov, Georgia Chalvatzaki, Diego Romeres, Devesh K. Jha, and Jan Peters. ‘Active Exploration for Robotic Manipulation’. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2022, pp. 9355–9362 (cited on page 49).
- [Sch+17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. ‘Proximal Policy Optimization Algorithms’. In: abs/1707.06347 (2017), p. 1707.06347 (cited on page 111).
- [Sek+20] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. ‘Planning to Explore via Self-Supervised World Models’. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 8583–8592 (cited on page 58).
- [Seq24] Sequoia Capital. *What’s next for AI Agentic Workflows Ft. Andrew Ng of AI Fund*. Mar. 2024 (cited on pages 47, 116).
- [Sha+20] Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. ‘Dynamics-Aware Unsupervised Discovery of Skills’. In: *International Conference on Learning Representations*. 2020 (cited on pages 16, 101, 102, 121).
- [Sha+23] Yonadav Shavit, Sandhini Agarwal, Miles Brundage, Steven Adler, Cullen O’Keefe, Rosie Campbell, Teddy Lee, Pamela Mishkin, Tyna Eloundou, Alan Hickey, et al. *Practices for Governing Agentic AI Systems*. Tech. rep. 2023 (cited on pages 47, 116, 117).
- [She+25] Haozhan Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. *VLM-R1: A Stable and Generalizable R1-style Large Vision-Language Model*. 2025 (cited on page 132).
- [SMF22a] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. ‘CLIPort: What and Where Pathways for Robotic Manipulation’. In: *Proceedings of the 5th Conference on Robot Learning*. Ed. by Aleksandra Faust, David Hsu, and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, Nov. 2022, pp. 894–906 (cited on page 72).
- [SMF22b] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. ‘Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation’. In: *Proceedings of the 6th Conference on Robot Learning (CoRL)*. 2022 (cited on page 72).
- [SK16] Bruno Siciliano and Oussama Khatib, eds. *Springer Handbook of Robotics*. Springer Handbooks. Cham: Springer International Publishing, 2016 (cited on page 40).

- [Sil+16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. ‘Mastering the Game of Go with Deep Neural Networks and Tree Search’. In: *Nature* 529.7587 (Jan. 2016), pp. 484–489 (cited on page 42).
- [Sil+21] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. ‘Reward Is Enough’. In: *Artificial Intelligence* 299 (2021), p. 103535 (cited on pages 57, 58).
- [Sin+22] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Coua-iron, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. ‘Flava: A Foundational Language and Vision Alignment Model’. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, p. 2112.04482 (cited on page 37).
- [Sin+20] Avi Singh, Albert Yu, Jonathan Yang, Jesse Zhang, Aviral Kumar, and Sergey Levine. ‘COG: Connecting New Skills to Past Experience with Offline Reinforcement Learning’. In: *Conference on Robot Learning (CoRL)*. Oct. 2020, p. 2010.14500 (cited on page 82).
- [STT24] Ishika Singh, David Traum, and Jesse Thomason. ‘TwoStep: Multi-Agent Task Planning Using Classical Planners and Large Language Models’. In: arXiv:2403.17246 (Mar. 2024), p. 2403.17246 (cited on page 133).
- [Sne+24] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. ‘Scaling LLM Test-Time Compute Optimally Can Be More Effective than Scaling Model Parameters’. In: arXiv:2408.03314 (Aug. 2024), p. 2408.03314 (cited on page 83).
- [Sri+23] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. ‘Beyond the Imitation Game: Quantifying and Extrapolating the Capabilities of Language Models’. In: (2023), p. 2206.04615 (cited on page 92).
- [SB03] Cyrill Stachniss and Wolfram Burgard. ‘Exploring Unknown Environments with Mobile Robots Using Coverage Maps’. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence*. IJCAI’03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 1127–1132 (cited on page 9).
- [Sti+20] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. ‘Learning to Summarize with Human Feedback’. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 3008–3021 (cited on page 121).
- [Str+22] DJ Strouse, Kate Baumli, David Warde-Farley, Volodymyr Mnih, and Steven Stenberg Hansen. ‘Learning More Skills through Optimistic Exploration’. In: *International Conference on Learning Representations*. 2022 (cited on page 14).

