Universidade Estadual de Campinas
Instituto de Computação

Alana de Santana Correia

DreamerRL: Empowering Representation Learning via Predictive World Models for Robot Manipulation Tasks

DreamerRL: Potencializando o Aprendizado de Representações por meio de Modelos Preditivos de Mundo para Tarefas de Manipulação Robótica

CAMPINAS
2025

Alana de Santana Correia

DreamerRL: Empowering Representation Learning via Predictive World Models for Robot Manipulation Tasks

DreamerRL: Potencializando o Aprendizado de Representações por meio de Modelos Preditivos de Mundo para Tarefas de Manipulação Robótica

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutora em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientadora: Prof. Esther Luna Colombini, Ph.D
Co-supervisor/Coorientadora: Prof. Paula Dornhofer Paro Costa, Ph.D

Este exemplar corresponde à versão final da Tese defendida por Alana de Santana Correia e orientada pela Prof. Esther Luna Colombini, Ph.D.

CAMPINAS
2025

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

Correia, Alana de Santana, 1993-
C817d DreamerRL : empowering representation learning via predictive world models for robot manipulation tasks / Alana de Santana Correia. – Campinas, SP : [s.n.], 2025.

Orientador: Esther Luna Colombini.
Coorientador: Paula Dornhofer Paro Costa.
Tese (doutorado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Computação.

1. Robótica. 2. Aprendizado por reforço profundo. 3. Motivação intrínseca. 4. Teoria de modelos de mundo. I. Colombini, Esther Luna, 1980-. II. Costa, Paula Dornhofer Paro, 1978-. III. Universidade Estadual de Campinas (UNICAMP). Instituto de Computação. IV. Título.

Informações complementares

Título em outro idioma: DreamerRL : potencializando o aprendizado de representações por meio de modelos preditivos de mundo para tarefas de manipulação robótica
Palavras-chave em inglês:
Robotics
Deep reinforcement learning
Intrinsic motivation
Theory of world models
Área de concentração: Ciência da Computação
Titulação: Doutora em Ciência da Computação
Banca examinadora:
Esther Luna Colombini [Orientador]
Ana Carolina Lorena
Anna Helena Reali Costa
Sandra Eliza Fontes de Avila
Hélio Pedrini
Data de defesa: 13-03-2025
Programa de Pós-Graduação: Ciência da Computação

Objetivos de Desenvolvimento Sustentável (ODS)
ODS: 9. Inovação e infraestrutura

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-7417-3727
- Currículo Lattes do autor: http://lattes.cnpq.br/8715294802755225

- Profa. Dra. Esther Luna Colombini
  IC/UNICAMP
- Profa. Dra. Ana Carolina Lorena
  ITA
- Profa. Dra. Anna Helena Reali Costa
  PCS/USP
- Profa. Dra. Sandra Eliza Fontes de Avila
  IC/UNICAMP
- Prof. Dr. Hélio Pedrini
  IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

The world is my representation.
(Arthur Schopenhauer)

Acknowledgements

First, I praise the Lord, the source of all that exists, has existed, and will exist, and the source of all knowledge, whose vastness and kindness have allowed for my existence and everything that surrounds me. I am deeply grateful for the gift of life and the opportunity to experience this earthly journey. I am thankful for the strength and health granted to me, enabling me to turn my dreams into reality, and for the knowledge that has guided me in completing this work.

I thank my parents for their love and for always being by my side throughout my life. To my mother, Ana Andrade de Santana Correia, my sincere appreciation for valuing education and striving to provide me access to quality learning. Without your dedication and efforts, I would not have reached this point. To my father, José Adelson Souza Correia, for his unwavering commitment to helping me achieve my dreams, always doing so with great care and love.

I thank my boyfriend, Iury Cleveston, for his love, companionship, and unwavering support throughout this journey. Our conversations and debates shaped this work in ways I could never have imagined. Your presence made graduate school easier, especially while I was far from home and friends. I am deeply grateful for your belief in me, even in moments of self-doubt. I love you.

I thank my godparents, Elisangela Matos da Cruz Celestino and Valdicelmo Santos Celestino, for being by me since childhood. You made my days happier and more enjoyable. I am grateful for your care, friendship, and support in helping me navigate life's challenges. You are much more than godparents to me, and I will always carry deep gratitude for these moments.

I thank my advisor, Prof. Esther Luna Colombini, for sharing her knowledge with me, for her trust, understanding, and support, and for believing in my potential from the first day we met. I also extend my gratitude to my co-advisor, Prof. Paula Dornhofer Paro Costa, for always bringing questions that encouraged me to reflect on the value of my work. These reflections have helped me grow as a researcher and improve this project and future ones.

I thank the University of Campinas professors for sharing their knowledge with me over the past few years. Also, I thank everyone who crossed my everyday path, offering valuable learning experiences and contributing directly and indirectly to completing this work.

Finally, I thank CAPES, Quinto Andar, and H.IIAC for their financial support throughout the development of this work. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. This project was supported by the Brazilian Ministry of Science, Technology and Innovations, with resources from Law nº 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published Apredizado em Arquiteturas Cognitivas (Phase 3), DOU 01245.003479/2024-10.

Resumo

Construir agentes robóticos complexos capazes de desenvolver habilidades motoras de forma autônoma, a partir da própria experiência sensorial e sem depender de instruções explícitas ou recompensas pré-definidas, continua sendo um problema em aberto na robótica. Embora o aprendizado supervisionado e o aprendizado por reforço tenham avançado, ainda enfrentam limitações, como a dependência de grandes volumes de dados rotulados ou de funções de recompensa projetadas manualmente. Além disso, agentes treinados de forma tradicional tendem a desenvolver políticas altamente especializadas, dificultando sua generalização e adaptação a novos cenários. Nesse contexto, propomos o **DreamerRL**, um *framework* voltado à criação de agentes humanoides mais adaptáveis e autônomos. Inspirado por teorias de modelos de mundo, o DreamerRL não otimiza políticas específicas para tarefas isoladas, mas promove o aprendizado de um modelo do mundo. Ao aprender a prever como o mundo funciona, o agente desenvolve políticas versáteis, capazes de sustentar o desenvolvimento de habilidades motoras em resposta à exploração de estados que ainda não consegue prever. Em nosso *framework*, demonstramos a importância da personificação, das estruturas neurais inspiradas no circuito neocortical e da motivação intrínseca no aprendizado do modelo de mundo. Nossos experimentos com o robô humanoide NAO evidenciam que esses elementos, quando combinados em um sistema unificado para aprender um modelo de mundo, favorecem o surgimento espontâneo de comportamentos motores complexos e a transferência eficaz de habilidades para novas tarefas. Inicialmente, o agente foi treinado apenas para prever a próxima observação visual do ambiente, recebendo recompensas de motivação intrínseca que o instigavam a buscar estados novos e desafiadores. Progressivamente, liberamos modalidades sensoriais adicionais para previsão, ativamos a mobilidade do pescoço e incorporamos a curiosidade multimodal como forma de recompensa intrínseca, ampliando a riqueza da interação sensório-motora e a complexidade das previsões sobre o mundo. Essa evolução possibilitou o surgimento autônomo de habilidades motoras e cognitivas sofisticadas, semelhantes às observadas no desenvolvimento infantil — como a melhora da destreza e da precisão dos movimentos das mãos, ações manipulativas mais complexas (como arrastar e levantar objetos), além do desenvolvimento da atenção visual e da sinergia sensório-motora voltada à satisfação de objetivos internos. Posteriormente, o agente foi capaz de transferir essas habilidades para uma tarefa específica e desafiadora, sem a necessidade de retreinamento substancial. O desempenho superior do nosso agente em relação a um agente clássico treinado via aprendizado por reforço confirma a eficácia do nosso *framework* em promover representações internas generalizáveis e adaptativas, oferecendo avanços para o desenvolvimento de agentes robóticos complexos verdadeiramente autônomos.

Abstract

Building complex robotic agents capable of autonomously developing motor skills from their own sensory experience without relying on explicit instructions or predefined rewards remains an open challenge in robotics. While supervised learning and reinforcement learning have achieved significant progress, they still face limitations, such as dependence on large amounts of labeled data or manually designed reward functions. Moreover, traditionally trained agents tend to develop highly specialized policies, which hinders their ability to generalize and adapt to new scenarios. In this context, we propose **DreamerRL**, a framework to enable more adaptable and autonomous humanoid agents. Inspired by world model theories, DreamerRL does not optimize task-specific policies; instead, it promotes learning a predictive model of the world. By learning to anticipate how the world works, the agent acquires versatile policies that support the progressive development of motor skills by exploring states it has not yet learned to predict. In our framework, we highlight the importance of embodiment, neural structures inspired by the neocortical circuit, and intrinsic motivation in the learning of world models. Our experiments with the NAO humanoid robot demonstrate that when combined in a unified system to predict the world, these elements facilitate the spontaneous emergence of complex motor behaviors and the effective transfer of skills to novel tasks. Initially, the agent was trained solely to predict the next visual observation of the environment, receiving intrinsic motivation rewards that encouraged the exploration of novel and challenging states. Over time, we progressively enabled additional sensory modalities for prediction, activated neck mobility, and incorporated multimodal curiosity as an intrinsic reward signal, enriching the agent's sensorimotor experience and increasing the complexity of its predictive model. This gradual evolution led to the autonomous emergence of sophisticated motor and cognitive skills resembling those observed in children's development — such as improvements in hand dexterity and precision, more complex object manipulation (e.g., dragging and lifting), the development of sustained visual attention and sensorimotor synergy to satisfy internal goals. Eventually, the agent could transfer these previously acquired skills to a specific, challenging downstream task without requiring substantial retraining. The agent's superior performance compared to a classical reinforcement learning agent confirms the effectiveness of our framework in promoting generalizable and adaptive internal representations, offering progress toward the development of truly autonomous robotic agents.

List of Figures

List of Tables

List of Abbreviations and Acronyms

BRIMs	Bidirectional Recurrent Independent Mechanisms
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
HTM	Hierarchical Temporal Memory
IM	Intrinsic Motivation
KL	Kullback–Leibler
LSTM	Long Short-Term Memory
MC	Monte Carlo
MDP	Markov Decision Process
OOD	Out-of-Distribution
PPO	Proximal Policy Optimization
RIMs	Recurrent Independent Mechanisms
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
SDR	Sparse Distributed Representations
TD	Temporal Difference
TRPO	Trust Region Policy Updates

Contents

Chapter 1

Introduction

Building complex robotic agents capable of autonomously developing motor skills from their own sensory experience without relying on explicit instructions or predefined rewards remains an open challenge in robotics. In the last twenty years, most of these robots have been trained using supervised learning protocols, which rely on vast amounts of labeled data to learn specific tasks [5]. However, supervised learning has notable limitations due to its dependence on large and accurately labeled datasets. Creating these datasets is labor-intensive and time-consuming, requiring expert knowledge, particularly in specialized domains. The vast diversity and complexity of real-world environments exacerbate this challenge, making it impractical and often infeasible to gather labeled data that covers all possible scenarios and tasks.

An alternative to the lack of labeled data is reinforcement learning (RL) [73], where an agent learns by interacting with the environment through trial and error to maximize a reward function that defines the desired behavior in that environment. This training protocol allows the agent to independently discover optimal strategies for achieving its objectives without needing a large volume of labeled data. However, this approach also has its limitations. For an agent to learn even a simple task, it must interact with the environment through an extremely high number of training steps, which is computationally expensive and time-consuming. Furthermore, the RL process is highly dependent on the design of the reward function, which must be carefully crafted to ensure the desired behavior is learned. Adapting this knowledge to different environments or slightly altered conditions is challenging once the agent learns to perform a task in a specific environment. This difficulty arises because the agent tends to develop highly specialized action policies tailored to the conditions of the original training, making it challenging to transfer training knowledge to new contexts [144].

In this scenario, transfer learning techniques have been proposed to mitigate issues in the adaptability of reinforcement learning [131]. This technique involves using representations that have been pre-trained in a specific context and applying them to a new task context to facilitate and accelerate the RL agent's learning process. While transfer learning is currently a promising approach in supervised learning, many challenges in RL still need to be overcome to provide better adaptation and generalization. The primary issue lies in the domain discrepancy between the pre-training context and the new task domain. The pre-trained model may have learned characteristics and patterns that are irrelevant or

unsuitable for the new environment. For instance, a model pre-trained to identify objects in static images of indoor environments may not transfer well to an outdoor exploration setting when used by a robotic agent. This difficulty arises due to differences in object types, lighting, and scenery, as well as the fact that these representations were trained without considering the sensory and motor aspects of the robot.

These approaches often produce highly inflexible representations that cannot be easily applied to new situations. This inflexibility arises mainly because the representations are developed during training to be highly specific to the task [62]. As a result, although agents may achieve superhuman performance in certain activities, they fail to adapt quickly to new situations or tasks. This limitation stems from the fact that these representations are not designed to understand the fundamental principles of how the world works but rather to be effective in performing a specific task. In contrast, humans spend a significant portion of their lives, especially during childhood, exploring the world and learning how it works, developing a broader and more flexible understanding that enables them to adapt to a wide range of new and unforeseen situations [109].

Numerous theories aim to explain how humans learn to represent the world in a flexible and broader way [110, 149, 147, 9]. Several research supported by psychological [20], neuroscientific [55], and biological evidence [40] argue that our brain is predictive and builds an internal model of the world as we explore the environment. This research emphasizes that we learn to represent the external world from the dynamic interaction between the body and the environment, and our knowledge is shaped by the body's physical structure, sensory capabilities, and how it moves through and manipulates its surroundings. We are embodied agents with intricate sensorimotor integration, where sensory inputs and motor actions are continuously coordinated and interconnected. This integration allows the body to interpret sensory stimuli, adjust actions, and adapt to new contexts, forming the basis for developing a coherent internal representation of the world. Together, the predictive nature of the brain and sensorimotor integration drive the lifelong construction and refinement of our world model, often guided by intrinsic and task-independent goals fueled by our desire to understand how the world works. As we acquire new information and interact with the environment, we continuously build and refine our internal world model to represent the structure and dynamics of the surrounding world. Throughout life, this model is constantly updated based on our experiences and perceptions, enabling us to anticipate events, make informed decisions, and adjust our behavior according to changing circumstances [84, 55, 85, 117].

The flexibility of our world model is enabled by the highly adaptable structure of the neocortex, which is organized hierarchically, modularly, and sparsely. This organization allows information to be processed at varying levels of complexity, from basic sensory features to the most abstract and sophisticated concepts. Within the neocortical structure, groups of neurons specialize in processing specific sensory or cognitive information types, forming semi-independent modules that interact through bottom-up and top-down connections. As new sensory stimuli are received, only a small fraction of neurons is activated to represent the incoming information, reflecting the efficiency and selectivity of our neural architecture. Moreover, this structure optimizes energy costs and neural processing and enhances the brain's capacity for generalization and adaptation to novel

challenges and environments [56].

In this context, we propose **DreamerRL**, a framework designed to advance the development of truly autonomous and adaptable humanoid agents. Addressing a central limitation in contemporary robotics — the reliance on learning task-specific policies and handcrafted rewards — DreamerRL draws inspiration from world model theories and centers the learning process on constructing a predictive internal model of the environment. By learning to anticipate how the world works, the agent is intrinsically driven to explore unfamiliar regions of the environment, progressively acquiring a repertoire of general-purpose motor strategies. Our approach emphasizes the critical role of three foundational elements often overlooked in prior work and essential for building effective world models: intrinsic motivation as a mechanism for self-directed exploration, neural structures inspired by neocortical circuits to support hierarchical knowledge organization, and embodiment as a key factor in shaping perceptual input and grounding cognition. We demonstrate that integrating these components into a unified predictive world model system leads to the autonomous emergence of complex motor and cognitive behaviors and robust skill transfer to novel tasks, offering a promising path toward developing truly autonomous robots.

To implement the embodiment, we utilize the humanoid robot NAO, which possesses sensors and a body structure similar to humans [132]. We employ intrinsically motivated rewards in reinforcement learning to guide the learning process. We chose curiosity among various intrinsic reward mechanisms, as it drives exploratory behavior, facilitating world discovery, which is a trait commonly observed in children during early cognitive development [110, 109]. Finally, we introduce neocortex-inspired structures, such as artificial neural networks, modularity, sparsity, and hierarchical biases. This design supports the creation of an efficient and adaptive representation structure, mirroring key functional aspects of the human neocortex [55]. Initially, all elements are introduced in a primitive way and are enriched during work development.

We selected an object manipulation environment to validate our framework because of its complexity and relevance to evaluating our approach. This environment presents significant learning and adaptation challenges, as manipulating objects with a complex robot like NAO requires executing precise, intricate, and highly coordinated movements involving multiple joints simultaneously; such a skill is challenging to acquire solely through supervised learning or even traditional reinforcement learning setups. In this environment, we aim to validate whether our approach maximizes autonomous development and adaptation challenges in complex robotic agents. Our framework, based on building a world model, can address this limitation by enabling a complex robot to develop autonomously during training, thereby acquiring task-independent skills that are inherently more adaptive. We hypothesize that how the robot constructs its internal representation will promote greater flexibility and adaptability, empowering the agent to handle new situations and tasks without requiring extensive retraining.

1.1 Objectives

The main objective of this work is to develop the **DreamerRL** framework based on the core elements of world model theories, such as embodiment, a neocortex-like structure, and self-supervision guided by intrinsic motivation. Also, we will investigate how the proposed framework contributes to the autonomous development and adaptability of a complex humanoid robotic agent.

To achieve the main objective, the following specific objectives are proposed:

1. Conducting a literature review on neocortical circuit theories, world models, reinforcement learning, embodied cognition, intrinsic motivation methods, representation learning for robots, and the latest advances in Deep Learning models;
2. Implementing the world model theories elements for autonomous development and adaptability of the robotic agent. The elements include embodied cognition, along with cognitive biases of sparsity, modularity, hierarchy, and intrinsic motivation;
3. Discussing autonomous robot development as these elements become increasingly enriched and complex;
4. Evaluating the quality of the DreamerRL results regarding the robot's adaptation to new tasks and environments;
5. Discussing and comparing our task adaptation results with extrinsic RL agents.

1.2 Hypotheses

Based on the specified objectives, the following research hypotheses are raised:

1. $H1$: A complex robotic agent, trained to model the world in an object manipulation environment, can accurately predict both the dynamics of the external environment and its behavior;
2. $H2$: A complex robotic agent autonomously develops structured object manipulation behaviors driven solely by the motivation to predict the world;
3. $H3$: Increasing embodiment enables the agent's complete immersion in the environment, promoting the autonomous development of more complex object manipulation skills;
4. $H4$: The world model learned through sensorimotor experiences enables the robotic agent to learn abstract concepts about how the world functions, allowing it to imagine and simulate novel situations not encountered during training;
5. $H5$: The behaviors learned during world exploration are task-independent, making the agent more adaptive and capable of quickly applying the acquired exploration skills to accelerate the adaptation to a new extrinsic task;
6. $H6$: Incorporating sparsity, modularity, and hierarchical biases enhances intrinsic policy adaptation to a new extrinsic task.

1.3 Contributions

The main contributions of this work are:

1. Development of an approach that enables a complex humanoid robot to autonomously acquire motor skills for use in object manipulation environments that require advanced motor coordination abilities;

2. Design of a multimodal curiosity-driven embodied agent capable of imagining and generalizing how the world works;

3. Implementation of intrinsic curiosity policies that improve humanoid robots' autonomous development and task adaptation;

4. Empirical analysis of cognitive biases in intrinsic learning, demonstrating that attentional sparsity, modular network structures, and hierarchical processing improvement task adaptation;

5. Development of a methodology for assessing the autonomous developmental trajectory of a complex humanoid agent, drawing parallels with stages of human infant development;

6. Development of an approach to improve the flexibility and adaptability of complex robotic agents trained through reinforcement learning;

7. Publication of two surveys on attention mechanisms in Deep Learning, covering architectural designs, cognitive inspirations, and applications in vision, language, and multimodal domains.

1.4 Publications

The following papers were directly produced in the context of this doctoral thesis:

1. Santana, A., & Colombini, E. L. (2022). Attention, please! A survey of neural attention models in deep learning. Artificial Intelligence Review, 55(8), 6037-6124. Impact factor: 12.0.

2. Santana, A., & Colombini, E. (2021). Neural attention models in deep learning: Survey and taxonomy. arXiv preprint arXiv:2112.05909.

3. Santana, A., Costa, P. P., & Colombini, E. L. (2025). Learning To Explore With Predictive World Model Via Self-Supervised Learning. arXiv preprint arXiv:2502.13200.

The following papers were produced in collaboration and are related to the subjects studied during the development of this thesis:

1. Santana, A., Cleveston, I., dos Santos, V. B., Avila, S., & Colombini, E. L. (2021). An attentional model for earthquake prediction using seismic data. In Highlights in Practical Applications of Agents, Multi-Agent Systems, and Social Good. The PAAMS Collection: International Workshops of PAAMS 2021, Salamanca, Spain, October 6–9, 2021, Proceedings 19 (pp. 53-64). Springer International Publishing.

2. Cleveston, I., Santana, A. C., Costa, P. D., Gudwin, R. R., Simões, A. S., & Colombini, E. L. (2025). InstructRobot: A Model-Free Framework for Mapping Natural Language Instructions into Robot Motion. arXiv preprint arXiv:2502.12861.

1.5 Text Structure

This thesis is structured in six chapters to provide the reader with the necessary fundaments for understanding the proposed work and the results achieved.

Chapter 1 introduced the problem we intend to solve and indicated our motivations and objectives with this work. Then, hypotheses were raised to guide our investigation. We also presented the contributions this work generated to the scientific community.

In Chapter 2, we will address the theoretical basis for our work, such as the world model theories, embodied cognition, neocortical circuit, and intrinsic motivation learning. This knowledge will be the foundation for comprehending this work and its results.

Chapter 3 presents the literature review, which includes other works related to our research. We analyze works in frameworks and cognitive architectures for world models, intrinsic motivation for learning agents, and task adaptation in robot manipulation.

Chapter 4 contains the materials used to construct our proposed model, such as environments for training models, metrics for evaluation, and software and hardware technologies. We will also describe in detail the development methodology.

Chapter 5 details the DreamerRL framework's construction, the experiments' discussion, and the results achieved.

Finally, Chapter 6 concludes this thesis by summarizing our results, benefits for the community, and future works.

Chapter 2

Theoretical Background

This chapter presents the theoretical background necessary to understand our work. Section 2.1 provides a theoretical overview of world models theory. The following sections delve deeper into each pillar of the theory used in our work to build the DreamerRL framework and the computational tools we used to implement these pillars. Specifically, Section 2.2 offers an overview of the main theories on embodied cognition and their relevance in constructing human world representation. Section 2.3 discusses neocortical circuit theories, detailing how these structures contribute to information processing. Finally, Section 2.4 explores the role of intrinsic motivation, focusing on curiosity and how it drives the agent's continuous and adaptive learning. Finally, Section 2.5 synthesizes all the elements and describes how they are linked for constructing our agent.

2.1 World Model Theories

Theories concerning world models trace back to the epoch of Immanuel Kant, whose transcendental philosophy initiated some of the earliest inquiries into the fundamental assumptions underlying how we come to know the world. He sought to balance empiricism (which holds that all knowledge originates from sensory experience) and rationalism (which asserts that reason alone can produce all valid knowledge). According to Kant [76], neither empiricism nor rationalism alone adequately explains how knowledge is formed in the human mind. For him, there is no knowledge without experience. However, experience is never neutral, as it is shaped by the a priori forms of sensibility and understanding, which are innate characteristics of human cognition. In other words, knowledge emerges from the interaction between the subject and the object. However, it is structured by these innate frameworks, which make experience possible and infuse perception with meaning.

For Kant, space and time are the fundamental forms of sensibility, existing independently of sensory experience. They function solely within the human mind as "frameworks" that allow us to perceive phenomena. Meanwhile, the categories of understanding are a priori concepts — innate structures of thought — that enable the mind to organize and interpret sensory data even before any actual experience takes place. Kant identifies twelve categories of understating distributed between four main groups: quantity, quality, relation, and modality. The quantity addresses the numerical aspects of experience,

including the categories of unity (singular instances), plurality (multiple instances), and totality (a complete set or whole). The quality is focused on the nature of experiences; this group includes the categories of reality (what exists), negation (what does not exist), and limitation (the boundaries of existence). The relation concerned with how elements of experience are connected; this group comprises the categories of substance and accident (the inherent and dependent properties of objects), causality (cause and effect), and community (interaction between substances). Finally, modality addresses the status of experiences in terms of possibility (what could be), existence (what is), and necessity (what must be).

Sensibility forms and categories of understanding work together to structure the world in our minds as we know it. For Kant, we structure the world always in the phenomena domain, which is the reality as it appears seen by our senses and is organized by our mind, but never access the noumenon (i.e., "the thing itself"), which is the world how it is, because we not resources for this. Kant's vision is that we cannot access the real world; we only project it with our senses. Posteriorly, some philosophers who disagree with this vision argue that the real world does not exist and that the world exists only through observation, and innate categories do not exist, asserting instead that all knowledge is acquired through experience. Despite theoretical divergences, these philosophical discussions laid a foundational framework for subsequent theories on how we represent the world. By asserting that the human mind actively structures the experience of the world, these ideas positioned the subject as an active participant in knowledge construction. This perspective inspired numerous subsequent theories and research endeavors aimed at elaborating and complementing this concept, such as constructivist theories of perception.

The constructivist perception theory [87, 99, 46, 108] suggests that the brain faces significant challenges rendering the world without having direct access to it and, therefore, imposes a priori restrictions to model the sensorial experience. The brain receives only sensorial codifications, not reality itself. It transforms these codifications, preventing different sensorial modalities, into a world comprehension of how we know. Even though it seems simple because our perception is continuous and fluid, this task is extremely complex because the brain only receives the effects the physical world imposes on the sensors, not the causes themselves (the elements of the world that originate perception). Consequently, to represent the world, the brain must infer the causes from effects, a process called the *inverse problem of perception* [111, 142].

The inverse problem of perception is exacerbated by the fact that there are infinitely many possible solutions, and sensorial information is ambiguous and noisy. For example, during visual perception, the brain receives information about the spatial distribution of intensity and length of incident light. From this information, the brain must infer the spatial disposition of objects (causes) that gave rise to the perceived image. This process is highly complex because different object sets, arranged in distinct manners and under different lighting conditions, can generate the same image in the retina. However, the brain makes this task appear simple, given that we have a continuous understanding of the world.

The brain solves the inverse problem, showing a stable solution when imposing a priori restrictions based on an internal world model molded by previous knowledge, experience,

and context. This model defines what is plausible, possible, and impossible. According to Ballard et al. [117], the world model is not only a cognitive abstraction but is embedded in neocortical structure as an integral part of brain functioning. For him, the brain is a structure with hierarchical and modular circuits that communicate using bottom-up and top-down signals. The top-down signals represent the brain's expectations about the following sensorial inputs, while bottom-up signals indicate incompatibility between neural real activity and the expected, generating a prevision error. This error corrects iteratively expectations in high layers, allowing the brain to make more accurate inferences about the world and learn from mistakes as we continually interact in the external world.

When discrepancies arise between actual and predicted signals, neurons adjust themselves to enhance their predictive accuracy, sustaining a continuous process of adaptation and learning. When predictions align with sensory input, the internal model is well-calibrated for that specific phenomenon. Accurate prediction implies that the brain can already simulate and infer the causes underlying the sensory stimulus, effectively representing the phenomenon that generated it. This inference is possible because the internal model incorporates representations of the laws or patterns governing the external world, meaning that predictive accuracy directly reflects the individual's knowledge of specific real-world situations. This knowledge is implicitly encoded in the top-down processing flow's neuronal activation and deactivation dynamics [93].

Several scientific validations and experiments support the hypothesis that the human neocortex, particularly in the prefrontal area, learns a predictive model of the world in a hierarchical form [27, 32, 40, 37, 117]. From a neuroscience perspective, this world model operates as a simulator, generating sensory predictions based on motor actions. In this line, Friston et al. [37] conceptualize the brain as a hierarchical inference machine. They proposed that the brain organizes the world as a hierarchy or cascade of systems encoding causal sensory relationships. They developed equations that model the neuronal dynamics within this hierarchical framework, enabling the recognition and prediction of sensory trajectories. According to the authors, predictive signals in the brain flow hierarchically in a top-down and bottom-up manner. When discrepancies arise between these signals, the brain employs the principle of free-energy minimization to reduce the differences between its internal model and the perceived input, either by adjusting neuronal structures or performing actions in the real world to align sensory perceptions with predictions. Similarly, Reilly et al. [101] investigated the brain's predictive function by suggesting that it has several models of specific aspects of the world embedded in different regions. This model encodes the possible causes of sensory inputs, such as a generative model, whose focus is to understand the trajectory of sensory representations.

The initial discoveries related to the brain's structure emerged from experiments conducted by David Hubel and Torsten Wiesel on the visual cortex. They inserted electrodes into mammals' visual cortex to record neurons' electrical activity while presenting different visual stimuli. They introduced a variety of visual stimuli, such as points of light and bars of light in various orientations. Upon analyzing the neuronal responses, they discovered that specific neurons in the primary visual cortex responded preferentially to edges or lines with particular orientations; these were termed *simple neurons*. In contrast, other neurons responded to more complex patterns, such as the movement of edges in a specific

direction, and these were termed *complex neurons*. Subsequent experiments showed that simple cells relay their information to complex cells, which in turn relay information to even more complex cells, demonstrating the hierarchy of the visual system [155, 39].

Subsequent discoveries have shown that the hierarchical structure of the neocortex is not limited to the visual cortex but extends throughout the entire neocortex. Vernon Mountcastle demonstrated that the human neocortex is organized horizontally into approximately six layers and vertically into cortical minicolumns. The cortical minicolumn is the smallest unit of a mature neocortex, composed of a column of neurons interconnected vertically across the horizontal layers. A set of cortical minicolumns linked by short-range horizontal connections, sharing the same static, dynamic, and physiological properties, forms a cortical column, also known as a module. Furthermore, Mountcastle observed the relationship between sensory perception and cortical columns, identifying specific columns for each sensory modality. He discovered that lesions in particular columns affected only specific sensory areas of the body, demonstrating the modular division of sensory signals. Thus, the neurons within a column handle responses from a small part of a specific sensory modality of the body [96].

Several theories and experimental studies have proved that the brain predicts the world by treating expected stimuli differently from unexpected ones. Press et al. [40, 112] published an opinion article emphasizing the importance of the receptive paradox in learning processes. In support of this, Gillon et al. [40] demonstrated experimental evidence from the visual cortex of rats, validating the hypothesis that the brain functions as a hierarchical predictive system, refining its internal models over time by distinguishing between expected and unexpected stimuli. Furthermore, Jeff Hawkins [53] proposed the *Thousand Brains Theory*, a framework derived from computational principles observed in the human neocortex through reverse engineering. This theory highlights the neocortex's ability to independently learn predictive models of the world based on an abstract system of expectations intrinsically tied to sensory perception. It also emphasizes the hierarchical, modular, and sparse structure that supports these predictive capabilities.

For some authors, the neocortex is the sparse coding of sensory inputs [100]. When we are subjected to sensory stimuli from the environment, only a few neurons are activated, with most neurons remaining inactive or showing low activity [119]. This evolutionary strategy has provided the neocortex many advantages, such as reducing complex information to a simple signal with few active neurons, which can be quickly processed in the hierarchy with low energy cost. Furthermore, this strategy significantly enhances the capacity for representation and memory association. According to Graham and Field [44], sparse coding has even more profound and more significant origins beyond the evolutionary advantage of energy efficiency, as it represents an efficient adaptation for better representing the world, given that the world itself is sparse. For example, most trees and vegetation form a constant and predictable background in natural scenes such as a dense forest. However, animals or fruits crucial for survival are rarer and less predictable.

The hierarchical organization of the neocortex works synergistically with sparse coding to facilitate modularity, enabling the brain to prioritize relevant and critical sensory information. Sparse coding, activating only a small subset of neurons, allows for the efficient representation of sensory inputs. This selective activation complements the hier-

archical processing structure, which aids in recognizing patterns, identifying objects, and predicting environmental interactions. Through continuous sensorimotor integration, the neocortex refines its internal models of the world, adapting to its sparse and complex conditions and improving its predictive capabilities [90, 161, 32].

In addition to the neuroscientific foundations that explain the functioning of the brain's predictive structure, psychology has also played a crucial role in understanding how our internal world model facilitates the development of intelligence and adaptation. In 1943, philosopher and psychologist Kenneth Craik was the first to formalize the term mental world model to explain that we understand the world by constructing a world model in our minds. Just as Kant argued that the human mind actively structures experience through innate categories, Craik expanded this notion by proposing that the brain builds a dynamic and functional representation of reality, enabling us to predict, interpret, and interact with the external world. Craik proposed that the human brain works as a machine capable of modeling real-world events to anticipate and guide their actions. He argued that this capacity is an evolutionary and essential trait for humans, primarily because it enables explanations of external events, the anticipation of future occurrences, and the adaptation of behaviors to solve problems [20].

Craik posits that through our mental model, we can simulate various alternatives to a given situation and identify the best course of action, thereby avoiding problems before they arise. He proposed that constructing such a model involves three stages following the observation of reality: first, translating the observation into words; second, deducing an assertion, which entails formulating a logical conclusion based on premises deemed true or false; and finally, connecting this assertion to the external world. In his formulation, emphasizing the importance of sensory experience in the real world, as without it, the subsequent three steps do not occur. By defining deduction as a key step in the process, he concluded that every mental world model represents an individual's assumption about reality. In other words, the mental model is not an exact replication of the world but rather an interpretation shaped by the individual's sensory experiences, perceptions, and prior knowledge. Consequently, these models are subject to errors and revisions. Nonetheless, they serve as a crucial adaptive tool, enabling humans to navigate the complexities of the real world effectively [20].

Subsequently, while studying the dynamics of complex systems, Jay Forrester [36] reinforced this theory by asserting that humans exist within a complex system (the external world) governed by principles not fully understood. He emphasized that, due to our cognitive limitations, we create simplified representations of reality through mental models to make it more comprehensible. Furthermore, he argued that the construction of these models is influenced by our sensory constraints, subjective perceptions, and individual experiences, which determine what is filtered and used to represent reality. Perception highlights certain aspects of the environment while disregarding others, rendering each mental model subjective and unique to the individual. Forrester also contended that what we manage to abstract from the real world is what we carry within these mental models. In other words, these models do not fully reflect reality but instead consist of selected concepts and their relationships. According to him, no one can fully conceptualize the complexity of the world, a government, or a nation. We rely on these simplified

representations to interpret and interact with the real system.

Similarly, Wind et al. [18] argue that the world model is a mental construct through which individuals form perceptions and guide their actions about various aspects of their existence. In their theory, they emphasize the role of selective perception in constructing this model, explaining how it underlies our flawed understanding of certain aspects of reality, often leading to errors in identifying actual threats and opportunities. They highlight that the human mind tends to disregard most sensory stimuli, focusing only on those aligned with internal beliefs and expectations. Subsequently, Michael Shermer [135] expanded on this notion of perceptual selectivity through belief-dependent realism (BDR), which posits that our beliefs shape our perceptions of reality. According to Shermer, these beliefs are formed for various subjective, personal, emotional, and psychological reasons within environments shaped by family, friends, colleagues, culture, and society. Once established, individuals defend, justify, and rationalize these beliefs using intellectual reasoning, compelling arguments, and rational explanations. Shermer asserts that while reality exists independently of the human mind, our understanding of it is modulated by our beliefs, which guide perception and lead us to construct a world model that aligns with our internal convictions.

Recently, Matsuo et al. [89] presented a theoretical framework with an overview of how the world model is constructed, modified, and used throughout our lives to improve our intelligence. The framework has two main systems, as illustrated by Figure 2.1: the low-level sensorimotor system responsible for processing concrete patterns of the real world and the high-level symbolic system, which represents world model abstract information through imagination, thought, and language. They highlight the fundamental role of sensory experience through active exploration of the world to build our initial model and the role of the symbolic system to represent this model and change it later. Furthermore, the symbolic system facilitates the creation of abstract concepts derived from physical experiences in the world model and enables complex planning.

In this framework, the authors demonstrated that in addition to sensory perception, language also acts as a fundamental component both to represent the world model and to change it. It can influence our belief system, perception, and world model. When we communicate, we share parts of our internal representation with others, structure our thoughts, and convey emotions and abstract concepts through narratives that organize our experiences and reflect our perception of reality. Moreover, since our world model is inherently fuzzy and incomplete, language helps to organize abstract concepts in a manner that can be systematically understood by the external world and structured in thought. During a conversation, we interact with individuals with world models different from our own, which linguistically emerge, influence, and modify our model. Each participant employs a distinct mental model to interpret the topic under discussion in a conversation. As the dialogue progresses, these models may evolve, even if the subject matter remains unchanged.

Thought is profoundly dependent on our world model. Through it, we manipulate our world model to imagine situations and test potential solutions without experimenting with all possibilities in the real world. Reasoning, for example, relies on these imaginations to arrive at the best conclusions. Johnson-Laird [72] were among the first researchers to

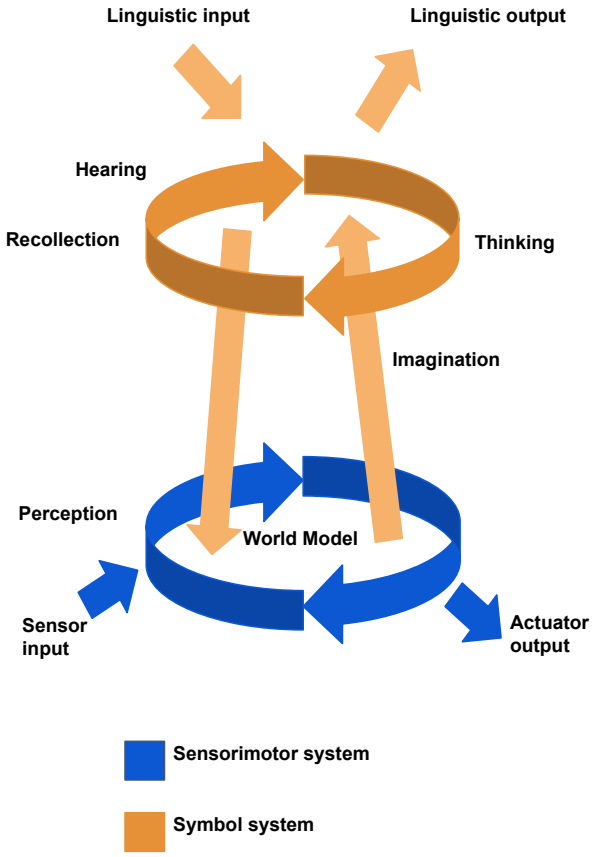


Figure 2.1: The Theoretical World Models Framework for humans [89]. A priori, the world model is constructed through sensorimotor interactions with the environment, also called a low-level system. This system generates the world model that emerges through symbolic elements of cognition, which are imagination, thought and language. The symbolic system in turn also has the capacity to alter the world model through language.

suggest that reasoning (a form of conscious thought) involves the manipulation of mental models to explore alternatives, predict outcomes, and evaluate consequences. He argues that different forms of reasoning, such as logical, deductive, and inductive, are based on inferences made from manipulations of the world model. According to him, reasoning errors occur because there are gaps in the world model being used. Peirce [107] also emphasized the importance of the world model in deductive reasoning. According to him, we construct a premises diagram, identify implicit relationships, and validate or refute these relationships through mental simulations with the world model, which we permit to extract conclusions.