- [Sun+24] Xiaowen Sun, Xufeng Zhao, Jae Hee Lee, Wenhao Lu, Matthias Kerzel, and Stefan Wermter. ‘Details Make a Difference: Object State-Sensitive Neurorobotic Task Planning’. In: *Artificial Neural Networks and Machine Learning (ICANN)*. Ed. by Michael Wand, Kristína Malinovská, Jürgen Schmidhuber, and Igor V. Tetko. Cham: Springer Nature Switzerland, 2024, pp. 261–275 (cited on pages [37](#), [124](#), [125](#), [134](#), [201](#)).
- [SB18] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018 (cited on pages [13](#), [14](#), [16](#), [39](#)).
- [SPS99] Richard S. Sutton, Doina Precup, and Satinder Singh. ‘Between Mdps and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning’. In: *Artificial Intelligence* 112.1 (1999), pp. 181–211 (cited on page [39](#)).
- [Tea25] Gemini Robotics Team. *Gemini Robotics: Bringing AI into the Physical World*. Tech. rep. Google DeepMind, Mar. 2025 (cited on pages [13](#), [47](#)).
- [TB96] Sebastian Thrun and Arno Bücken. ‘Integrating Grid-Based and Topological Maps for Mobile Robot Navigation’. In: *AAAI/IAAI, Vol. 2*. 1996 (cited on page [12](#)).
- [TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. ‘MuJoCo: A Physics Engine for Model-Based Control’. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Oct. 2012, pp. 5026–5033 (cited on pages [9](#), [22](#)).
- [Ton+22] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. ‘VideoMAE: Masked Autoencoders Are Data-Efficient Learners for Self-Supervised Video Pre-Training’. In: *arXiv:2203.12602* (Oct. 2022), p. 2203.12602 (cited on page [82](#)).
- [Tor+24] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. ‘Reconciling Reality through Simulation: A Real-to-Sim-to-Real Approach for Robust Manipulation’. In: *arXiv:2403.03949* (Nov. 2024), p. 2403.03949 (cited on page [117](#)).
- [Tou+23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. ‘LLaMA: Open and Efficient Foundation Language Models’. In: *arXiv:2302.13971* (Feb. 2023), p. 2302.13971 (cited on page [93](#)).
- [TCM19] James Traer, Maddie Cusimano, and Josh H McDermott. ‘A Perceptually Inspired Generative Model of Rigid-Body Contact Sounds’. In: *Digital Audio Effects (Dafx)*. Vol. 1. 2019, p. 3 (cited on pages [20](#), [63](#), [141](#)).
- [vLV18] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. ‘Representation Learning with Contrastive Predictive Coding’. In: *abs/1807.03748* (2018), p. 1807.03748 (cited on pages [13](#), [29](#)).

- [Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. ‘Attention Is All You Need’. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017 (cited on page 29).
- [VS07] Argiro Vatakis and Charles Spence. ‘Crossmodal Binding: Evaluating the “Unity Assumption” Using Audiovisual Speech Stimuli’. In: *Perception & Psychophysics* 69.5 (July 2007), pp. 744–756 (cited on page 56).
- [VR23] Tiago Veiga and Jennifer Renoux. ‘From Reactive to Active Sensing: A Survey on Information Gathering in Decision-Theoretic Planning’. In: *ACM Computing Surveys* 55.13s (July 2023) (cited on page 14).
- [Vij+25] Prasanna Vijayaraghavan, Jeffrey Frederic Queißer, Sergio Verduzco Flores, and Jun Tani. ‘Development of Compositionality through Interactive Learning of Language and Action of Robots’. In: *Science Robotics* 10.98 (Jan. 2025), eadp0751 (cited on page 134).
- [WKS21] Jennifer Wakulicz, He Kong, and Salah Sukkarieh. ‘Active Information Acquisition under Arbitrary Unknown Disturbances’. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 8429–8435 (cited on page 72).
- [WL21] Feng Wang and Hhuaping Liu. ‘Understanding the Behaviour of Contrastive Loss’. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2021, pp. 2495–2504 (cited on page 13).
- [Wan+24a] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. ‘Voyager: An Open-Ended Embodied Agent with Large Language Models’. In: (2024), p. 2305.16291 (cited on pages 50, 109, 112, 132, 134).
- [Wan+22] Hu Wang, Jianpeng Zhang, Yuanhong Chen, Congbo Ma, Jodie Avery, Louise Hull, and Gustavo Carneiro. ‘Uncertainty-Aware Multi-Modal Learning via Cross-Modal Random Network Prediction’. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVII*. Springer. 2022, pp. 200–217 (cited on page 73).
- [Wan+24b] Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. ‘Large Language Models for Robotics: Opportunities, Challenges, and Perspectives’. In: *Journal of Automation and Intelligence* (Dec. 2024) (cited on page 24).
- [Wan+23a] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. ‘A Survey on Large Language Model Based Autonomous Agents’. In: arXiv:2308.11432 (Sept. 2023), p. 2308.11432 (cited on page 9).