Similarly, Lecun [84] asserts that reasoning relies on a world model because it is highly adaptable. According to him, we do not have multiple world models but rather one highly configurable and flexible model that is shared across different tasks, enabling reasoning by analogy. Furthermore, we are equipped with pre-programmed basic behaviors and intrinsic goals to update this world model while satisfying basic survival needs. Once we know how the world works, we learn new complex tasks, predict the consequences of our actions, plan and foresee the outcomes of successful actions, and avoid dangerous situations. For instance, in a soccer game, a player only has milliseconds to decide the

best course of action, which is less time than it takes for visual signals to reach the brain. However, they act swiftly due to their internal world model, which predicts where the ball will be in the next instant, allowing for preemptive positioning. Similarly, if a car brakes suddenly while driving on a busy highway, the driver reacts almost instantly, even before consciously processing all the details, because their world model assumes that cars maintain a steady speed or decelerate gradually. An unexpected sudden brake violates this expectation, triggering an immediate alert and enabling the driver to brake or swerve to avoid a collision.

Lecun highlights that in addition to the neocortical structure and learning based on unexpected events, the body, and intrinsic goals play a crucial role in constructing the world model. They enable continuous environment exploration, providing neocortical circuits with a constant flow of new and diverse information. Constant exploration, combined with the predictive structure of the brain, allows us to learn fundamental concepts about the world in the first days, weeks, and months of life. In a short time, we acquire spatial notions and understand that the world is three-dimensional and that objects and sounds have relative distances from us. By developing the notion of depth, we recognize the existence and occlusion of objects and categorize them based on their appearance and behavior. We also learned that objects have a dynamic movement and do not appear spontaneously, disappear, or change shape but rather move through space, occupying only one place at a time. These concepts allow the development of intuitive physics, including notions such as stability, gravity, and inertia, and enable the understanding of the effects of objects and our actions in the world. From that point, we infer cause-and-effect relationships that are the basis for acquiring linguistic and social knowledge.

The construction of the world model involves many elements, from sensory experiences and emotions to language and social interaction. Given the vast existing theory, we considered **some of its main concepts** in implementing our framework: **sensory experience**, **intrinsic motivation**, and **the structural and predictive aspects of the neocortical circuit**. There is a consensus among philosophers and researchers that the world model, a priori, is constructed through continuously exploring the body in the environment. The body plays an essential role in this process, as its senses filter, encode, and send the information received to the neocortex. The actuators execute actions in the environment guided by primitive intrinsic objectives, especially at the beginning of life, allowing various experiences. We, therefore, focus on integrating these pillars to develop a sensorimotor world model in our agent without the influence of symbolic elements. We also highlight the importance of the neocortical structure, which provides flexibility and adaptability to the model. In the following sections, we present each of these pillars individually with some important theoretical foundations and the computational tools that support the implementation of each pillar in our computational framework.

2.2 Embodied Cognition

Embodied Cognition is an approach in cognitive science that emphasizes that the body (e.g., sensations and bodily experiences) is essential to understanding the world, construct-

ing conceptual knowledge, and forming meaning. According to this perspective, cognition cannot be comprehended solely as an internal brain process; instead, it should be viewed as an activity distributed among the brain, the body, and the external world [153]. In this view, the body inherently constrains, regulates, and shapes the nature of mental activity, which is an integral part of cognitive processing. Furthermore, instead of being centralized and distinct from low-level sensorimotor functions, mental activity is profoundly grounded in these functions [34, 134].

The embodiment thesis challenges three main principles defended by traditional approaches. First, the information conveyed by a mental representation has no modality-specific features. In this sense, representations are autonomous from the sensorimotor system and its operational details. Second, knowledge is represented propositionally, meaning emerges from the relations among the constituent symbols. Finally, internal representations instruct the motor system, which is essentially separate and independent of cognition, and so cognitive processing is not significantly limited, constrained, or shaped by bodily actions [34].

Experiments provide evidence for the role of bodily experiences in the mental reconstruction of events. Sensorimotor regions are activated even without direct sensory stimuli or behavioral responses, such as during processes of imagination, planning, or recollection, suggesting that, even when disconnected from the environment, knowledge processing and representation continue to be supported by patterns of sensorimotor experiences [134]. In this line, Barsalou [10] shows that during perceptual experience, association areas in the brain capture bottom-up activation patterns in sensory-motor areas. Subsequently, in a top-down manner, these association areas partially reactivate sensory-motor regions to implement perceptual symbols. For example, when we think about a *lake*, we activate sensory-motor areas of the brain that were engaged during previous encounters with real lakes. A lake-related thought reactivates areas of the visual cortex that respond to visual information corresponding to lakes, areas of the auditory cortex that respond to auditory information related to lakes, and areas of the motor cortex corresponding to actions typically associated with lakes. However, this activation is suppressed to not result in actual movement. The result is a *lake* concept reflecting the types of sensory and motor activities unique to the human body and its sensory systems. A *lake* signifies something like to "a thing that looks like this, sounds like this, smells like this, allows me to swim in it like this."

For many years, behaviors such as gesturing, our bodily movements to visualize the environment, the use of mirror neurons in social understanding, and the externalization of cognitive processes through the body have been extensively observed in studies [34]. Behavioral research indicates that gesturing, body postures, and facial expressions serve as strategies to simplify mental processing, alleviating cognitive load and rendering tasks more manageable. For instance, using fingers for counting enhances the representation and understanding of mathematical concepts, facilitating arithmetic learning and reducing the cognitive load involved in calculations [134]. Furthermore, behaviors involving head and limb control reveal that our perception is influenced by the actions we undertake to perceive, demonstrating that perception is not a passive process but an active one guided by various cognitive elements. The role of mirror neurons in understanding the actions

of others has also been a focal point of analysis. These neurons activate when we observe someone acting and carry out the same action ourselves. This simultaneous activation suggests that our ability to comprehend the intentions and mental states of others is deeply rooted in our own motor experiences. Therefore, our understanding of one another presupposes our motor system [134, 34].

Foglia et al. [34] argue that the body plays two distinct yet interconnected roles in cognition. The first role pertains to how the body can function as a cognitive constraint; for example, in color perception, sound localization, categorization, and spatial metaphor. Concepts and experiences of colors reflect the properties of the retinal cells and the features of the visual apparatus; sound detection owes its peculiarity to the distance between the ears and spatial metaphors are rooted in sensorimotor experiences. In this line, Lakoff et al. [82, 81] argue that conceptualization, thought, and language are deeply rooted in bodily experiences, asserting that the nature of our bodies shapes our possibilities for conceptualizing and categorizing objects in the external world. They contend that many basic concepts we possess are acquired through direct physical experience and subsequently used to learn more abstract concepts through metaphors.

For instance, the basic concepts of *forward* and *backward* emerge from having only two eyes and a frontal side, toward which we direct our vision and actions, and a rear side that we cannot see directly. This leads us to create abstract concepts, metaphorically associating them with physical experiences, such as linking the future to something that *lies ahead* and the past to something that has been *left behind*, or when we express emotions, as when saying, *today I feel very down*, associating the abstract concept of sadness with a basic spatial concept related to being crouched. This perspective implies that different types of bodies would lead to different ways of conceptualizing the world; for example, an organism with radial symmetry, such as a jellyfish, may not have a clear distinction between front and back, resulting in a categorization of space that is entirely different from our own. Metaphors, therefore, make communication more captivating and reflect our embodied experience as exploring beings [81].

The second role occurs when the body distributes cognition, spreads cognitive tasks between neural and non-neural structures, and partially realizes mental phenomena. Examples are during gestures in oral communication. Although traditionally considered solely as communication tools, gestures significantly impact cognitive development, especially language and vocabulary acquisition. Research indicates that gesturing while speaking makes it easier for the listener to understand the message and contributes to vocabulary enrichment and language learning for the speaker [121, 122]. Studies have shown that children who use gestures during communication have more robust linguistic development and more ability to express and understand complex concepts. Similarly, studies show that practicing motor activities such as squeezing a sponge for three consecutive weeks improves hand and wrist performance tests and presents a significant expansion of the primary motor cortex and somatosensory cortex, reinforcing how the body shapes cognitive processes [61].

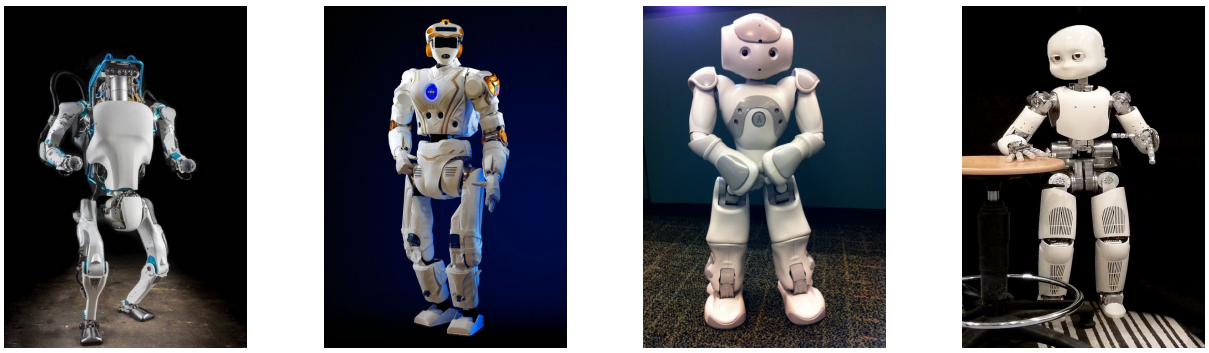
The integration of multiple sensory modalities is another central aspect of embodied cognition. The human neocortex merges information from different senses, such as vision, hearing, touch, smell, and taste, allowing a unified and coherent perception of the

environment. This capacity for sensory integration is fundamental for the formation of concepts and the execution of coordinated actions, reflecting the deeply interconnected nature of cognitive and bodily processes. Neuroimaging studies corroborate this interdependence, showing that sensorimotor areas are activated during conceptual tasks, indicating that cognition is distributed across several brain regions associated with bodily experiences [10].

In the context of artificial agents, embodied cognition has significant implications for the design and functionality of these systems. For embodied artificial agents to interact effectively in complex and dynamic environments, they need to constantly engage in environment integrating multiple sensory modalities and be able to adapt their behaviors based on bodily feedback, suggesting that consideration should be given not only to the use of complex learning algorithms but also the incorporation of physical structures and sensors that allow for rich, contextualized interaction with the real world.

2.2.1 Humanoid Robots

Humanoid robots are essential for the experimentation of fundamental concepts of embodied cognition, such as active perception, sensorimotor learning, multimodal learning, adaptation, and world representation based on bodily experience, approaching the principles of biological intelligence. A humanoid robot is a robotic system designed with a body structure that resembles that of a human, as illustrated in Figure 2.2. These robots possess a head, torso, arms, and legs articulated in an anthropomorphic shape, allowing them to navigate and interact within unstructured environments. Their legs, equipped with complex motors and actuators, facilitate locomotion across uneven terrains and enable the ascent and descent of stairs. Furthermore, they have various sensors that ensure advanced sensory perception, including stereo vision, LiDAR, tactile, and pressure sensors. They also possess auditory and voice sensors, enabling them to capture ambient sounds and interact with humans. These robots have highly sophisticated low-level controllers that continuously monitor and adjust variables such as force, torque, and joint



(a) Atlas Robot (b) Valkyre Robot (c) NAO Robot (d) iCub Robot

Figure 2.2: Examples of different humanoid robots. In (a), we have the humanoid robot Atlas developed by Boston Dynamics for industrial tasks. In (b), we have the humanoid robot Valkyrie, developed by NASA for space exploration and disaster rescue. In (c) and (d), we have the NAO and iCub robots for research and education.

position, ensuring smooth, precise, and stable movements.

Depending on their daily use, humanoid robots can be designed with different sensory and structural configurations, categorized according to their application. For instance, industrial humanoid robots are characterized by a robust body structure and high precision for assembly, welding, and material handling tasks. Similarly, humanoid robots designed for space exploration and disaster response exhibit a sturdy body structure, are tall, and possess autonomous capabilities, as NASA's Valkyrie robot exemplifies. In contrast, humanoid robots for research and education are smaller, more versatile, and programmable, with body structures resembling a child's. They are primarily focused on studying human-robot interaction and technological advancements in robotics. These robots are distinguished by their diverse tactile and auditory sensors, enabling them to feel, communicate with humans, and hear their surrounding environment. Notable examples in this category include NAO and iCub.

2.2.2 Reinforcement Learning

Reinforcement Learning (RL) consists of machine learning paradigms that allow artificial agents to continuously engage in sensorimotor experiences, providing a more suitable learning method for embodied agents. RL consists of an agent who explores the environment by choosing actions to maximize the cumulative reward. Initially inspired by human behavior and control theory, RL is now applied to economics, game theory, and information theory. Also, RL is not only limited to classification or regression tasks, but it is also a framework for decision-making, knowledge representation, planning, and reacting to new and unknown elements [144]. This paradigm is of utmost importance when no labeled data is available or when the dynamics are not differentiable; in such circumstances, the model can still learn from the reward signals given by the environment.

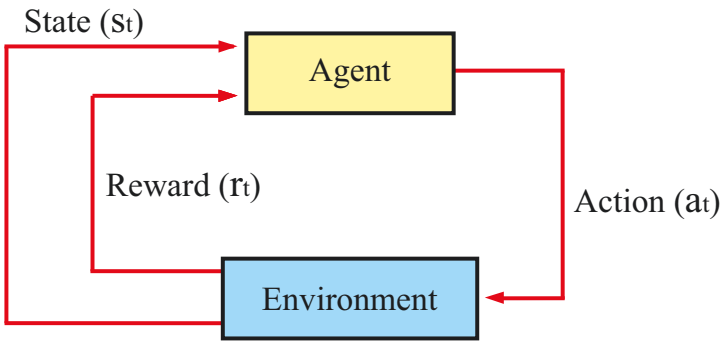


Figure 2.3: Reinforcement learning paradigm. The agent executes a policy $\pi(a|s)$ by choosing an action a_t in state s_t, then, the agent receives a reward r_t and a new state s_{t+1}. Adapted from [144]

.

The RL structure consists of various components that interact, including the environment, actions, and rewards, as shown in Figure 2.3. At each time step t, the agent follows a policy $\pi(a|s)$ to choose an action a_t based on the current state s_t. After executing the action, the agent receives a reward r_t and transitions to a new state s_{t+1}. The policy is a crucial element of reinforcement learning, as it determines the action a that the agent

takes in each state s. In this context, learning refers to the process of discovering the optimal policy, with the agent's goal being to maximize the cumulative reward over time.

Markov Decision Process (MDP) is a discrete-time stochastic control process ordinarily used to represent the environment in a reinforcement learning setup. MDP gives a framework for decision-making in which the conditional probability distribution of future states depends only upon the current state. This property is also called the Markov property, meaning that the sequence of previous states does not add new information. The MDP description is defined as a tuple $M = (S, A, P, R)$, where S is a discrete and finite set of states that model the environment, A is a finite set of actions, $P\left(s^{'}|s, a\right)$ is a probabilistic transition function that describes the effects of choosing an action $a \in A$ in a state $s \in S$ and ending in a state $s' \in S$. $R\left(s, a\right)$ is the reward earned by executing an action $a \in A$ in a state $s \in S$. Solving the MDP requires maximizing the total expected reward G_t, feasible only for small environments. Normally, a discount factor γ is used to prevent infinite accumulated values in non-episodic MDPs, with $0 \leq \gamma < 1$. Therefore, after t steps, the reward is discounted by γ^t. The total expected discounted return is

$$G_t = \sum_{k=0}^{\infty} \gamma^t R_{t+k+1}. \tag{2.1}$$

The policy $\pi : S \rightarrow A$ is the solution for an MDP; and it defines the action $a = \pi(s)$ to be chosen in each state s. A policy can be either stationary or non-stationary concerning its evolution over time. Considering the state-action relation, the policy can also be deterministic or stochastic. In stationary policies, the best choice in the state s is always the same, despite the time. In non-stationary policies, the action depends on state-action information. In deterministic policies, each state $s \in S$ is always mapped into a single action; in stochastic policies, states are mapped into a probability distribution of actions; accordingly, each action has a probability of being picked. Amongst all policies able to solve an MDP, we want to determine the optimal policy π_* that maximizes the expected total return. This can be done with a Value Iteration or Policy Iteration algorithm implemented with dynamic programming.

Policy Iteration. The Policy iteration algorithms find the optimal policy π_* by iteratively evaluating and improving a random initial policy π_0 until convergence, which means no more improvement is possible. For these methods, the policy is evaluated numerous times in order to approximate the state-value function v_π by

$$v_{k+1}(s) = \sum_{a} \pi(a|s) \sum_{s',r} P\left(s'|s, a\right) \Big[r + \gamma v_k\left(s'\right)\Big]. \tag{2.2}$$

After the state-value is updated, the policy is also updated by

$$\pi_{k+1}(s) = \operatorname{argmax} \sum_{s',r} P\left(s'|s, a\right) \Big[r + \gamma v_k\left(s'\right)\Big]. \tag{2.3}$$

where s denotes the current state, s' is the possible next state after taking action a, a is the action selected given state s, r represents the immediate reward received after executing action a in state s, $\pi(a|s)$ is the probability of selecting action a given state s

under the current policy π, $P\left(s' \mid s, a\right)$ is the transition probability to state s' given state s and action a, $v_k(s')$ denotes the estimated expected value of being in state s', and γ is the discount factor between 0 and 1 that weights future rewards.

Value Iteration. The Value iteration algorithms learn the state-value v_* for each state $s \in S$. They can be viewed as an improvement over the Policy Iteration because the state-value function does not need improvement. Thus, from an arbitrary v_0, this approach performs updates in all states of S as follows

$$v_{k+1}(s) = \max_{a \in A} \sum_{s', r} P\left(s' | s, a\right) \Big[r + \gamma v_k\left(s'\right) \Big]. \tag{2.4}$$

In the convergence, the state-value function v_k are equal to v_*, that is $\lim_{t \to \infty} |v_k\left(s\right) - v_{k-1}\left(s\right)| = 0$; and the optimal policy π_* can be extracted directly from v_*.