- [Wan+21] Tingting Wang, Xufeng Zhao, Qiujuan Lv, Bo Hu, and Degang Sun. ‘Density Weighted Diversity Based Query Strategy for Active Learning’. In: *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. May 2021, pp. 156–161 (cited on pages [15](#), [28](#), [29](#), [49](#)).
- [Wan+23b] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. ‘Self-Consistency Improves Chain of Thought Reasoning in Language Models’. In: *The Eleventh International Conference on Learning Representations*. 2023 (cited on page [86](#)).
- [Wan+24c] Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. ‘RL-VLM-F: Reinforcement Learning from Vision Language Foundation Model Feedback’. In: *Proceedings of the 41st International Conference on Machine Learning*. Vol. 235. ICML’24. Vienna, Austria: JMLR.org, July 2024, pp. 51484–51501 (cited on page [123](#)).
- [Wei+22a] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. ‘Chain-of-Thought Prompting Elicits Reasoning in Large Language Models’. In: *Advances in Neural Information Processing Systems*. May 2022 (cited on pages [53](#), [84](#), [86](#), [87](#), [89](#), [93](#), [98](#), [131](#)).
- [Wei+22b] Yake Wei, Di Hu, Yapeng Tian, and Xuelong Li. ‘Learning in Audio-Visual Context: A Review, Analysis, and New Perspective’. In: (2022), p. 2208.09579 (cited on page [72](#)).
- [WW80] Robert B Welch and David H Warren. ‘Immediate Perceptual Response to Intersensory Discrepancy.’ In: *Psychological bulletin* 88.3 (1980), p. 638 (cited on page [56](#)).
- [Wel+23] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. ‘Generating Sequences by Learning to Self-Correct’. In: *The Eleventh International Conference on Learning Representations*. arXiv:2211.00053. arXiv, 2023, p. 2211.00053 (cited on page [87](#)).
- [WS00a] Stefan Wermter and Ron Sun. ‘An Overview of Hybrid Neural Systems’. In: *Hybrid Neural Systems*. Ed. by Stefan Wermter and Ron Sun. Berlin, Heidelberg: Springer, 2000, pp. 1–13 (cited on page [88](#)).
- [WS00b] Stefan Wermter and Ron Sun. *Hybrid Neural Systems*. 1778. Springer Science & Business Media, 2000 (cited on page [88](#)).
- [WM97] David H. Wolpert and William G. Macready. ‘No Free Lunch Theorems for Optimization’. In: *IEEE Transactions on Evolutionary Computation* 1.1 (Apr. 1997), pp. 67–82 (cited on page [15](#)).
- [Wu+24] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. ‘Grit: A Generative Region-to-Text Transformer for Object Understanding’. In: *European Conference on Computer Vision*. Springer, 2024, pp. 207–224 (cited on page [125](#)).

- [Wu+23] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. ‘Tidybot: Personalized Robot Assistance with Large Language Models’. In: *Autonomous Robots* 47.8 (2023), pp. 1087–1102 (cited on page 100).
- [Wu+20] Yanmin Wu, Yunzhou Zhang, Delong Zhu, Xin Chen, S. Coleman, Wenkai Sun, Xinggang Hu, and Zhiqiang Deng. ‘Object SLAM-based Active Mapping and Robotic Grasping’. In: *2021 International Conference on 3D Vision (3DV)*. 2020, pp. 1372–1381 (cited on page 25).
- [Wu+25] Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. ‘When More Is Less: Understanding Chain-of-Thought Length in LLMs’. In: arXiv:2502.07266 (Feb. 2025), p. 2502.07266 (cited on page 84).
- [Xi+25] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. ‘The Rise and Potential of Large Language Model Based Agents: A Survey’. In: *Science China Information Sciences* 68.2 (Jan. 2025), p. 121101 (cited on pages 9, 24, 132).
- [Xio+24] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. ‘Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-constraint’. In: *Proceedings of the 41st International Conference on Machine Learning. ICML’24*. Vienna, Austria: JMLR.org, 2024 (cited on page 121).
- [Yam97] Brian Yamauchi. ‘A Frontier-Based Approach for Autonomous Exploration’. In: *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. ‘Towards New Computational Principles for Robotics and Automation’*. July 1997, pp. 146–151 (cited on pages 12, 49, 52).
- [Yan+24] Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. ‘RLCD: Reinforcement Learning from Contrastive Distillation for LM Alignment’. In: *The Twelfth International Conference on Learning Representations*. 2024 (cited on page 84).
- [Yan+25] Zhao Yang, Thomas M. Moerland, Mike Preuss, Aske Plaat, and Edward S. Hu. ‘Guiding Skill Discovery with Foundation Models’. In: Openreview (2025) (cited on pages 16, 120, 132).
- [Yao+23a] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. ‘Tree of Thoughts: Deliberate Problem Solving with Large Language Models’. In: *Thirty-Seventh Conference on Neural Information Processing Systems*. arXiv:2305.10601. 2023, p. 2305.10601 (cited on pages 86, 87).
- [Yao+23b] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ‘ReAct: Synergizing Reasoning and Acting in Language Models’. In: *The Eleventh International Conference on Learning Representations*. 2023 (cited on pages 85, 87).

- [YKF21] Denis Yarats, Ilya Kostrikov, and Rob Fergus. ‘Image Augmentation Is All You Need: Regularizing Deep Reinforcement Learning from Pixels’. In: *International Conference on Learning Representations*. 2021 (cited on page 16).
- [Yok+24] Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. ‘VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation’. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. May 2024, pp. 42–48 (cited on page 12).
- [You+22] Bang You, Oleg Arenz, Youping Chen, and Jan Peters. ‘Integrating Contrastive Learning with Dynamic Models for Reinforcement Learning from Images’. In: *Neurocomputing* 476 (Mar. 2022), pp. 102–114 (cited on page 13).
- [YZW23] Fei Yu, Hongbo Zhang, and Benyou Wang. ‘Nature Language Reasoning, a Survey’. In: (2023), p. 2303.14725 (cited on page 53).
- [Yu+23] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montse Gonzalez Arenas, Hao-Tien Lewis Chiang, Tom Erez, Leonard Hasenclever, Jan Humplik, et al. ‘Language to Rewards for Robotic Skill Synthesis’. In: (2023), p. 2306.08647 (cited on pages 103, 105).
- [Zen+23a] Andy Zeng, Maria Attarian, brian ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, et al. ‘Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language’. In: *International Conference on Learning Representations*. 2023 (cited on pages 72, 73, 76).
- [Zen+23b] Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S. Yu. ‘Large Language Models for Robotics: A Survey’. In: arXiv:2311.07226 (Nov. 2023), p. 2311.07226 (cited on pages 9, 24, 45).
- [Zha+25] Honggen Zhang, Xufeng Zhao, Igor Molybog, and June Zhang. ‘REAL: Response Embedding-Based Alignment for LLMs’. In: *IJCAI 2025 Workshop on Causal Learning RecSys* (Dec. 2025) (cited on pages 120, 122, 201).
- [Zha+23a] Jesse Zhang, Jiahui Zhang, Karl Pertsch, Ziyi Liu, Xiang Ren, Minsuk Chang, Shao-Hua Sun, and Joseph J. Lim. ‘Bootstrap Your Own Skills: Learning to Solve New Tasks with Large Language Model Guidance’. In: *7th Annual Conference on Robot Learning*. arXiv, Oct. 2023, p. 2310.10021 (cited on pages 17, 100–102).
- [Zha+24a] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. ‘AFlow: Automating Agentic Workflow Generation’. In: arXiv:2410.10762 (Oct. 2024), p. 2410.10762 (cited on page 116).