Dynamic programming methods based on policy iteration or value iteration become unfeasible when the environment's dynamics are unknown or difficult to compute. In such cases, different methods are required, such as Monte Carlo (MC) or Temporal Difference (TD). MC methods require only samples of states, actions, and rewards from the interactions with the environment either in simulation or online. These methods aim to solve the learning problem by averaging the sampled returns; MC methods compute the return G_t from only complete episodes $S_1, A_1, R_1, ..., S_T$; which is also its main weakness when the MDP is non-episodic. The state-value update rule is defined as

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]. \tag{2.5}$$

On the other hand, TD methods update the expected value function $V(S_t)$ for state S_t after n-steps by visiting and storing the next n-steps before the update. Popular TD methods are SARSA [123] and Q-learning [23]. For the particular case of TD(0), the update is executed immediately after a visit to S_{t+1} as

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)], \tag{2.6}$$

where R_{t+1} is the reward for the next state, α is the learning rate, and $R_{t+1} + \gamma V(S_{t+1})$ is the target for this update. TD methods are called bootstrapped since they rely on state-value estimates of future states, and not only in the current one.

2.2.3 Policy Gradient Methods

Policy gradient methods improve the policy directly by learning a function approximator parameterized by the weights $\boldsymbol{\theta}$. Therefore, this class of methods does not need to compute each state's value before determining the actions, although it might be interesting to increase the training speed and lower the variance as seen in actor-critic versions. The policy is defined as

$$\pi(a|s, \boldsymbol{\theta}) = Pr\{A_t = a | S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\}, \tag{2.7}$$

which expresses the probability of action a be chosen at time t regarding the environment is in state s at time t with parameters $\boldsymbol{\theta}$. The learning process uses the gradient ascent

to update the weights and maximize performance through

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \nabla J(\boldsymbol{\theta}_t), \tag{2.8}$$

where $\nabla J(\boldsymbol{\theta}_t)$ is the stochastic estimate whose expectation approximates the gradient in respect to $\boldsymbol{\theta}$, α is the learning rate, which determines the step magnitude. $J(\boldsymbol{\theta})$ is regularly defined using the value function for the initial state as $v_{\pi_\theta}(s_0)$. Then, the policy gradient theorem asserts that

$$\nabla J(\boldsymbol{\theta}) = \sum_s d_\pi(s) \sum_a q_\pi(s,a) \nabla_\theta \pi(a|s, \boldsymbol{\theta}), \tag{2.9}$$

where $d_{\pi_\theta(s)}$ is stationary distribution of the Markov chain using π_θ.

The parameters $\boldsymbol{\theta}$ are commonly represented by the weights between neurons in an artificial neural network. The configuration of the weights can be made in any way since the policy is differentiable regarding its parameters. The methods that follow this rule for updating the weights are policy gradients, despite using a value function. An indispensable point of these methods is their need for exploration; for this reason, the policy cannot become deterministic during training. Besides, the policy gradient theorem ensures convergence for this class of methods compared to value-based methods with non-linear function approximators.

The Proximal Policy Optimization (PPO) algorithm. PPO [129] is based on the policy gradient theorem and serves as an enhanced version of the TRPO [127] algorithm. Its primary objective is to minimize a surrogate function, which helps control the magnitude of policy updates. While TRPO employs the Kullback–Leibler (KL) divergence to assess the difference between the current and previous policies, PPO simplifies this by calculating the probability ratio and ensuring a consistent policy improvement. Essentially, the aim is to adjust the policy parameters within reliable regions, and to achieve this, the objective function is defined as

$$J(\boldsymbol{\theta}) = \mathbb{E}_t \Big[\frac{\pi_{\boldsymbol{\theta}}(a|s)}{\pi_{\boldsymbol{\theta}_{old}}(a|s)} A_t \Big], \tag{2.10}$$

where A_t is the advantage function and $\boldsymbol{\theta}_{old}$ are the old policy's parameters. The use of probability ratios is known as importance sampling, allowing unbiased estimates for the policy's samples. However, importance sampling is unbounded and often causes overestimation and underestimation. One way to solve this is to use the surrogate function, defined as

$$J(\boldsymbol{\theta}) = \mathbb{E}_t \left[\min \left(r(\boldsymbol{\theta}), \text{clip} \left(r(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon \right) \right) A_t \right], \tag{2.11}$$

assuming the probability ratio $r(\boldsymbol{\theta}) = \frac{\pi_{\boldsymbol{\theta}}(a_t, s_t)}{\pi_{\boldsymbol{\theta}_{old}}(a_t, s_t)}$, and ϵ as a hyperparameter commonly defined as 0.2. With this surrogate function, only the overestimation problem is corrected. The underestimation is considered harmless and favors the objective function's concavity. The complete PPO algorithm is described in Algorithm 1.

Algorithm 1 Proximal Policy Optimization [129].

Input: initial policy parameters θ_0, initial value function parameters ϕ_0;
for $k = 0, 1, 2, ...$ **do**

Collect set of trajectories $D_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment;

Compute rewards-to-go $\widehat{R}_t$;

Compute advantage estimates, $\widehat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k};

Update the policy by maximizing the PPO-Clip objective:

$\theta_{k+1} = \arg\max_\theta \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), g\left(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)\right)\right)$ typically via stochastic gradient ascent with Adam;

Fit value function by regression on mean-squared error:

$\phi_{k+1} = \arg\min_\theta \frac{1}{|D_k|T} \sum_{\tau \in D_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t\right)^2$, typically via some gradient descent algorithm;

end

2.3 Neo-cortical Circuit

The neocortex is structurally organized into six laminar layers (I to VI), primarily composed of pyramidal neurons and complex interlaminar connections, as illustrated in Figure 2.4. These layers are arranged into vertical cortical columns, forming a hierarchical, modular, and sparse architecture [96]. This organization enables the neocortex to efficiently process and integrate sensory and motor information across multiple levels of abstraction. Modularity allows different cortical regions — or microcircuits — to specialize in processing distinct types of information while maintaining a degree of independence, enhancing robustness and reusability across tasks. Hierarchy supports the progressive transformation of low-level sensory inputs into increasingly abstract and semantically rich representations, which are fundamental for higher cognitive processes such as planning, reasoning, and generalization [55, 62]. Sparse connectivity — where only a subset of neurons are active at a given time — improves neural coding efficiency, reduces redundancy, and supports pattern separation, making it easier for the brain to distinguish between similar inputs and adapt to novel situations.

Various theories propose mechanisms to explain the neocortex efficiency, highlighting how the brain dynamically and adaptively integrates information. One important theory is predictive coding, which suggests that the neocortex constructs internal world models and continuously updates these representations based on new sensory stimuli [93]. This theoretical framework posits that the brain does not merely react passively to external stimuli but actively anticipates future events by comparing internal predictions with incoming sensory information. Predictive coding implies that neural activity is governed by a continuous cycle of prediction and prediction error, where discrepancies between expectations and reality lead to adjustments in internal representations. This process occurs across different hierarchical levels, enabling the formation of increasingly abstract and generalizable representations. Such a mechanism accounts for phenomena like the

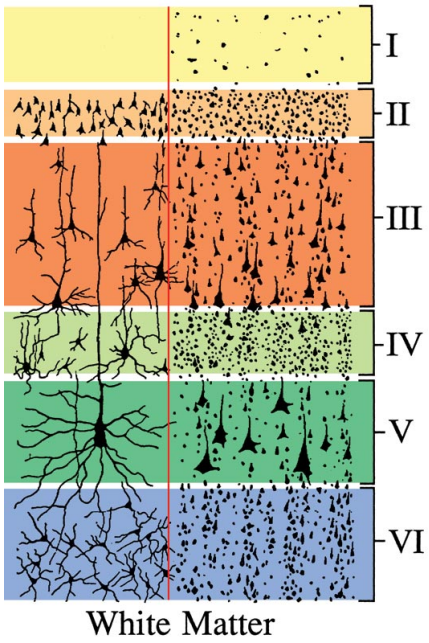


Figure 2.4: The cerebral neocortex has six laminar layers, identified from I to VI, each with specific characteristics. The thickness of each layer may vary depending on the region of the neocortex. On the left side of the illustration, the individual cell profiles are represented as they would be visualized in a Golgi stain. On the right side, populations of cell bodies are observed as they appear on Nissl staining. The layers are classified as: I = molecular layer; II = external granular layer; III = external pyramidal layer; IV = internal granular layer; V = inner pyramidal layer; VI = multiform layer. Adapted from [116].

rapid and efficient perception of environmental patterns, decision-making under incomplete information, and even lifelong learning. Other theories complement this perspective, suggesting that the organization of the neocortical circuit can also be explained by principles such as energy efficiency [37]. These theories emphasize the brain's ability to optimize computational resources while maintaining high processing accuracy.

In recent years, deep neural networks have increasingly incorporated fundamental principles of the neocortex through attention mechanisms, which simulate aspects of the hierarchical and selective processing of the human brain. Inspired by how the neocortex efficiently allocates neural resources, these networks have been designed to focus their processing capacity on specific regions of the input data or neural layers, enabling more efficient distribution of computational resources and improving learning dynamics. By selectively weighting different parts of the input, attention mechanisms allow neural networks to prioritize relevant features while filtering out less informative elements. This capability has proven particularly effective in domains such as natural language processing, computer vision, and reinforcement learning, where contextual awareness and adaptive information processing are crucial.

2.3.1 Attentional Neural Networks

Neural attention mechanisms have significantly transformed deep learning architectures, enabling models to process and prioritize information more effectively. These mechanisms dynamically assign different weights to parts of the input and neurons, allowing networks to focus on the most relevant features while ignoring less informative elements. This process closely resembles how the neocortex integrates and prioritizes sensory inputs, enhancing cognitive efficiency. By selectively allocating computational resources, attention mechanisms improve the scalability and adaptability of deep neural networks.

In deep learning, *attention* is a system composed of one or multiple modules, which allocate structural or temporal resources, select or modulate signals to perform a task. Each module consists of a function or multiple non-linear functions trained in conjunction with the neural network. Specifically, each module outputs a selective or modulating mask for an input signal. The structural resources allocated are elements of the architecture (e.g., number of neurons, number of layers), time resources refer to computation per step, number of time steps, processing time in modules of the architectures or frameworks. The task is the goal application (e.g., classification, regression, segmentation, object recognition, control, among others), and signals are given at any abstraction level (e.g., features, visual information, audio, text, memories, latent space vectors) [19].

An attentional neural network contains a set of attentional subsystems to allocate resources for processes, even in a recursive manner. An attentional subsystem, at each time step t, receives as input $\mathbf{a}$, a contextual input $\mathbf{c_t}$, a focus target $\tau_\mathbf{t}$, and inner state $\mathbf{i_{t-1}}$. And produces as output a current inner state $\mathbf{i_t}$, and current focus output $\mathbf{a_t}$, as shown Figure 2.5 and 2.6. The focused output is the main element of the subsystem because it assigns targets an importance score. Together, several attentional subsystems always perform actions to provide selection capabilities. The subsystem profile depends on the data structure and the desired output.

Different attention mechanisms have been developed, such as soft, hard, global, and local attention, each offering trade-offs between computational efficiency and interpretability [19]. Among these, self-attention is crucial in modern architectures, enabling long-range dependencies to be captured without sequential constraints. Additionally, attention mechanisms enhance efficient resource allocation, similar to how the brain dynamically modulates neural activity based on cognitive demands. When integrated into recurrent and memory-augmented networks, attention improves learning efficiency, adaptability, and long-term information retention.

Among the most recent attention-based neural networks, Recurrent Independent Mechanisms (RIMs) stand out for their similarity to the neocortical circuit, as they pioneered the introduction of attention in dynamic connections within a modular and sparse architecture for recurrent processing [42]. Unlike traditional recurrent models, RIMs employ self-attention to selectively activate independent modules based on the characteristics of the input. This modular structure aligns with the neocortical organization, where modules flexibly engage depending on sensory and cognitive demands. By combining attention with modularity, RIMs enhance generalization, adaptability, and long-term memory, making them well-suited for out-of-distribution generalization.

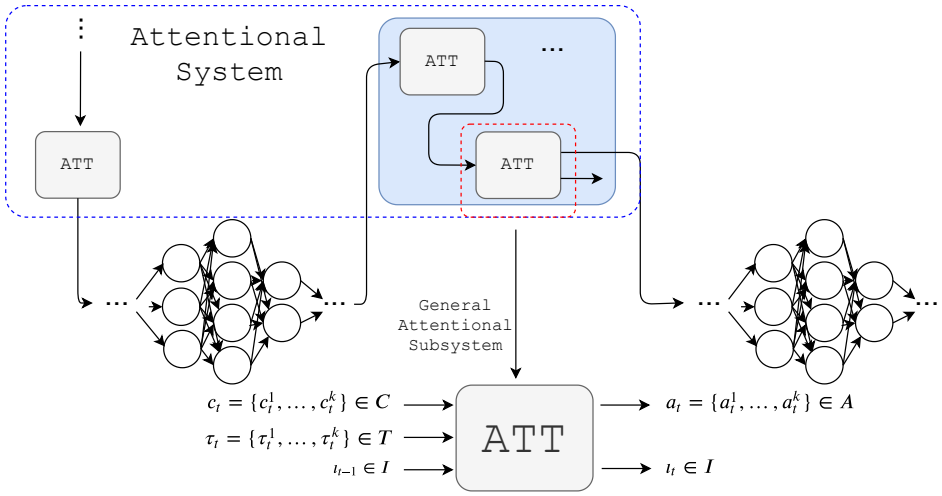


Figure 2.5: Illustration of *attention* in deep neural networks, in which several attentional subsystems are coupled in the neural networks sequentially or recurrently. Each subsystem has a different profile based on the input data's structure and sensory modality. A single subsystem receives as the primary input the focus target (i.e., the stimulus to be filtered), and sometimes auxiliary inputs (e.g., contextual information and subsystem's previous internal state) to help the mechanism guide the focus in time. Adapted from [19]

.

Symbol	Description
Context	
k	Sensory modality index.
C	Contextual input set, $C = \{c_{t-1}, \ldots, c_t\}$, $C \in \mathbb{R}$, (e.g., hidden states, memory data, sensory data).
c_t	Contextual input at time t, $c_t = \{c_t^1, \ldots, c_t^k\}$, $c_t \in C$.
c_t^k	Contextual input from sensory modality k at time t, $c_t^k = \{c_{t,1}^k, \ldots, c_{t,n_{ck}}^k\}$, $c_{t,j}^k \in \mathbb{R}^{F_c}$, where F_c is amount of features.
Focus target	
T	Focus target set, $T = \{\tau_{t-1}, \ldots, \tau_t\}$, $T \in \mathbb{R}$.
τ_t	Focus target at time t, $\tau_t = \{\tau_t^1, \ldots, \tau_t^k\}$, $\tau_t \in T$.
τ_t^k	Focus target from sensory modality k Features for $n_{\tau k}$ elements, if τ_t^k is a data, Hyperparameters or index, if τ_t^k is a program. $\tau_t^k = \{\tau_{t,1}^k, \ldots, \tau_{t,n_{\tau k}}^k\}$, $\tau_{t,j}^k \in \mathbb{R}^{F_{\tau k}}$, where $F_{\tau k}$ is amount of features.
Inner state	
I	Inner state set, $I = \{i_{t-1}, \ldots, i_t\}$, $I \in \mathbb{R}$.
i_t	Inner state at time t, $\iota_t \in I$.
$i_t - 1$	Past inner state at time $t-1$, $\iota_{t-1} \in I$.
Focus output	
A	Focus output set, $A = \{a_{t-1}, \ldots, a_t\}$, $A = \{x \in \mathbb{R} : 0 < x < 1\}$ or $A = \{x \in \mathbb{Z} : 0 \leq x \leq 1\}$.
a_t	Focus output at time t, $a_t = \{a_t^1, \ldots, a_t^k\} \in A$.
a_t^k	Focus output from sensory modality k at time t, $a_t^k = \{a_{t,1}^k, \ldots, a_{t,n_{\tau k}}^k\}$ are attention scores, $a_{t,j}^k \in \mathbb{R}^{F_{\tau k}}$ or $a_{t,j}^k \in \mathbb{R}$, $a_t^k \in \mathbb{R}^{n_{\tau k} \times F_{\tau k}}$ or $a_t^k \in \mathbb{R}^{n_{\tau k}}$.

Figure 2.6: Notation for unified attention model. Note the notation supports recurrence and multimodality. Adapted from [19].

These modules are highly configurable, allowing for different structural organizations. Madan et al. [4] integrated different sources of knowledge into the modules, such as sentiment analysis and syntactic analysis data, and observed that the modules interact effectively, significantly reducing parameter space and resources. They also identified differentiated activation patterns, indicating that the modules are activated according to the relevance of the inputs to the target task. Finally, Mittal et al. [94] explored the hierarchical organization of the recurrent independent modules with bidirectional bottom-up and top-down connections and demonstrated how this change improved model learning and generalization. This work is particularly interested in the organizational structure

proposed by Mittal et al. [94], which closely resembles the three key structural principles of the neocortex: modularity, sparsity, and hierarchy with bidirectional communication.

Bidirectional Recurrent Independent Mechanisms (BRIMs)

The Bidirectional Recurrent Independent Mechanisms (BRIMs) [94] is explicitly built to route the flow of bottom-up and top-down information between modules, promoting selection iteration between the two levels of stimuli, as shown in Figure 2.7. BRIMs have received much attention from the scientific community for presenting results surpassing the state-of-the-art OOD generalization. When the distribution of the test set changes in a minimal aspect, the classic models fail significantly, including some fully attentional neural networks, such as Neural Transformers [148]. In contrast, BRIMs perform state-of-the-art results in OOD generalization [94].

BRIMs mainly use self-attention to link identical LSTM modules, generating a very sparse and modular architecture with only a small portion of modules actives at time t. The approach separates the hidden state into several modules so that upward interactions between bottom-up and top-down signals can be appropriately focused. The layer structure has concurrent modules so that each hierarchical layer can send information in the bottom-up and top-down directions. Bottom-up attentional subsystems communicate between modules of the same layer, as well as the composition of hidden states

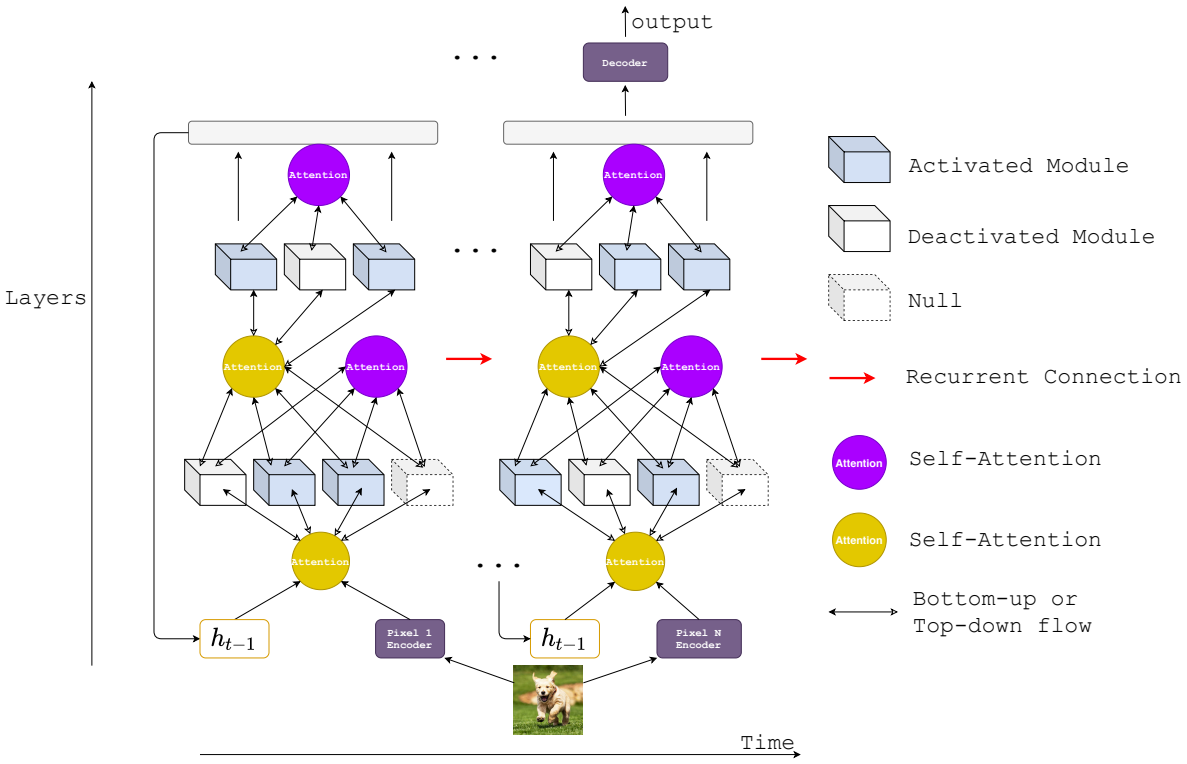


Figure 2.7: The BRIMs architecture is composed of several layers of recurrent independent mechanisms (RIMs) that enable the information flow between activated modules in a top-down and bottom-up manner. BRIMs provides better generalization results for out-of-distribution problems due to mainly the introduced sparsity between modules via attention.