- [Zha+24b] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. ‘LLaMA-adapter: Efficient Fine-Tuning of Language Models with Zero-Init Attention’. In: *The Twelfth International Conference on Learning Representations*. arXiv, 2024, p. 2303.16199 (cited on page 113).
- [Zha+18] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. ‘The Sound of Pixels’. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 570–586 (cited on pages 56, 61).
- [Zha+23b] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. ‘A Survey of Large Language Models’. In: arXiv:2303.18223 (Sept. 2023), p. 2303.18223 (cited on pages 45, 81).
- [Zha+24c] Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, and Stefan Wermter. ‘Enhancing Zero-Shot Chain-of-Thought Reasoning in Large Language Models through Logic’. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue. Torino, Italia: ELRA and ICCL, May 2024, pp. 6144–6166 (cited on pages 52, 53, 200).
- [Zha+23c] Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. ‘Chat with the Environment: Interactive Multi-modal Perception Using Large Language Models’. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3590–3596 (cited on pages 10, 18, 27, 28, 44, 85, 100, 103, 110, 200).
- [Zha+22] Xufeng Zhao, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. ‘Impact Makes a Sound and Sound Makes an Impact: Sound Guides Representations and Explorations’. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2512–2518 (cited on pages 14, 26, 72, 75, 200).
- [ZWW24] Xufeng Zhao, Cornelius Weber, and Stefan Wermter. ‘Agentic Skill Discovery’. In: *8th Conference on Robot Learning (CoRL 2024) Workshop on Language and Robot Learning: Language as an Interface (LangRob)*, Munich, Germany. Aug. 2024 (cited on pages 22, 38, 50, 52, 53, 134, 200).
- [Zha+24d] Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. ‘Marco-OI: Towards Open Reasoning Models for Open-Ended Solutions’. In: arXiv:2411.14405 (Nov. 2024), p. 2411.14405 (cited on page 131).
- [Zhe+23] Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. ‘Progressive-Hint Prompting Improves Reasoning in Large Language Models’. In: arXiv:2304.09797 (May 2023), p. 2304.09797 (cited on pages 86, 87).

- [Zho+23] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. ‘Least-to-Most Prompting Enables Complex Reasoning in Large Language Models’. In: *The Eleventh International Conference on Learning Representations*. arXiv:2205.10625. 2023, p. 2205.10625 (cited on pages 86, 87, 89).
- [Zhu+18] Delong Zhu, Tingguang Li, Danny Ho, Chaoqun Wang, and Max Q.-H. Meng. ‘Deep Reinforcement Learning Supervised Autonomous Exploration in Office Environments’. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. May 2018, pp. 7548–7555 (cited on page 9).
- [Zhu+23] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. ‘MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models’. In: arXiv:2304.10592 (Apr. 2023), p. 2304.10592 (cited on page 82).
- [Zhu+21] Hao Zhu, Man-Di Luo, Rui Wang, Ai-Hua Zheng, and Ran He. ‘Deep Audio-Visual Learning: A Survey’. In: *International Journal of Automation and Computing* 18 (2021), pp. 351–376 (cited on page 72).
- [Zie+08] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. ‘Maximum Entropy Inverse Reinforcement Learning’. In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3. AAAI’08*. Chicago, Illinois: AAAI Press, 2008, pp. 1433–1438 (cited on page 44).
- [Zit+23] 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: *Proceedings of the 7th Conference on Robot Learning*. Ed. by Jie Tan, Marc Toussaint, and Kourosh Darvish. Vol. 229. Proceedings of Machine Learning Research. PMLR, Nov. 2023, p. 2307.15818 (cited on pages 47, 82).

LIST OF PUBLICATIONS

Related publications during my doctoral study are listed below.