Algorithm 2 Single recurrent step for an L layered BRIM model [94].

Result: RNN Cell forward for L layered BRIMs

z: Input h_l: Hidden state of layer l represented as flat vector $h_l[k]$: Hidden state of k^{th} module of layer l n_l: Number of modules in layer l m_l: Number of modules kept active in layer l ϕ: Null vector

All Query, Key, Value networks are fully connected neural networks $\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{A_S}, \mathbf{A_R}$ denotes matrices

Note: Unless specified, all indexing start with 1

`Function BRIMsCell(`*x, h_l, ..., h*`)`**{**

h_0 = x

for $l = 1$ **to** L **do**

for $k = 1$ **to** n_l **do**

$\mathbf{Q}[k]$ = Input Query_l,k(h_l[k])

end

$\mathbf{K}[0], \mathbf{V}[0]$ = Null Key Value_l(ϕ)

$\mathbf{K}[1], \mathbf{V}[1]$ = Input Key Value_l(h_l-1)

$\mathbf{K}[2], \mathbf{V}[2]$ = Top-Down Key Value_l(h_l+1)

$\mathbf{A_S} = \text{Softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d_{att}})$

$\mathbf{A_R} = \mathbf{S}\mathbf{V}$

Sort $\mathbf{A_S}[:, 0]$ and take lowest m_l as active **for** *k s.t. module k is active* **do**

$h_l[k]$ = RNN_l,k($A_R[k]$, $h_l[k]$)

end

for $k = 1$ **to** n_l **do**

$\mathbf{Q}[k]$ = Communication Query_l,k($h_l[k]$)

$\mathbf{K}[k]$ = Communication Key_l,k($h_l[k]$)

$\mathbf{V}[k]$ = Communication Value_l,k($h_l[k]$)

end

$\mathbf{A_R} = \text{Softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d_{att}})\mathbf{V}$

for *k s.t. module k is active* **do**

$h_l[k]$ += $\mathbf{A_R}[k]$;

end

end

return h;

in initial layers using the entry x_t as the target, and via top-down attention modules in different layers communicate with each other requesting information about hidden states of previous and posterior layers to compose the current hidden state. Next, we present Algorithm 2 for the L layered BRIMs and a detailed explanation for the entire BRIMs construction.

Multi-layer Stacked Recurrent Networks. Most multi-layer recurrent networks are configured to operate feed-forward and bottom-up, meaning that higher layers are fed with information processed by inferior layers. In this sense, the traditional stacked RNN for L levels is defined as

$$\mathbf{y}_t = D(\mathbf{h}_t^L), \tag{2.12}$$

$$\mathbf{h}_t^l = F^l(\mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^l), \tag{2.13}$$

$$\mathbf{h}_t^0 = E(\mathbf{x}_t), \tag{2.14}$$

where $l = 0, 1, ..., L$. For a specific time step t, $\mathbf{y}_t$ refers to the prediction, $\mathbf{x}_t$ to the input, and $\mathbf{h}_t^l$ to the hidden state at layer l. D defines the decoder, E is the encoder, and F^l represents the recurrent dynamic at layer l (e.g., LSTM, GRU).

Recurrent Independent Mechanisms (RIMs). Proposed by Goyal et al. [42], RIM is a single-layered recurrent architecture consisting of hidden state $\mathbf{h}_t$ decomposed into n modules. The main property introduced in this model is that only a small subset of modules is activated on a specific time step. In this sense, the hidden states are updated following these steps: a) a subset of modules is activated depending on their relevance to the input; b) the activated modules independently process the information; c) the activated modules have contextual information from the other modules and update their hidden state to store such information.

Key-Value Attention. The Key-Value Attention, also called the Scaled Dot Product, is responsible for the updates in RIM. This attentional mechanism is also employed in the self-attention modules and is widely used in Transformer architectures. The attention score $\mathbf{A}_S$ and an attention modulated result $\mathbf{A}_R$ are computed as

$$\mathbf{A}_S = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right), \tag{2.15}$$

$$\mathbf{A}_R = \mathbf{A}_S\mathbf{V}, \tag{2.16}$$

where $\mathbf{Q}$ is the set of queries, $\mathbf{K}$ are the keys with d dimensions and $\mathbf{V}$ are the values.

Selective Activation. The selective activation is employed by defining that each module creates queries $\bar{\mathbf{Q}} = Q_{inp}(h_{t-1})$ which are then combined with the keys $\bar{\mathbf{K}} = K_{inp}(\phi, x_t)$ and values $\bar{\mathbf{V}} = V_{inp}(\phi, x_t)$ obtained from the input x_t and zero vectors ϕ to get both the attention score $\bar{\mathbf{A}}_S$ and attention modulated input $\bar{\mathbf{A}}_R$. Based on this attention score, a fixed number of modules m are activated for which the input information is most relevant. In this sense, the null module provides no new information and has a low attention score. The activated set per time step is $\mathcal{S}_t$.

Independent Dynamics. After the input is modulated by attention, each activated module has its own hidden-state update procedure, as

$$\bar{\mathbf{h}}_{t,k} = \begin{cases} F_k(\bar{\mathbf{A}}_{R_k}, \mathbf{h}_{t-1,k}) & k \in \mathcal{S}_t \\ \mathbf{h}_{t-1,k} & k \notin \mathcal{S}_t, \end{cases} \tag{2.17}$$

where F_k is any recurrent update procedure (e.g., GRU, LSTM).

Communication. Each module consolidates the information from all the other modules for every independent update step. The attention mechanism is utilized to consolidate

this information in a similar way as in selective activation. The active modules create queries $\hat{\mathbf{Q}} = Q_{com}(h_t)$ which act with the keys $\hat{\mathbf{K}} = K_{com}(h_t)$ and values $\hat{\mathbf{V}} = V_{com}(h_t)$ generated by all modules and the result of attention $\hat{\mathbf{A}}_R$ is combined to the state in time step t as

$$\mathbf{h}_{t,k} = \begin{cases} \bar{\mathbf{h}}_{t,k} + \hat{\mathbf{A}}_{R_k} & k \in \mathcal{S}_t \\ \bar{\mathbf{h}}_{t,k} & k \notin \mathcal{S}_t. \end{cases} \tag{2.18}$$

Composition of Modules. The original hidden state $\mathbf{h}_t^l$ found in RIM is decomposed into separate modules for each layer l and time t. Therefore, instead of representing the state as just a fixed dimensional vector $\mathbf{h}_t^l$, the representation is defined as $\{((\mathbf{h}_{t,k}^l)_{k=1}^{n_l}, \mathcal{S}_t^l)\}$ where n_l denotes the number of modules in layer l and S_t^l is the set of active modules at time t in layer l. $|\mathcal{S}_t^l| = m_l$, where m_l is a hyperparameter defining the number of modules active in each layer l at any time; layers may have different active modules. Setting m_l to be half of n_l provided good performance.

Communication Between Layers. The communication links are defined between multiple layers using key-value attention. Traditional RNNs build a strictly bottom-up multi-layer dependency; in BRIMs, instead, the multi-layer dependency considers queries $\bar{\mathbf{Q}} = Q_{lay}(\mathbf{h}_{t-1}^l)$ from modules in layer l and keys $\bar{\mathbf{K}} = K_{lay}(\phi, \mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^{l+1})$ and values $\bar{\mathbf{V}} = V_{lay}(\phi, \mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^{l+1})$ from all the modules in the lower and higher layers. Thus, the attention mechanism acts in three directions and generates the attention score $\bar{\mathbf{A}}_S^l$ and output $\bar{\mathbf{A}}_R$. The attention-receiving information from the higher layer is given by the same layer in the previous time step; the same layer in the current time step also gives the attention-receiving information from the lower layer. Only the lower layer is used for the deepest layer, and for the first layer, the input's embedded state serves as the lower layer [94].

Sparse Activation. The set S_t^l is built based on the attention score $\bar{\mathbf{A}}_S^l$, which contains modules for which null information has little importance. Every activated module gets a separate input version, which is obtained via the attention output $\bar{\mathbf{A}}_R^l$. In practice, for each activated module, the representation is defined as

$$\bar{\mathbf{h}}_{t,k}^l = F_k^l(\bar{\mathbf{A}}_{R_k}^l, \mathbf{h}_{t-1,k}^l) \qquad k \in \mathcal{S}_t^l, \tag{2.19}$$

where F_k^l represents the recurrent update unit.

Communication Within Layers. The communication is made between the different modules within each layer using the key-value attention. This communication between modules within a layer permits the modules to share information through the attention bottleneck. In the same way, queries are generated $\hat{\mathbf{Q}} = Q_{com}(\bar{\mathbf{h}}_t^l)$ from active modules and keys $\hat{\mathbf{K}} = K_{com}(\bar{\mathbf{h}}_t^l)$ and values $\hat{\mathbf{V}} = V_{com}(\bar{\mathbf{h}}_t^l)$ from all the modules to obtain the final update to the module state through residual attention $\hat{\mathbf{A}}_R^l$. The state update rule is

$$\mathbf{h}_{t,k}^l = \begin{cases} \bar{\mathbf{h}}_{t,k}^l + \bar{\mathbf{A}}_{R_k}^l & k \in \mathcal{S}_t^l \\ \bar{\mathbf{h}}_{t-1,k}^l & k \notin \mathcal{S}_t^l. \end{cases} \tag{2.20}$$

2.4 Intrinsic Motivation

Intrinsic motivation (IM) is defined as doing an activity for its inherent satisfactions rather than for some separable consequence. When intrinsically motivated, a person is moved to act for the fun or challenge rather than because of external prods, pressures, or rewards [124]. Intrinsic motivation was initially identified in experimental studies on animal behavior, where researchers observed that many organisms engage in exploratory behaviors, play, and act driven by curiosity, even without external reinforcements or rewards [152]. These behaviors occur spontaneously and provide adaptive advantages to the organism, exercising and extending one's capacities.

According to Ryan et al. [124, 102], the central characteristic that distinguishes intrinsic from extrinsic motivation is *instrumentalization*. In motivation, instrumentalization occurs when an activity is performed not for its inherent value or pleasure but because it leads to other benefits. For example, the activity is instrumentalized when a child completes their homework solely to avoid parental punishment or because they believe it will secure a good job in the future. In contrast, intrinsic motivation is at play if children do their homework because the task is enjoyable and engaging, to the point that it is as pleasurable as playing a video game. In everyday life, motivations for various actions are often mixed, with intrinsic and extrinsic goals being weighed simultaneously. However, intrinsic motivations are more easily identifiable in young children, particularly in the early stages of life. It is much easier to observe children engaging in activities such as grasping objects, playing, biting, running, and shouting without a specific goal, simply for the inherent satisfaction these activities provide.

With the emergence of the concept of intrinsic motivation, several distinct theories have been proposed to explain which characteristics of activities enable intrinsic motivation. In the 1950s, intrinsic motivation was initially explained through drive theory, which posited that it was a homeostatic mechanism arising in organisms to reduce physiological deficits [64]. Later, White [152] and Berlyne [13] criticized this theory, demonstrating that animals and humans engage in activities out of sheer interest and pleasure, contradicting the idea that biological deficits solely drive intrinsic motivation. Instead, they argued that intrinsic motivation emerges as an autonomous engagement process with the environment. Many intrinsically motivated behaviors, such as play, exploration, and learning, occur even without a physiological need to be supplied. Subsequently, Festinger [33] introduced cognitive dissonance theory, suggesting that intrinsic motivation might be related to reducing inconsistencies between internal cognitive structures and perceived information from the environment. Expanding on this idea, Kagan [74] proposed that human intrinsic motivation is driven by the desire to reduce uncertainty. However, this perspective was also criticized, as many human behaviors actively seek to increase uncertainty in a controlled manner.

Subsequently, Hunt [92] proposed the optimal incongruity theory, asserting that children and adults seek an optimal level of discrepancy between perceived stimuli and internal patterns, making certain stimuli more engaging. Dember et al. [13] expanded on this perspective by suggesting that the most rewarding situations are those with a moderate level of novelty, neither entirely familiar nor completely unknown. Other approaches have

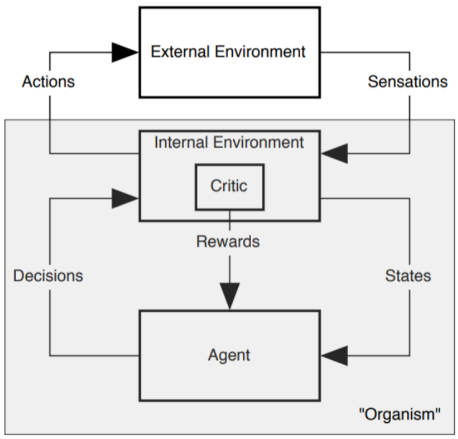


Figure 2.8: The Intrinsically Motivated Reinforcement Learning paradigm. The organism is composed of the agent and its internal environment. The organism acts in the external environment by choosing actions from its internal space while receiving sensations from the external. All rewards are internal in this paradigm, which favors the construction of organisms independent of the task and the ability to utilize high-level skills. Adapted from [140].

argued that intrinsic motivation is linked to the degree of control one has over the environment. White [152] introduced the concept of effectance motivation, while Charms [24] proposed the notion of personal causation, emphasizing the fundamental need to perceive oneself as the agent of one own actions.

In the 1970s and 1980s, Deci et al. [26] developed the self-determination theory with different aspects from previous approaches. Instead of highlighting the physiological aspects of being, they highlighted the importance of satisfying psychological aspects in the task so that intrinsic motivation arises. They highlighted that individuals become more intrinsically motivated when the needs for autonomy, competence, and relatedness are satisfied. Autonomy refers to the desire to feel in control of one's actions and decisions. Competence refers to the need to feel effective in one's activities. Moreover, relatedness involves the need to feel connected to others. When these needs are satisfied, individuals tend to be more intrinsically motivated. More recently, a more psychological line has been followed. Csikszentmihalyi [21] introduced the concept of *flow* and the elements that provide it, facilitating the emergence of intrinsic motivation.

Currently, there is no consensus on which of these theories most accurately explains the origin of intrinsic motivation in humans. However, each theory highlights valuable aspects that contribute to a broader understanding of how intrinsic motivation emerges.

Intrinsic Motivation in RL. Computationally, intrinsic motivation has been integrated into reinforcement learning systems. In this context, intrinsic motivation drives an agent to exhibit specific behaviors without relying on direct feedback from the environment. This leads to constructing a more complex organism, as shown in Figure 2.8. The intrinsically motivated organism contains the agent as a decision-maker and its internal environment, which is influenced by such decisions. In this new paradigm, the agent is only responsible for making decisions that affect the current internal state; in this process, the agent is rewarded and receives the next state. Once the internal environment

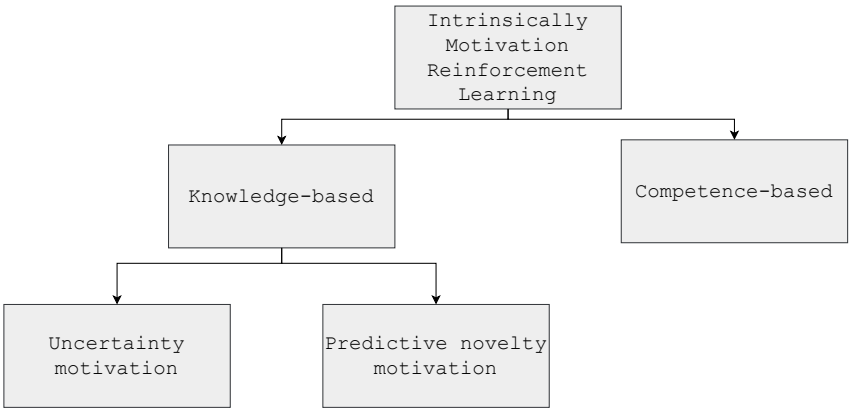


Figure 2.9: Intrinsic motivation in reinforcement learning is categorized into a knowledge-based and competence-based approach. Our work follows the knowledge-based branch, where the agent aims to predict novelty or reduce uncertainty.

changes, the organism acts in the external environment and receives new sensations. In the intrinsic motivated RL approach, all rewards are internal, which means the agent only makes decisions that affect its internal environment, and then such decisions are reflected in the external environment by actions [11].

Traditionally, approaches to intrinsic motivation in reinforcement learning are categorized into *knowledge-based* and *competence-based* approaches [102], as illustrated in Figure 2.9. Knowledge-based approaches stimulate the agent to acquire new knowledge about its environment. This approach relies on measures of dissonance or resonance between the situations experienced by an agent and the knowledge and expectations the agent holds regarding those situations. In this context, a situation refers to both a passive observation activity in which the agent does not perform actions in the environment but focuses its attention on a particular aspect of the environment, as well as active activities in which the agent executes actions and compares the actual outcomes of those actions with its knowledge and expectations. Intrinsic rewards commonly used in these approaches include curiosity, novelty, surprise, and empowerment. In contrast, competence-based approaches pertain to the agent's performance on self-generated goals. Competence-based approaches are less developed computationally and draw inspiration from the theories of effectance, self-determination, and flow. Our work is a knowledge-based approach based on curiosity reward.

2.5 Considerations

This chapter presented the foundational concepts for understanding **DreamerRL**, a framework for learning world models. Our framework employs an embodied humanoid robotic agent to explore its environment through intrinsic curiosity, using a reinforcement learning protocol. This process enables the agent to interact actively with its surroundings, exploring possibilities and adjusting its behavior based on proprioceptive feedback. To enhance embodiment, we integrate multimodality and active visual perception, fostering greater immersion in the environment and allowing the agent to combine sensory

information from multiple sources. This approach strengthens the robot's environmental awareness and facilitates the discovery of novel interaction patterns in an object manipulation setting, promoting a richer and more context-aware world representation. Furthermore, we aim to model the agent's world representation by drawing inspiration from the organizational structure of the human neocortex. To achieve this, we incorporate artificial neural networks and introduce structural biases such as sparsity, modularity, and hierarchy, which contribute to developing flexible and adaptive representations.