As 1st / co-1st (*) author:

1. Xufeng Zhao, Cornelius Weber, and Stefan Wermter. ‘Agentic Skill Discovery’. In: *8th Conference on Robot Learning (CoRL 2024) Workshop on Language and Robot Learning: Language as an Interface (LangRob)*, Munich, Germany. Aug. 2024 [Under Review in *Robotics and Autonomous Systems* 2025.]
2. Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, and Stefan Wermter. ‘Enhancing Zero-Shot Chain-of-Thought Reasoning in Large Language Models through Logic’. In: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. Ed. by Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue. Torino, Italia: ELRA and ICCL, May 2024, pp. 6144-6166
3. Xufeng Zhao, Mengdi Li, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. ‘Chat with the Environment: Interactive Multimodal Perception Using Large Language Models’. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3590-3596
4. Mengdi Li*, Xufeng Zhao*, Jae Hee Lee, Cornelius Weber, and Stefan Wermter. ‘Internally Rewarded Reinforcement Learning’. In: *Proceedings of the 40th International Conference on Machine Learning (ICML)*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 2023, pp. 20556-20574
5. Xufeng Zhao, Cornelius Weber, Muhammad Burhan Hafez, and Stefan Wermter. ‘Impact Makes a Sound and Sound Makes an Impact: Sound Guides Representations and Explorations’. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2512-2518

As co-author:

1. Guanlue Li, Xufeng Zhao, Fang Wu, and Sören Laue. ‘Joint Design of Protein Surface and Backbone Using a Diffusion Bridge Model’. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025
2. Mengdi Li, Jiaye Lin, Xufeng Zhao, Wenhao Lu, Peilin Zhao, Stefan Wermter, and Di Wang. ‘Curriculum-RLAIF: Curriculum Alignment with Reinforcement Learning from AI Feedback’. In: arXiv:2505.20075 (May 2025) [Under Review.]
3. Kun Chu, Xufeng Zhao, Cornelius Weber, and Stefan Wermter. ‘LLM+MAP: Bimanual Robot Task Planning Using Large Language Models and Planning Domain Definition Language’. In: arXiv:2503.17309 (Mar. 2025) [Under Review]
4. Wenhao Lu, Xufeng Zhao, Josua Spisak, Jae Hee Lee, and Stefan Wermter. ‘Mental Modelling of Reinforcement Learning Agents by Language Models’. In: *Transactions on Machine Learning Research (TMLR)* (2025), pp. 2835–8856

5. Honggen Zhang, Xufeng Zhao, Igor Molybog, and June Zhang. 'REAL: Response Embedding-Based Alignment for LLMs'. In: *IJCAI 2025 Workshop on Causal Learning RecSys* (Dec. 2025)
6. Xiaowen Sun, Xufeng Zhao, Jae Hee Lee, Wenhao Lu, Matthias Kerzel, and Stefan Wermter. 'Details Make a Difference: Object State-Sensitive Neurorobotic Task Planning'. In: *Artificial Neural Networks and Machine Learning (ICANN)*. Ed. by Michael Wand, Kristína Malinovská, Jürgen Schmidhuber, and Igor V. Tetko. Cham: Springer Nature Switzerland, 2024. ISBN: 978-3-031-72341-4, pp. 261-275
7. Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, Wenhao Lu, and Stefan Wermter. 'Large Language Models for Orchestrating Bimanual Robots'. In: *The 2024 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. 2024
8. Wenhao Lu, Xufeng Zhao, Thilo Fryen, Jae Hee Lee, Mengdi Li, Sven Magg, and Stefan Wermter. 'Causal State Distillation for Explainable Reinforcement Learning'. In: *Proceedings of the Third Conference on Causal Learning and Reasoning (CLEaR)*. Ed. by Francesco Locatello and Vanessa Didelez. Vol. 236. Proceedings of Machine Learning Research. PMLR, Apr. 2024, pp. 106-142
9. Wenhao Lu, Xufeng Zhao, Sven Magg, Martin Gromniak, Mengdi Li, and Stefan Wermter. 'A Closer Look at Reward Decomposition for High-Level Robotic Explanations'. In: *2023 IEEE International Conference on Development and Learning (ICDL)*. 2023
10. Kun Chu, Xufeng Zhao, Cornelius Weber, Mengdi Li, and Stefan Wermter. 'Accelerating Reinforcement Learning of Robotic Manipulations via Feedback from Large Language Models'. In: *7th Conference on Robot Learning (CoRL 2023) Workshop on Bridging the Gap Between Cognitive Science and Robot Learning in the Real World: Progresses and New Directions (CRL_WS)*, Atlanta, Georgia USA. 2024