Our main objective is to investigate whether constructing a world model using these three mechanisms enables the agent to develop autonomously, acquiring skills that are not solely dependent on a specific task but can be adapted to other tasks or scenarios. We hypothesize that DreamerRL intrinsic exploration can provide the robot with a cognitive foundation that facilitates adaptation to new situations without requiring extensive retraining. This capability is crucial for robotic agents operating in dynamic and unstructured environments, where the unpredictability of interactions demands more versatile agents capable of handling complex challenges and interacting with the world more naturally and adaptively.

Chapter 3

Related Work

This chapter presents the most relevant literature related to our work across different dimensions. In Section 3.1, we review frameworks and architectures that incorporate world model concepts and share theoretical foundations with our approach. Section 3.2 discusses works related to curiosity in intrinsic motivation for learning agents, while Section 3.3 covers key studies in robotic manipulation focused on task adaptation.

3.1 Frameworks and Architectures for World Models

Various frameworks and architectures have been proposed to enable agents to build internal representations of their environment. One of the earliest biologically inspired approaches to world modeling is Hierarchical Temporal Memory (HTM) [57]. HTM is grounded in the neocortical theory of prediction, leveraging sparse distributed representations (SDRs) and temporal pooling to learn spatiotemporal patterns in an unsupervised manner. This framework primarily models the brain's modular, sparse, and hierarchical organization and employs a local and continuous learning strategy to predict the world. It comprises regions of computational pyramidal neurons, as illustrated in Figure 3.1 (a), which serve as a computational replica of the biological pyramidal neurons, the most abundant type in the neocortex human. These regions are further organized into vertical cortical columns, in Figure 3.1 (b), each exhibiting a uniform laminar structure composed of six horizontal layers stacked on top of one another. Within these layers, mini-columns of neurons are formed, establishing intricate connections across multiple layers. Notably, each mini-column can span several layers, and all cortical columns operate under a shared learning principle known as the common cortical algorithm [54, 56].

HTM neurons process sparse inputs through three distinct dendritic zones: the *Apical Zone*, which receives top-down feedback; the *Basal Zone*, which is responsible for lateral connections; and the *Proximal Zone*, which directly processes feedforward inputs. In neurons in layer 0, these inputs are encoded sensorial data. Each dendritic segment independently identifies patterns, with proximal dendrites specializing in input recognition, while basal and apical dendrites modulate neuronal activation. Neurons can exist in active, predictive, or inactive states, with their learning dynamics governed by permanence values, which adjust synaptic strength over time [56]. To model raw input sensorial data,

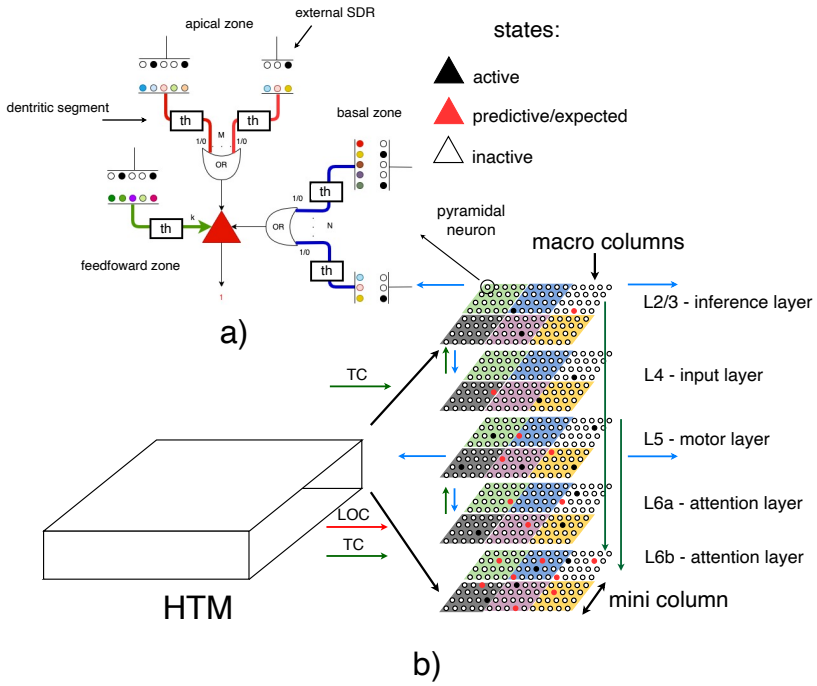


Figure 3.1: Hierarchical Temporal Memory framework. a) Pyramidal neurons. Pyramidal neurons have apical, basal, and feedforward zones. b) Hierarchical Temporal Memory with cortical connections. Arrows represent documented pathways. L4 is the input layer, L2/3 is the output layer. Green arrows (TC) are feedforward pathways, from thalamocortical (TC) relay cells, to L4, to L2/3, to L6a, and L5. Blue arrows are lateral and feedback connections. L2/3 is the inference/output layer, L4 is the input layer, L5 is the motor layer, L6a/6b are attention layers. This illustration has six macro columns and five minicolumns in each macro column.

HTM employs Sparse Distributed Representations (SDRs) [113] to encode sensory data biologically inspiredly, where a small percentage of active bits capture semantic meaning. SDRs ensure robustness to noise and redundancy, mirroring biological encoding mechanisms. Effective SDR design must preserve semantic similarity, maintain deterministic encoding, enforce a fixed dimensionality, and ensure sparsity across different inputs. In HTM, SDRs relied on rule-based hashing techniques. However, recent approaches integrate pre-trained CNNs with quantization strategies to encode more complex inputs [83].

HTM enables cortical columns to build independent object models at various hierarchy levels, with the final layer integrating these representations into a coherent and robust model. Each column processes sensory inputs independently, later combining their predictions through lateral connections. For instance, in a configuration where two macrocolumns receive touch inputs from separate sensory sub-matrices, one column may initially classify an object as a mug. At the same time, the other identifies it as an apple. However, after multiple interactions and integration at the inference layer, the system converges on the most probable representation, correctly recognizing the object as a mug, as shown in Figure 3.2. This process illustrates HTM's biologically plausible support for sensory multimodality, allowing different cortical regions to construct models later refined through cross-column communication.

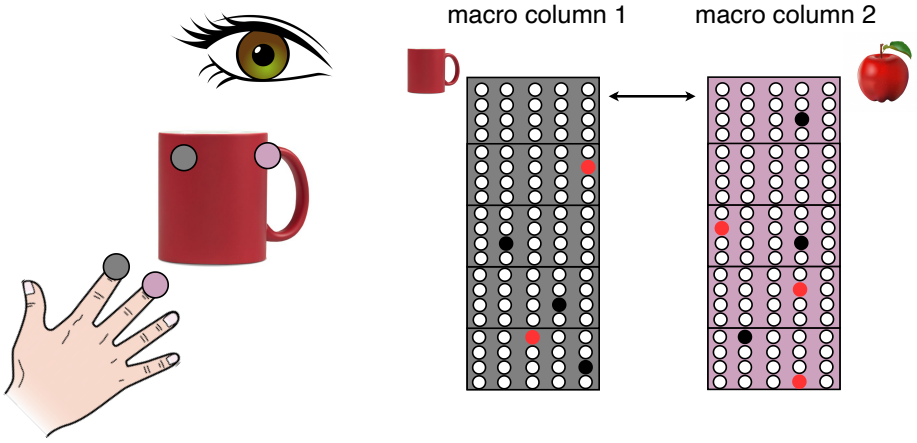


Figure 3.2: Example of the independence of cortical columns. Column 1 predicts the object as a mug, and column 2 independently predicts the object as an apple; then, through lateral communication between the columns, the model finally concludes it is a mug.

HTM neurons learn locally whenever a neuron's expected state does not align with its current activation. In other words, when sensory input violates a neuron's prediction, synaptic permanence values are adjusted accordingly. This mechanism continuously refines neural expectations to match real activation states, enhancing the ability to predict sensory patterns. As HTM is a framework derived from the reverse engineering of the brain's learning circuits, it becomes evident that the concept of a predictive world model is deeply embedded in neural structures. HTM further demonstrates that even a single neuron can generate minimal predictions about environmental stimuli, underscoring the fundamental role of this mechanism in learning [56, 58, 54].

Similarly, Lee et al. [85] proposed a sparse and hierarchical neural network, which learns via predictive coding strategy model scenes with occlusion. The SNN-PC model uses spiking neurons with asynchronous communication and adopts a hierarchical organization with Hebbian learning to generate representations in a biologically plausible manner. The model separates positive and negative error signals, contributing to greater biological plausibility, and integrates an exponential adaptive neuronal behavior. The information processing occurs in both feedforward (generating more abstract representations) and feedback (adjusting lower-level representations based on predictions made in higher layers) directions. The main aim of SNN-PC was to demonstrate that structural principles of neocortical circuits to learn a model of the world are fundamental to image task generalization. The results demonstrate that the network could reconstruct images using only 8.5% of the MNIST data and has robustness to noise and partial occlusions, maintaining accuracy in image reconstruction even under adverse conditions. In particular, the SNN-PC maintained stable reconstruction performance even with high levels of Gaussian noise, ranging from 0% to 200%. Even with up to 200% noise, the network could denoise the images, preserving the integrity of visual information and maintaining essential features of the images. Additionally, the model demonstrated a remarkable ability to generate hierarchical internal representations of the digits, enabling accurate

reconstruction of unseen images and the completion of missing parts in partially occluded images.

Dora et al. [29] proposed a hierarchical neural network trained through the predictive coding strategy to mimic the feedforward and feedback connectivity of the human neocortex, with the primary objective of investigating whether the created neuronal structure can replicate properties of biological neural responses when exposed to visual inputs. The study specifically focused on recognizing behaviors such as orientation and object selectivity without imposing explicit constraints on the training and the network design. The neurons were interconnected by receptive and projective fields, facilitating communication between different hierarchical levels and mimicking the visual processing of the biological system. The model was tested with images of airplanes and cars, and its generalization ability was evaluated on unseen classes. The results indicated that the reconstruction quality slightly decreased when starting the process from higher layers. However, the model could still capture natural image statistics and perform reconstructions that suggest a good generalization ability for unseen categories.

Recently, Yann LeCun [84] proposed a theoretical cognitive architecture model based on fundamental principles of human brain function. The architecture comprises six interconnected modules that enable continuous learning and adaptation to various environments and tasks, namely: configurator, perception, world model, actor, critic, cost, and short-term memory, as shown in Figure 3.3. The configurator acts as a central executive, responsible for setting the parameters and directing the attention of all modules, adjusting them to achieve predefined objectives. The perception module processes sensory inputs and generates a representation of the world's current state, prioritizing the most relevant information for the task at hand through an attention mechanism. At the architecture's core lies the world model, which learns to represent the environment, estimates missing information about the external world, and predicts future states analogous to an internal

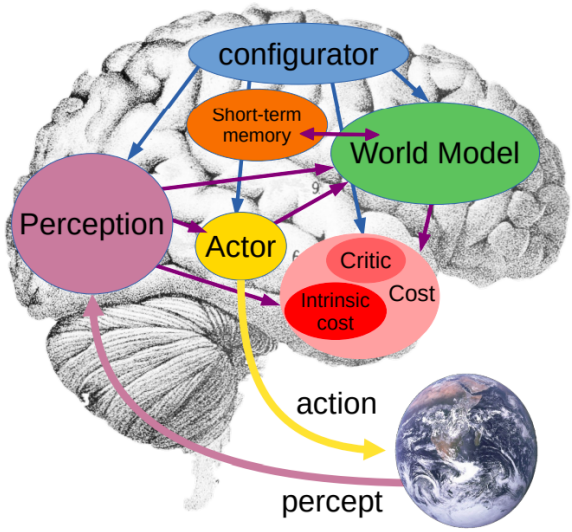


Figure 3.3: A system architecture for autonomous intelligence by Lecun [84]. The architecture comprises the world model, cost, perception, actor, short-term memory, and configurator modules.

simulator. This model is hierarchical and modular, learning to construct representations at multiple levels of abstraction through primarily self-supervised learning.

The critic monitors and evaluates the agent's behavior, helping to adjust decisions and actions based on past states and associated costs. The actor receives the current state generated by the perception module and proposes a sequence of actions to be executed in the environment, guided by configured goals and feedback from the cost module. The short-term memory module stores relevant information about past and present states, as well as the associated costs of these states, facilitating the learning and decision-making process by providing temporal data for the world model. The cost module measures the agent's level of discomfort, considering both predefined costs (hard-coded), such as hunger or pain, and costs learned over time. The agent aims to minimize these costs, adapting its behavior based on information from the perception, critic, and world model modules.

The architecture operates in two distinct modes. In *Mode-1* or the reactive mode, the agent interacts with the environment based on current perceptions and executes actions without explicit reasoning or planning. Sensorimotor interactions continuously update the world model, with incoming information primarily driving this mode. In contrast, *Mode-2*, or the reasoning and planning mode, leverages the predicted states of the world model to learn about new situations and tasks, enabling greater flexibility and adaptation in more complex contexts. In this mode, the agent can plan its actions based on future predictions and explore different solutions to achieve its objectives. The architecture also allows for integrating or alternating these two modes, enhancing its completeness. One of the most significant contributions of this approach is how it articulates the construction of the world model alongside sensorimotor integration. This relationship had been largely unexplored in the literature. Lecun further emphasizes that the world model is highly flexible due to its modular and hierarchical structure, allowing its knowledge to be effectively utilized in reasoning, planning, and adapting to new tasks.

Albus et al. [2] proposed the RCS, a cognitive architecture designed for multi-agent systems. This architecture features modularity and integrates a world model to enable agents to perceive, plan, and act in a coordinated manner within complex environments. A distinctive aspect of RCS is its direct connection between the agent's constructed world model and symbolic representations, an area that remains underexplored in existing literature. This linkage is achieved through attention processes that direct sensors toward relevant regions of the environment and segmentation processes that apply context-sensitive grouping hypotheses to sensory inputs. These groupings are linked to symbolic data structures representing hypothesized entities and events. Geometric and temporal attributes of these groups are computed, and relationships between entities and events are established and maintained. Finally, entities and events are classified and recognized by comparing observed attributes with prototypes stored in the world model. This iterative comparison between world model expectations and sensory observations provides symbolic grounding for agents, ensuring that symbolic representations are anchored to real-world entities and events.

McCall et al. [91] proposed incorporating the concept of predictive coding from the brain into the LIDA cognitive architecture, enabling a more accurate representation of real-world dynamics. This approach utilizes predictive coding to adjust internal repre-

sentations of sensory stimuli, thereby minimizing prediction errors and facilitating the system's learning and adaptation. Such a methodology offers a robust foundation for developing more advanced cognitive systems to process information more efficiently and adaptively. The research emphasizes the importance of integrating cortical learning mechanisms with predictive coding models to understand cognitive functions better and enhance cognitive architectures' performance in complex tasks. The results suggest that combining these methods could significantly advance the creation of cognitive systems that closely resemble human capabilities.

Some approaches construct world models that predict the next state of the environment; however, they do not focus on the theoretical foundations or resemble how humans build these models. These models are developed with an emphasis on predicting future states of the environment, prioritizing the practical application of predictions to solve real-world problems across various domains without exploring advanced theoretical aspects in the model-building process. We observe that these models exhibit only the predictive characteristic of the brain; however, they are constructed separately from the agent's policy or through random actions executed by the agent, and they do not take into account aspects of sensorimotor integration, and neocortical biases, as well as symbolic aspects that are theoretically emphasized as necessary for constructing a human-like world model. In this regard, Wang et al. [150] propose a world model capable of predicting the next state in driving environments to create a robust platform for the development and testing of driving policies in simulated environments that accurately reflect the complexities of the real world. Similarly, Wu et al. [154] apply a world model to enable the simulation of agents within this virtual environment, minimizing contact with real-world environments in training scenarios where agent-environment interaction is costly. Hafner et al. [49] constructed a world model using only the visual sensory modality, separate from the agent's policy. Bruce et al. [14] introduced Genie, a foundational world model trained unsupervised using unlabelled Internet videos. This model can generate various action-controllable virtual worlds based on textual prompts. UniSim is also a unified simulation platform that integrates world models created from extensive training with video sequences to allow the simulation of agents in these virtual environments [158].

Similarly, Ge et al. [38] developed WorldGPT, a world model that acquires an understanding of world dynamics by analyzing millions of videos across various domains. To enhance its capabilities in specialized scenarios and long-term tasks, WorldGPT is integrated into a new cognitive architecture that combines memory offloading, knowledge retrieval, and contextual reflection. This architecture allows WorldGPT to utilize multimodal information, memory, and knowledge retrieval modules to create realistic virtual environments, thereby facilitating the training of virtual agents. To evaluate the performance of WorldGPT, WorldNet was developed, a multimodal state transition prediction benchmark encompassing diverse real-world scenarios. The results demonstrated that WorldGPT can accurately model state transition patterns, confirming its effectiveness in understanding and predicting the dynamics of complex scenarios. Moreover, WorldGPT has shown emerging potential as a world simulator, assisting multimodal agents in generalizing to unknown domains through the efficient synthesis of multimodal instruction instances, which have proven to be as reliable as authentic data for fine-tuning purposes.

This capability expands the applications of WorldGPT across various fields, including the training of virtual agents, the simulation of complex scenarios, and the development of more robust and adaptable artificial intelligence systems.

In this section, we highlight our main contribution: **building a world model through the agent's sensorimotor integration**, where an **embodied agent actively guides the learning process**. Most current approaches focus primarily on the structural aspects of the neocortex and its predictive mechanisms, which operate through error signals propagated by discrepancies between bottom-up and top-down information. These models often remain detached from the agent's corporeal nature and lack exploratory mechanisms, treating world learning as a passive mapping process. However, **the body is fundamental, as we are embodied agents** that use the body to filter and select the information received from the environment. Additionally, many of the most influential works adopt a more theoretical perspective, with architectures not exposed to real training scenarios or realistic environments. A notable example is LeCun's proposed cognitive architecture [84], which shares several similarities with our approach but remains a theoretical model whose all components have yet to be implemented and validated through practical experimentation.

3.2 Intrinsic Motivation for Learning Agents

Intrinsic motivation is a very studied topic in the reinforcement learning field, and a good summary is presented by Barto et al. [11], Aubret et al. [3], Singh et al. [140], and Hao et al. [52]. Initially, intrinsic motivation in reinforcement learning was framed using concepts drawn from various psychological theories, such as emotion, surprise, empowerment, entropy, and information gain, to create effective intrinsic rewards. These ideas reflect the innate drive for exploration and learning within agents that do not rely solely on external rewards. Sequeira et al. [130] explored the hypothesis that affective states encode vital information influencing an agent's learning decision-making process. They proposed that emotional responses can be a form of internal feedback, guiding the agent's learning behavior and helping it focus on tasks that could yield valuable information. This was one of the first attempts to integrate emotion into learning agents, emphasizing the role of affective states in steering decisions and promoting efficient exploration of the environment.

Achiam et al. [1] focused on surprise as an intrinsic motivator. They introduced a novel approach where the agent concurrently learns a probability transition model of a Markov Decision Process (MDP) and its policy. This approach generates intrinsic rewards by measuring the agent's surprise, which is quantified through the Kullback-Leibler (KL) divergence between the agent's learned model of state transitions and the true environment dynamics. Surprise, in this context, indicates how much the agent's current model deviates from reality, thus motivating the agent to explore and correct inaccuracies in its understanding of the environment. The surprise-based model enabled agents to improve their internal world's representations continuously, driving their learning process toward more accurate and efficient predictions.

Similarly, Mohamed et al. [95] expanded computational intrinsic motivation by incorporating empowerment, which is the ability of an agent to control its environment. Their work combined empowerment with variational autoencoders and convolutional neural networks, providing a stochastic optimization framework directly from image pixels. This method allows agents to maximize their control over the environment by predicting the outcomes of their actions, making empowerment an effective intrinsic motivator for agents tasked with complex and dynamic environments. By equipping agents with the capacity to "empower" themselves, the framework encourages agents to explore actions that increase their ability to influence future states.

Expanding on empowerment, Klyubin et al. [79] proposed a formulation of intrinsic rewards grounded in entropy. They viewed empowerment as gaining information from the entropy of actions in a given state. The higher the entropy, the more an agent can influence its environment, promoting the exploration of actions that maximize this influence. Their work formalized the relationship between entropy and empowerment, illustrating how an agent's intrinsic motivation can be driven by a desire to increase its informational control over the environment. This theory highlights how entropy and the potential for self-determined action are potent motivators in an agent's exploration process, enabling it to seek out novel situations where it can expand its control over the environment.

Currently, approaches based on prediction error in the feature space have been extensively explored in the literature. Stadie et al. [143] started their research using the feature space of an autoencoder to measure interesting states to explore. Pathak et al. [105] proposed an approach based on an inverse dynamics model capable of scaling to high-dimensional continuous spaces and minimizing the difficulties of predicting directly in pixels, in addition to ignoring aspects of the environment that do not affect the agent. The approach showed that making predictions directly from the raw sensory space is unfeasible because it is challenging to predict pixels directly. Furthermore, some sensory spaces may be irrelevant to the agent's task. Agents trained with purely intrinsic rewards were able to learn task-relevant cognitive behaviors, demonstrating promising results in sparse environments. Similarly, Taylor et al. [145] proposed an inverse dynamics model to assess the role of sensory space composition in the performance of an intrinsically motivated robotic arm that should manipulate objects on a table. Results showed that the approach works like an "inside-out" curriculum learning. The agent begins to explore its own body first, and only after acquiring knowledge does it explore its surroundings more frequently. Such results explain early motor behavior in infants and reinforce the hypothesis that discovering new patterns drives behavior.

Burda et al. [15] investigated, in various Atari games, how curious agents and different feature spaces alter the results and performance of intrinsic agents. The results showed that: 1) generating the intrinsic reward from prediction error directly from the pixel space is challenging in high-dimensional environments; 2) variational autoencoders (VAEs) are a good summary of the observation but may contain many irrelevant details; 3) random features are fixed and insufficient in several scenarios; and 4) prediction error from inverse dynamic features is currently the best option to guarantee that the learned features contain essential aspects for the agent. Recently, Pathak et al. [106] presented an approach to deal with the challenge of stochasticity of environments. The authors used ideas from

active learning to formulate an approach based on ensemble models.

Houthooft et al. [63] proposed VIME, a curiosity-driven reinforcement learning framework that encourages agents to select actions that maximize information gain regarding the environment's dynamics. This approach is grounded in maximizing the information acquired about the agent's beliefs concerning the environmental dynamics, employing variational inference in Bayesian neural networks to implement this strategy. Experimental results showed that VIME outperformed traditional heuristic-based exploration methods across continuous control tasks, including those with sparse rewards. VIME successfully achieved most of the objectives in environments with sparse rewards, demonstrating that curiosity drove the agent to efficiently explore the environment to obtain sparse rewards, an endeavor that naive exploration failed to accomplish. The results also indicated that even in environments with dense rewards, VIME, when combined with the TRPO algorithm, could avoid premature convergence to suboptimal policies. The authors analyzed the distribution of states visited by naive exploration and exploration utilizing VIME, revealing that naive exploration resulted in a more condensed visitation pattern. In contrast, VIME facilitated broader and more efficient exploration, enabling the agent to reach its objectives more efficiently.

Dean et al. [25] propose a novel form of curiosity in reinforcement learning agents inspired by human exploration through multiple senses. Instead of relying on prediction novelty to guide the agent's exploration within a single modality, they leverage unprecedented multimodal associations to direct the exploratory policy. The authors employ a discriminator trained to differentiate between true audio-visual pairs and misaligned pairs. During training, the agent collects trajectories consisting of visual and auditory feature pairs sequences. The discriminator receives these pairs and learns to predict the likelihood of alignment between them. Correctly aligned pairs are considered positive, while misaligned pairs are negative. The agent receives intrinsic rewards based on the discriminator's uncertainty regarding true aligned pairs. Suppose the discriminator is uncertain about the veracity of a pair. In that case, this indicates a novel association that the agent is unaware of, resulting in a high reward for the agent and encouraging it to explore states that yield audio-visual associations that it has not yet encountered. Experiments conducted in the Atari [12] and Habitat [126] environments demonstrate that this method outperforms traditional curiosity approaches in standard tasks and is an effective strategy to enhance the exploration of reinforcement learning agents.

Kim et al. [77] proposed a curiosity signal called γ-Progress to direct better the agent's exploration towards complex learnable dynamic activities, preventing the curious agent from falling into the "white noise" problem when exploring states that cannot be learned. When using γ-Progress, the agent is encouraged to explore areas of the environment that present significant challenges but are understandable and learnable. The agent builds a world model through the γ-Progress curiosity signal while exploring a room with noisy agents and static and dynamic objects across the floor. The agent's observations are processed by an oracle encoder, which generates a representation oriented to the position of external agents, objects, and the agent's orientation. The results showed that γ-Progress significantly improved the curious agent's sensitivity to noisy elements in the environment. In an experiment with a room containing noisy agents, γ-Progress was the

only agent compared to others that did not get stuck exploring the behavior of these agents during training.

Haber et al. [48] demonstrated that the intrinsically motivated agent exploring the environment to build a world model learned non-trivial behaviors in navigation exploration. They introduced one ball as a simple agent in a maze. They observed that the agent initiated random behaviors in training and posteriorly developed a controlled motion navigating to make more coordinated iterations with objects. The model presents two neural networks, the *world-model* which learns to predict the dynamic consequences of the agent's actions, while the *self-model* learns to predict errors in the agent's world model. The agent then uses the *self-model* to choose actions that it believes will adversely challenge the current state of its *world-model*. This learning occurs through a self-supervised emergent process in which new abilities emerge in developmental milestones, as in human babies. In addition, the agent also learns improved visual encodings in specific tasks, such as detection, location, object recognition, and the prediction of physical dynamics better than other state-of-the-art approaches.

Forestier et al. [35] proposed a competence-based approach to skill learning through agent self-generated goals. This approach is explored in the Intrinsically Motivated Goal Exploration Processes (IMGEP) architecture, where the agent selects its goals as parameterized fitness functions to achieve its objectives. Goals are chosen based on intrinsic rewards that reflect learning progress, and knowledge acquired from one goal can be reused to enhance the agent's performance on others. This approach is an automatic curriculum generator, allowing the agent to discover and refine its skills autonomously. The experiments demonstrated that a humanoid robot could explore multiple goal spaces with hundreds of continuous dimensions and distractions. Without specific target goals, the agent independently auto-determines various goals that are stepping stones for learning more complex skills, such as nested tool use.

Jaques et al. [68] investigate using social influence as an intrinsic motivation reward to enhance learning in multi-agent scenarios. They model social influence as a measure of an agent's empowerment over another, defining a reward function for each agent that is weighted by both the environmental task and the influence one agent exerts on the actions of the other. This approach ensures that agents take action to complete their designated tasks and are encouraged to select actions that significantly impact their counterparts' behavior. As a result, collaborative behaviors naturally emerge without the need for explicit external rewards. Experimental results demonstrate that this approach leads to more efficient learning and the emergence of collective behaviors. The agents develop more collaborative strategies that maximize exploration and task-solving efficiency in shared environments, compared to traditional multi-agent reinforcement learning methods.

In this section, our core distinction lies in the design of a **curious, multimodal, and embodied agent** — a combination still rarely explored in the literature. Our approach is one of the few to explore multimodal curiosity as a mechanism for guiding the agent toward informative states across three sensory modalities. **This capability enables the agent to engage in a richer and more diverse exploration**, allowing us to draw parallels between its behavior and the learning processes observed in children. Infant curiosity is inherently multimodal, as babies and young children explore the world by

actively integrating vision, touch, hearing, and proprioception. However, most intrinsically motivated agents in the literature still operate in an unimodal fashion, limiting their exploration to discrete action spaces in simplified and unrealistic environments. In this context, experiments with an embodied agent are particularly relevant to the field, as the agent's physical structure directly influences how curiosity emerges and evolves.

3.3 Task Adaptation in Robot Manipulation

Research on task adaptation in manipulation robotics can be categorized into three main approaches: (1) utilizing pre-trained representations to enhance the robot's ability to adapt to new tasks or domains; (2) utilizing language to facilitate the learning of abstract concepts and improve the agent's adaptation to novel contexts, and (3) utilizing data augmentation techniques. Many works simultaneously explore more than one of these dimensions, and a limited number of works investigate alternative approaches, such as multi-task learning and the role of sensory perception in robot adaptation and generalization.

Pre-Trained Representations. One of the most significant works on pre-trained representations was proposed by Nair et al. [98], who investigated whether visual representations derived from human videos could be effectively repurposed for robotic manipulation tasks. They noted that many methods to improve robotic representation rely solely on domain-specific datasets, while tests are conducted with out-of-domain data. Believing that training with pre-trained representations from out-of-robotic-domain data could enhance performance and generalization, they developed R3M, a pre-trained visual representation using the Ego4D [45] dataset comprising human videos. R3M achieved a generic representation combining temporal contrastive learning, video-language alignment, and an L1 penalty to encourage sparse and compact representations. The resulting representation was then employed as a frozen perception module for subsequent policy learning. Tests were conducted across 12 robotic manipulation environments, revealing that R3M improves task success by approximately 20% comparing training from scratch and by 10% compared to state-of-the-art visual representations generated by CLIP [114] and MoCo [104, 59].

The R3M framework enabled the Franka Emika Panda robotic arm to learn manipulation tasks in a real and cluttered apartment setting using only 20 demonstrations, amounting to approximately 10 minutes of demonstration data. All experiments aimed to simulate a graduate student performing household tasks in an apartment, such as placing lettuce in a pot in the kitchen, pushing a mug to a target position on a dining table, closing a drawer, placing objects in a drawer, and folding a towel. Concurrently, Xiao et al. [156] investigated using human interaction data to pre-train visual representations for robotic control. However, their learned representation relied solely on video static frames and did not incorporate temporal or semantic information. The main contribution of their approach was demonstrating that using real-world images, such as those from YouTube videos or egocentric recordings, yields better results than using images from ImageNet. Singh et al. [141] proposed a similar image pretraining scheme to learn

representations focusing on semantic and low-level objects' features. To achieve this, they utilized VirTex [28], a pretraining method where an image encoder inputs an image captioning decoder. As this task is semantically rich, the encoder can enhance its visual understanding of the scene by capturing semantic information that can facilitate subsequent robotic tasks.

Shah et al. [131] proposed RRL to explore how features learned by a pre-trained ResNet on a large amount of vision data could effectively enhance the performance of RL agents in control tasks. Instead of training a new network from scratch, RRL harnesses the capabilities of a ResNet trained on an image classification dataset like ImageNet and adapts them for the RL task by replacing the last fully connected layer of the ResNet with a new layer corresponding to the number of possible actions in the RL environment. Subsequently, the RL agent is trained using the representation encoded by the ResNet as input to learn the optimal action policy.

Results demonstrated that RRL offers a promising approach by leveraging powerful pre-trained visual representations and tailoring them to specific control tasks, potentially leading to significant advancements in reinforcement learning. Tests conducted on the ADROIT manipulation suite [115] involving complex manipulation tasks such as object relocation, in-hand manipulation (e.g., pen repositioning), tool use (e.g., hammering a nail), interacting with human-centric environments (e.g., opening a door), showcased the strength of this approach. However, the authors believe that one crucial limitation of this approach is that real-world datasets used to train ResNet features are from human-centric environments. Although we desire robots to operate in similar environments, there are still differences in their morphology and modes of operation. Furthermore, ResNet and similar models acquire features from data primarily composed of static scenes. In contrast, embodied agents require rich features of dynamic and interactive movements.

Pari et al. [103] introduced a framework for visual imitation that distinctly separates representation learning from behavioral learning. Initially, the pre-trained ResNet was tuned offline with data from demonstrations of robotic tasks in order to build the representation vector for the states of the environment. Subsequently, the encoder pre-trained with the task data was used to generate a vector of embeddings for each input; the vector is then compared via nearest neighbors with embeddings from a broad set of robotic manipulation demonstrations to find the demonstration more similar. Their algorithm assumes that demonstrations similar to the system input embedding vector result in similar actions. After the k-nearest neighbors are found, the next action is defined as the weighted average of the actions associated with the k-nearest neighbors. The strength of this approach is that the behavior learning stage is non-parametric and does not require extensive training. However, it requires a database with very varied demonstrations. The results demonstrated that the framework could learn the push, stacking, and door-opening behaviors. Additionally, it was found that this approach is competitive with end-to-end behavior cloning methods. Through a series of generalization experiments, this framework achieved an 80% success rate on doors present in the demonstration dataset and 40% on opening doors in novel scenes.

Chen et al. [16] explored mid-level image representations to create more invariant representations. Instead of training directly with raw pixels, they first extracted repre-

sentations driven by traditional computer vision objectives and used them as observation inputs for RL. The networks performing this extraction of mid-level visual representations are trained asynchronously, meaning they can be trained independently and on a different schedule from RL training. This approach has shown promise in continuous control manipulation. The results indicated that mid-level visual representations provide a helpful way to incorporate invariant features for hard tasks, compared to training from scratch. These representations simplify the learning problem, opening up the possibility of successfully training on more challenging problems that would otherwise fail. Additionally, the representations improve robustness to distribution shifts, both from simulation to the real world and within the simulation itself.

Ma et al. [86] proposed pre-training visual representations on out-of-domain natural and human data as an effective solution to acquire a general visual representation for robotic manipulation and address the challenge of reward specification in the real world. The authors demonstrate that a general reward model can be derived from a pre-trained visual representation by treating representation learning from diverse human video data as an offline goal-conditioned reinforcement learning problem. They propose an innovative approach called Value-Implicit Pre-training (VIP), which uses reinforcement learning itself as a pre-training mechanism for reinforcement learning. VIP trains a dual value function without actions in a self-supervised manner, effectively capturing long-range temporal dependencies and injecting local temporal smoothness into the representations. Trained on a large-scale human video dataset, VIP significantly outperforms previous pre-trained representations in reward-based policy learning paradigms. It achieves success rates of up to 40% in online visual RL. It enables offline RL with few samples in the real world across various robot manipulation tasks with just 20 trajectories.

Given the extensive body of work leveraging pre-trained representations from non-robotic domains to facilitate learning and generalization in subsequent robotic tasks, Parisi et al. [104] investigated the impact of training methods for pre-trained backbones, data augmentation, and feature hierarchies on the learning process. Various pre-trained representations were evaluated, including supervised and self-supervised methods such as Residual Network [60], Momentum Contrast (Moco) [59], Contrastive Language-Image Pretraining (CLIP) [114], random features, and from scratch. The pre-trained backbone models were integrated to represent the environment state. The method involved three distinct and pre-trained convolutional network blocks, each processing a frame of the scene, resulting in the representation of the state across three frames. The output vectors of each convolutional backbone, termed pre-trained visual representations (PVRs), were concatenated and used as input to the policy network. Subsequently, the policy network was trained using the behavioral cloning algorithm based on optimal agent trajectories. Through extensive evaluation across various control domains, including the challenging DeepMind Control domain for object manipulation, the researchers observed that pre-trained visual representations could be as competitive as or even superior to other tested state representations, even when utilizing data outside the robot vision domain.

Despite these results, some authors question the effectiveness of using purely pre-trained backbones from non-robotic environments as frozen representations. Huo et al. [65] contend that using pre-trained self-supervised objectives is ineffective in constructing gen-

eralizable representations for robotic behaviors. According to them, humans generalize by mastering simple perceptual skills from the real world, such as spatial-temporal understanding and hand-object contact estimation, which are pivotal for various everyday tasks [78]. In this way, they proposed a framework called Task Fusion Decoder to learn robot representations guided human-inspired skills. Their framework incorporates cross-attention and self-attention mechanisms to learn these simple perceptual skills from the Ego4D [45] dataset using three representative tasks: object state change classification (OSCC), point-of-no-return temporal localization (PNR), and state change object detection (SCOD). Their framework is a multitask learner designed to work with various vision backbones, such as ResNet [60], ViT [30], R3M [98], and Transformer [148]. Initially, it is trained jointly with one backbone to learn OSCC, PNR, and SCOD tasks simultaneously. Posteriorly, one fine-tuning is realized using behavior cloning to robot manipulation tasks with few demonstrations. The experimental results showed that this fine-tuning strategy improved robotic performance in manipulation tasks from 2% to 15%, depending on the task.

Yen-Chen et al. [159] investigated methods for transferring latent features from visual to policy models. They observed that simply transferring these features results in poor exploration, as the randomly initialized policy head still explores environments randomly. To address this issue, they proposed a two-stage learning approach. First, the visual backbone and a vision head are trained for primitive visual tasks such as edge detection, object detection, background segmentation, and object center detection. Then, the system is trained for tasks where the robot has active vision and can manipulate objects. This approach was inspired by the idea that humans benefit from perceptual visual cues about the world, such as object structure, to facilitate exploration and learning. The results show an improvement of over 10% in agent generalization in grasping and suction tasks when the system first learns any visual cue tasks through passive vision.

Language. Recently, some researchers have explored the integration of language as a facilitator of robot agents' adaptation, as it stimulates the learning of more abstract concepts in the agent's knowledge representation. Following this, Shridhar et al. [137] propose the CLIPORT, a language-conditioned imitation learning framework for robot visual manipulation tasks that blends manipulation skills with reasoning about abstract concepts through language. The framework integrates the broad semantic understanding of CLIP [114] with the spatial precision of the Transporter [162] to strengthen the robot with semantic and spatial pathways for visual manipulation. The CLIPORT is a two-stream end-to-end framework that can tackle various language-specified tasks on a flat surface, from packing unknown objects to folding fabrics, without explicit representations of object poses, instance segmentations, memory, symbolic states, or syntactic structures. Experiments conducted in simulated and real-world settings demonstrate the efficiency of the approach in data and few-shot settings, effectively generalizing to seen and unseen semantic concepts. Furthermore, a multi-task policy is learned for ten simulated and nine real-world tasks, showing superior or comparable performance to single-task policies.

Silva et al. [138] propose LACON-LEARN, an architecture to enable multi-task learning agents to understand and execute tasks specified through natural language commands and attention. This approach uses natural language sequences into semantically task-

relevant goal embeddings and attention to allow agents to attend to salient components of language commands to activate task-relevant skills. This architecture learns a policy directly from task information and activates sub-components of the policy according to goal embeddings. This model enables robots to learn from language commands and corrective feedback through imitation learning, achieving significant improvements in zero-shot skill transfer compared to baselines. The approach sets a new state-of-the-art for zero-shot task success and few-shot knowledge transfer, showcasing its potential for enabling robots to quickly adapt to new tasks in real-world scenarios.