NOTATION

The next list describes several symbols that are used within the thesis.

$\mathbb{L}(\cdot)$ Information loss

$\mathcal{A}$ Action space

$\mathcal{D}$ Dataset

$\mathcal{J}(\cdot)$ Interest function

$\mathcal{J}(\cdot)$ Objective function

$\mathcal{L}\{\cdot\}$ Laplace transform

$\mathcal{L}^{-1}\{\cdot\}$ Inverse Laplace transform

$\mathcal{N}(\cdot)$ Normal distribution

$\mathcal{S}$ State space

$\mathcal{Z}$ Latent space

$\pi_{\text{ref}}(\cdot)$ Reference policy function

$\pi_{\theta}(\cdot)$ Policy function

τ Agent trajectory $\tau = (s_0, a_0, s_1, a_1, \dots, s_T)$

θ Neural network parameters

a Or a_t for the (current) action at time t

$I(\cdot)$ Information gain

$L(\cdot)$ Loss function

s Or s_t for the (current) state at time t

s' Or s_{t+1} for the (next) state at time $t + 1$

T Final time step

t Time step

z Latent variable, usually being used to denote a latent representation of the skill

GREEK LETTERS WITH PRONUNCIATIONS

Character	Name	Character	Name
α	alpha <i>AL-fuh</i>	β	beta <i>BAY-tuh</i>
γ, Γ	gamma <i>GAM-muh</i>	ϕ, Φ	phi <i>FEE, or FI (as in hi)</i>
δ, Δ	delta <i>DEL-tuh</i>	π, Π	pi <i>PIE</i>
ϵ	epsilon <i>EP-suh-lon</i>	θ, Θ	theta <i>THAY-tuh</i>
λ, Λ	lambda <i>LAM-duh</i>	ψ, Ψ	psi <i>SIGH, or PSIGH</i>
μ	mu <i>MEW</i>	ω, Ω	omega <i>oh-MAY-guh</i>
τ	tau <i>TOW (as in cow)</i>	σ, Σ	sigma <i>SIG-muh</i>

Greek letters that are used within the thesis with pronunciations

EIDESSTATTLICHE VERSICHERUNG

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Hamburg 14.10.2025
Ort, Datum

Buten
Unterschrift

ERKLÄRUNG ZUR VERÖFFENTLICHUNG

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

Hamburg
10.10.2025
Ort, Datum

Antony Zhou
Unterschrift

ACRONYMS

A

AI Artificial Intelligence. [iii](#), [9](#), [11](#), [15](#), [22–24](#), [35](#), [36](#), [46](#), [47](#), [60](#), [63](#), [83](#), [116](#), [118](#), [130](#), [131](#), [134](#)
ASD Agentic Skills Discovery. [xii](#), [xiv](#), [6](#), [50](#), [53](#), [99–101](#), [103](#), [104](#), [106](#), [109](#), [111–113](#), [115–117](#), [136](#), [158](#), [160–162](#), [164](#)
ASLAM Active Simultaneous Localization and Mapping. [10](#), [23](#), [52](#)

B

BC Behavior Cloning. [43](#)

C

CLIP Contrastive Language-Image Pretraining. [xii](#), [36](#)
CNN Convolutional Neural Network. [63](#)
CoT Chain-of-Thought. [xii](#), [xiv](#), [6](#), [84](#), [86–88](#), [90–94](#), [96–98](#), [136](#), [155](#), [156](#)

D

DDPG Deep Deterministic Policy Gradient. [60](#), [64–67](#)
DDPM Denoising Diffusion Probabilistic Model. [44](#)
DoF Degrees of Freedom. [18](#), [19](#), [63](#), [104](#), [111](#), [118](#)
DPO Direct Preference Optimization. [121](#)

E

ELBO Evidence Lower Bound. [33](#)

H

HRL Hierarchical Reinforcement Learning. [39](#), [109](#), [135](#)

I

ICM Intrinsic Curiosity Module. [58–67](#)
IDE Integrated Development Environment. [52](#)
IL Imitation Learning. [43](#)
IRL Inverse Reinforcement Learning. [43](#), [44](#)
IRRL Internally Rewarded Reinforcement Learning. [xiii](#), [83](#), [120](#), [122](#)
ISCM Intrinsic Sound Curiosity Module. [xii](#), [xiv](#), [5](#), [6](#), [55](#), [56](#), [59–64](#), [66–68](#), [136](#)

L

LABOR Language-model-based Bimanual ORchestration. [xiii](#), [126](#)
Lafite-RL Language agent feedback interactive Reinforcement Learning. [xiii](#), [123](#), [124](#)
LLM Large Language Model. [iii](#), [iv](#), [xiv](#), [5–8](#), [12](#), [15](#), [17](#), [22–24](#), [27](#), [31](#), [34](#), [39](#), [44–46](#), [50–52](#), [69–113](#), [115](#), [116](#), [118–120](#), [122–134](#), [136](#), [144](#), [145](#), [147](#), [157–159](#), [161](#), [162](#), [164](#), [165](#), [167](#)
LLM+MAP LLM + Multi-Agent Planning with PDDL. [xiii](#), [126](#), [127](#)
LoT Logical Thoughts. [xii–xiv](#), [6](#), [84–98](#), [136](#), [144–147](#), [155–157](#)

M

Matcha Multimodal environment chatting. xii, xiv, 6, 69, 72–74, 76–80, 82, 124, 136

MCTS Monte Carlo Tree Search. 42

MDP Markov Decision Process. 16, 29, 38, 52, 59, 65

MPC Model Predictive Control. 42

N

NFL No Free Lunch Theorem. 15

NICOL Neuro-Inspired CoLLaborative semi-human robot. xii, 18–21, 77, 126

O

ORM Outcome-supervised Reward Model. 51, 124

OSSA Object-State Sensitive Agent. xiii, 125

P

PDDL Planning Domain Definition Language. xii, 39, 40, 46, 119, 124, 126, 127

PDF Probability Density Function. 138

PPO Proximal Policy Optimization. 111

PRM Process-supervised Reward Model. 51, 124

R

RAG Retrieval Augmented Generation. 101, 109, 112, 114–116

RL Reinforcement Learning. xiii, 5–8, 13, 14, 17, 20–23, 25, 38, 39, 41–43, 46, 48, 49, 51, 52, 55, 57–60, 62, 65, 68, 69, 82–84, 100–107, 109–114, 116, 118–124, 127–134, 136, 158, 159, 161, 162

RLAIF Reinforcement Learning from AI Feedback. 46, 83, 98, 120, 121, 123

RLHF Reinforcement Learning from Human Feedback. 46, 83, 120, 121, 132

ROS Robot Operating System. 22

ROS2 Robot Operating System 2. 22

RRT Rapidly-exploring Random Trees. 11

RT-1 Robotic Transformer. 47

RT-2 Robotic Transformer 2. 47

S

SDT Self-Determination Theory. 47, 48

SLAM Simultaneous Localization and Mapping. 10, 11, 23

SRN Simple Recurrent Network. 35

STFT Short-Time Fourier Transform. 63, 64

T

TAMP Task and motion planning. 40, 41

U

URL Unsupervised Reinforcement Learning. 17, 28, 55, 57, 65, 99, 102, 112, 132

URLB Unsupervised Reinforcement Learning Benchmark. 63, 65

V

VAE Variational Autoencoder. 13

VLA Vision-Language-Action Model. 13, 35, 47, 134, 135

VLM Vision Language Model. 46, 51, 52, 69, 99–101, 103, 106–108, 111, 113, 116, 123, 125, 134, 158

X

XAI Explainable AI. 128