Shao et al. [133] proposed Concept2Robot, a framework for learning by demonstration that leverages large datasets of human videos performing manipulation actions. The work distinguishes itself from previous research by not solely focusing on lexical concepts corresponding to words in natural language but specifically addressing the acquisition of manipulation concepts. The proposed framework involves learning single-task policies through reinforcement learning. Subsequently, a multi-task policy is trained through imitation learning to mimic all single-task policies. The framework takes as input a natural language instruction describing the task along with an RGB image of the initial scene. These inputs are processed by a semantic context network to merge natural language information with the robot's visual perception, resulting in a joint representation of the desired task. Next, the task representation is fed into a policy network, synthesizing the robot's motion. Finally, the robot executes the trajectory using Operational Space Control [80].

During the initial training phase, manipulation concepts are acquired from human demonstrations by assessing the degree of similarity between a robot's execution and that of a human performing the same task. This evaluation employs an action classifier trained on videos depicting human activities [43]. The classifier's outputs serve as rewards for reinforcement learning, which learns a policy for each task. Subsequently, the system undergoes imitation learning to acquire a multi-task policy based on the single-task policies learned through reinforcement learning. The outcome is a multi-task policy capable of receiving a new natural language instruction and an environmental image and executing the desired task by drawing upon the knowledge gained from previously learned 78 tasks.

Jiang et al. [69] introduced the VisuoMotor Attention (VIMA), a model to improve out-of-distribution generalization through a concise multimodal representation. The model utilizes a multimodal prompt to collect linguistic task instructions and frontal and top-view images of the scene. To construct an object-centered representation, they employ a pre-trained R-CNN mask to segment objects in the scene. Subsequently, all inputs are tokenized to self-attention layers in Transformer encoder-decoder architecture. These choices simplify and concisely represent the environment, facilitating generalization to new tasks and scenarios. The robot controller is then conditioned on the input's multimodal elements and the history of previous iterations. Training is conducted using imitation learning through the behavioral cloning algorithm, minimizing the negative log-likelihood of predicted actions compared to actual actions observed in the dataset. To generate the imitation training loss, they constructed an oracle that generates desired movements for the robot to learn to imitate.

To assess VIMA's generalization capability, researchers devised an evaluation protocol

named VIMA-BENCH, scrutinizing the model's performance across various generalization levels, from random object placements to entirely novel tasks. The findings unveiled that the imitation-learned policy proficiently extends to fresh scenarios and tasks, with language playing a pivotal role in this accomplishment, enabling task disentanglement. Multiple VIMA models were trained with diverse capacities ranging from 2M to 200M parameters, showcasing VIMA's superiority over previous methods across all generalization tiers, even with limited training data. Furthermore, simulation sets, training datasets, algorithm implementation, and pre-trained model checkpoints were provided to foster reproducibility and future advancements in the field.

Zhu et al. [164] introduced a novel language-conditioned robotic manipulation framework, RFST, drawing inspiration from the human cognitive theory of fast and slow thinking [22]. The aim is to condition policy learning on language and proper switching between two different systems for tasks involving reasoning in robotic manipulation. This framework categorizes tasks and makes decisions in two systems, depending on the complexity of user instructions. The RFST includes an instruction discriminator to determine which system should be activated based on user instructions and a slow-thinking system composed of a vision-language model and policy networks. This model enables the robot to recognize user intent or perform reasoning tasks. The study involved the construction of a dataset with real-world trajectories covering a range of actions, from spontaneous impulses to tasks requiring deliberate contemplation. Experimental results in simulations and real-world scenarios demonstrate that RFST effectively manages complex tasks requiring intent recognition and reasoning.

Ghosh et al. [146] utilized Transformers and multimodal prompts to construct Octo, a transformer-based diffusion policy pre-trained on 800 thousand robot trajectories from the Open X-Embodiment dataset. Octo demonstrates high flexibility, supporting multiple RGB camera inputs, diverse robot arms, and instructions through language commands or goal images. The primary focus is enabling effective fine-tuning of Octo for new sensory inputs, action spaces, and morphologies, using only a small dataset from the target domain and accessible computational budgets. This effort aims to overcome the limitations of existing generalist robot policies by providing open-access resources for training, using, reproducing, and fine-tuning Octo models. The ultimate goal is transforming robotics learning research, like the widespread adoption of large pre-trained language models in natural language processing.

Recently, Kalithasan et al. [75] introduced a framework that employs LLMs to enhance neuro-symbolic learning, significantly improving the agent's inductive knowledge representation of the world. The approach consists of three main steps: sketch, plan, and generalize. In *sketch*, a natural language instruction and its corresponding demonstration are provided. The Task Generator module, composed of GPT-4, translates the instruction into a programmatic signature specifying the name and parameters of the concept to be learned. For instance, if the demonstration involves violet cubes on a table, the generated programmatic signature might resemble FILTER(magenta, cubes). In the *plan*, the demonstration's visual encoder, the programmatic signature, and a library of previous programs are used via the Monte Carlo Tree Search algorithm to learn the new concept's representation. The result is a new program for action execution, which feeds into a neu-

ral network responsible for generating the agent's actions. The rewards obtained in the environment are used as feedback to optimize the algorithm. Finally, in the *generalizing* step, the created program is distilled to demonstrate a highly generalized program. These three steps, facilitated by using an LLM, enable the learning of simple and increasingly complex symbolic representations.

Multi-Task Learning. Singh et al. [139] introduced an approach for learning generalizable sensorimotor control policies through multitask learning. The method, named Generalized Policy Learning with Attentional Classifier (GPLAC), integrates iterative learning from task demonstrations with passive learning from weakly labeled data classification. GPLAC employs two models sharing parameters during training: a convolutional model to predict agent actions and another to classify weakly labeled images into binary classes. During training, the robot receives both demonstrations on how to grasp a mug in a specific environment and a set of images collected during robotic interaction in other environments, where the label indicates only the presence of a mug in the image. The robot learns a sensorimotor policy applicable in new environments with this information. A spatial attention layer is utilized to handle irrelevant distractors in the scene, facilitating generalization in the presence of domain shift. Experiments conducted in simulated manipulation tasks and with a real robotic manipulator underscore the significance of the spatial attention mechanism and multitask training with weakly labeled data to achieve substantial generalization with minimal interaction.

Active Perception. Zaky et al. [160] focused on exploring active perception to enhance and refine the environment representations learned by the robot. They conducted experiments in a robotic manipulation environment comprising two robots. The first manipulator was tasked with physically manipulating objects arranged on a tray. In contrast, the second manipulator, referred to as the *head*, was equipped with a camera attached to its wrist, enabling it to adjust the camera's viewpoint. This configuration with two manipulators allowed the system to perform two tasks simultaneously: physically manipulating objects and capturing different scene perspectives. The researchers adopted a multimodal system that combines images and proprioceptive inputs from both robots into a single representation. The results revealed that when applying the proposed model to a simulated grasping task with a 6-degree-of-freedom action space, the active model outperformed its passive fixed-camera counterpart, achieving an 8% improvement in performance. Additionally, the active model was four times more sample-efficient than conventional deep Q-learning algorithms.

Data Augmentation. Xie et al. [157] conducted experiments to comprehend the factors contributing to the generalization difficulty of vision-based robotic manipulators trained through imitation learning. Utilizing a real robotic manipulator, they explored over 20 test scenarios with diverse lighting conditions, distractor objects, backgrounds, table textures, and camera positions. Additionally, they developed 19 simulated tasks equipped with 11 additional configurable environmental factors. The findings revealed that most pairs of factors did not exhibit a significant compounded effect on generalization performance, indicating that it is not inherently more challenging to generalize to new table textures and objects simultaneously than to new table textures alone. They also observed that random crop augmentation is a lightweight approach to enhance a

generalization across spatial factors (e.g., camera positions) and non-spatial factors (e.g., distractor objects and table textures). Furthermore, training the robot with visual data from tasks outside the domain can dramatically enhance generalization. For instance, training on tasks such as opening a refrigerator or operating a cereal dispenser can significantly improve performance in object selection on a tabletop.

Jang et al. [67] investigated how data collection diversity can enhance a robotic system's ability to learn and generalize across a wide range of real-world tasks. They developed a flexible imitation learning system capable of learning from expert demonstrations and interventions to correct the robot's current policy. Data collection was conducted using a teleoperation system connected to the robot's onboard computer, enabling the operator to control the robot with two manual controllers and real-time third-person vision. Tasks were performed in an environment containing a table with 6 to 15 household objects in various random poses. Initially, expert demonstrations were collected for 100 pre-specified tasks, covering nine underlying skills, such as pushing and pick-and-place. Sequentially, a multi-task policy was learned exclusively from expert data, consisting of human videos and language commands of the tasks.

During the policy deployment, data collection continued in *shared autonomy* mode, where the robot attempted tasks while supervised by a human. The human can intervene using an override switch to correct the robot's execution. The resulting dataset included 25,877 robot demonstrations, encompassing demonstrations solely by experts and those collected during policy deployment iterations. Language commands and demonstration videos were encoded into task embeddings using separate encoders in the training process to train the model. A pre-trained multilingual sentence encoder was employed for language commands, while a convolutional neural network based on ResNet-18 was used for videos. These encoders were trained end-to-end with paired human video data and corresponding robot demonstrations. An auxiliary language regression loss was also introduced to enhance semantic alignment and task generalization. The results demonstrated that the system could execute 24 unseen manipulation tasks with a success rate of 44%.

Hansen et al. [51] introduced Soft Data Augmentation (SODA), an approach to enhance the generalization capability of vision-based reinforcement learning methods. In contrast to previous approaches that directly learn from augmented data, SODA decouples the data augmentation step from the policy learning process. It exclusively utilizes non-augmented data for policy learning while conducting auxiliary representation learning using augmented data. SODA aims to maximize the mutual information between the latent representations of augmented and non-augmented data, thereby facilitating generalization. Experimental results demonstrated significant improvements in sampling efficiency, generalization, and training stability compared to state-of-the-art vision-based RL methods. The findings underscore that SODA considerably outperforms previous approaches regarding generalization to visually diverse environments not observed during training.

Zhan et al. [163] introduced the Framework for Efficient Robotic Manipulation (FERM), which combines ideas from contrastive pre-training, data augmentation, and demonstrations to enable robotic agents to learn skills directly from pixel inputs in a data-efficient manner, requiring less than an hour of training. FERM leverages recent advances in un-

supervised representation learning and data augmentation, which have proven effective in simulated and video game robotic environments. The approach involves collecting just ten demonstrations stored in a replay buffer. The convolutional encoder's weights are initialized with unsupervised contrastive pre-training on demonstration data. Finally, an off-policy RL algorithm is trained with augmented images using online data collected during training and the initial demonstrations. This methodology enabled learning optimal policies in 6 diverse manipulation tasks in just 15-50 minutes of total training time for each task. The approach also facilitated efficient training on real robotic hardware, whereas previous related approaches successful in simulation failed to learn robust policies on real robots.

Hansen et al. [50] identified that data augmentation in reinforcement learning induces instability in off-policy algorithms. This instability arises primarily due to the indiscriminate application of data augmentation, which results in high-variance Q-targets. Moreover, estimating Q-values exclusively from augmented data leads to over-regularization. To address these issues, the authors proposed SVEA: Stabilized Q-Value Estimation under Augmentation, a simple and effective framework for data augmentation in off-policy RL that significantly enhances the stability of Q-value estimation. The method involves applying data augmentation only to the Q-value estimation of the current state, formulating a modified Q-objective that optimizes Q-value estimation over both augmented and unaugmented copies of observations, and optimizing the actor strictly on unaugmented data. The authors perform an extensive empirical evaluation on several tasks, including the DeepMind Control Suite and robotic manipulation tasks. The results show that the proposed method significantly improves the Q-value estimation under a set of strong data augmentations and achieves sampling efficiency, asymptotic performance, and competitive generalization or better than previous methods in all tasks considered. Furthermore, the method is scalable to RL with Vision Transformers (ViT), being especially effective in avoiding overfitting in ViT-based architectures.

In this section, our **key distinction** lies in **developing an agent for manipulation tasks grounded in cutting-edge world model theories** while integrating a bioinspired perspective into the field. While most current approaches rely heavily on imitation learning — resulting in limited interaction between robots and their physical environments — our framework challenges this approach by emphasizing the vital need for iterative engagement with the real world. Existing methods often demand **massive datasets of natural and human environments**, which are data-intensive and fail to address the robot's embodied experience. Pre-trained encoders, in particular, have not delivered significant improvements and are hindered by their inability to build effective representations within the robot's body. Similarly, data augmentation techniques in the field are falling short. Moreover, most current research limits itself to simple robotic arms, further underselling the potential for more sophisticated manipulative behaviors. In contrast, **our framework embraces the complexity of humanoid robots**, adapting them to their inherent physical constraints while unlocking the full potential of complex, real-world manipulation.

Chapter 4

Materials and Methods

This chapter presents the materials and methods employed to develop this work. The environments used to train our agents will be detailed, including the metrics to evaluate the quality of the results. We specify the software and hardware technologies used in the experiments. Also, we present the methodology employed to build our models and reach our results.

4.1 Materials

This section describes the materials and resources that supported the development of our experiments. We detail the robotic platform selected for the tasks, the simulation environments designed for training and evaluation, and the technological infrastructure utilized.

4.1.1 Simulator and NAO Robot

To validate our work, we chose the NAO humanoid robot [132], as shown in Figure 4.1, because it mimics humans with similar sensory modalities and a highly articulated torso, head, arms, and legs. The NAO robot was launched in 2006 by SoftBank Robotics to interact with humans naturally using its body language, voice, and a wide array of sensors, including touch, image, and sound. NAO is 58 cm tall, weighs 4.3 kg, and has 25 motorized joints, providing a wide range of movements and expressions, including the ability to walk, raise its arms, move its head, and tilt its torso. The joints are classified into four types: shoulder, elbow, wrist, and hip joints. Each joint has one or more degrees of freedom (Table 4.1), allowing the NAO to perform various movements precisely. The shoulder and elbow joints have two degrees of freedom each, allowing it to move its arms in various positions and orientations. The hip joints are also highly flexible, allowing it to walk quickly and safely on different surfaces. These features make it ideal for testing in object manipulation tasks. Moreover, this robot is programmable in various languages, such as Python and C++, allowing developers to create customized applications and behaviors.

We use the NAO robot in the Coppelia Simulator v4.3 [118], which is a robotics simulator to create and simulate 3D environments for robots, autonomous vehicles, and other applications. It is based on a distributed control architecture, where each object can

Figure 4.1: An overview of the NAO humanoid robot. It showcases highly articulated limbs and torso, enabling natural interaction and movement emulation. Equipped with a range of sensors, NAO is a versatile platform for research in human-robot interaction and cognitive robotics.

be individually controlled via C/C++, Python, Java, and Lua. CoppeliaSim presents an easy and intuitive graphical user interface, as shown in Figure 4.2, simplifying modeling, programming, and simulating robotic applications. It also offers many resources, including pre-existing robot model libraries, an integrated programming language, and an external API. Furthermore, CoppeliaSim offers various functionalities that make the robotic simulation even more realistic. For example, it is possible to simulate sensors (i.e., cameras, force sensors, proximity sensors) and the behavior of actuators, such as motors and pneumatic actuators. These features enable roboticists to test and validate robotic systems before implementing them in physical hardware, saving time and money in robotics project development. CoppeliaSim is one of the most versatile simulation tools available today.

To interface between Coppeliasim and our models, we utilized PyRep[1], a Python toolkit developed in 2019 by researchers from the Robotics and Artificial Intelligence Laboratory (RAIL) at Carnegie Mellon University. PyRep offers an accessible Python interface tailored for controlling the CoppeliaSim simulation environment, thereby simplifying the development of robot control algorithms. PyRep furnishes a straightforward programming interface for CoppeliaSim control and interaction with simulated objects, including robots, sensors, and actuators. Moreover, it facilitates distributed experiment execution across multiple computers and provides visualization and debugging capabilities. By integrating PyRep with CoppeliaSim, developers can craft highly customized simulation environments and conduct large-scale robot simulation experiments, replicating real-world conditions closely. PyRep boasts compatibility with diverse robots and sensors, encompassing wheeled robots, robotic arms, drones, and image and depth sensors. Furthermore, it supports creating and customizing simulation environments, empowering

[1] https://github.com/stepjam/PyRep

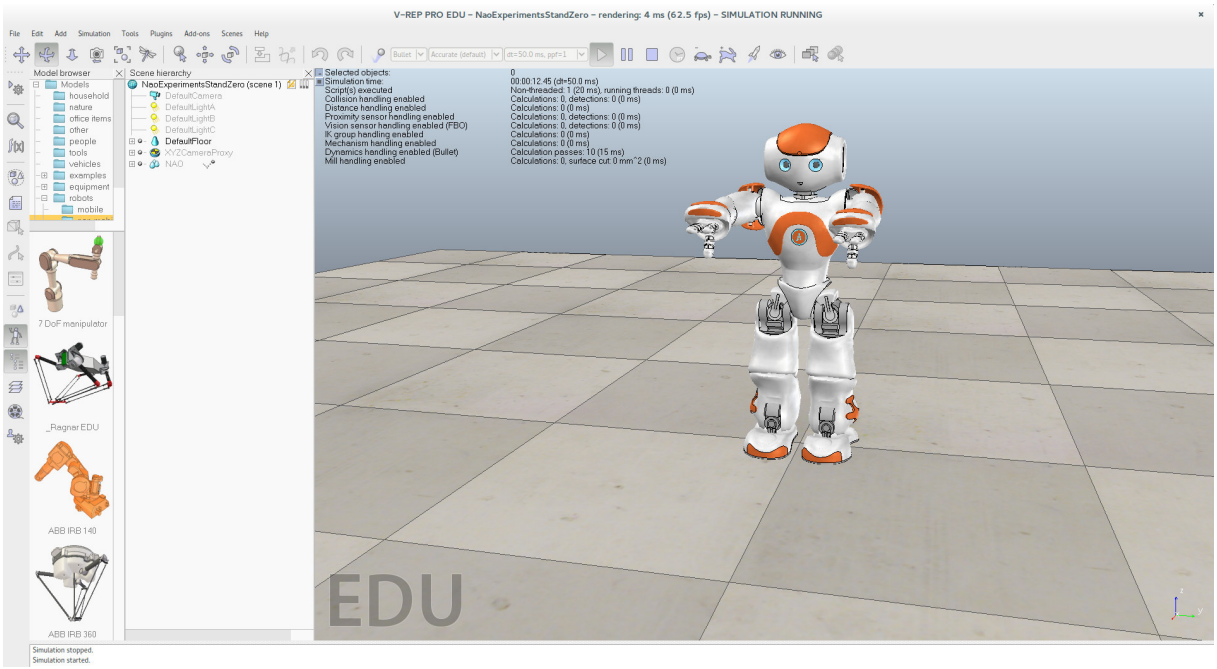


Figure 4.2: The NAO humanoid robot simulated within the CoppeliaSim environment. CoppeliaSim provides a simple and friendly graphical interface for testing and development, allowing researchers to explore various robotic behaviors and interactions in simulated environments.

the modeling of real-world scenarios and the instantiation of specific robotic setups.

Since its initial release, PyRep has been widely adopted by robotics researchers and engineers worldwide who appreciate the Python interface's ease of use and flexibility. The project is constantly evolving, with new features and improvements regularly added, making it one of the best options for robotics simulation. In our work, the integration of PyRep into CoppeliaSim favors agent training using the Pytorch packages provided by Python to build machine learning models. It allows the parallelization of multiple environments, facilitating the collection of trajectories for model training.

4.1.2 Experimental Environment

In our experiments, we built a robotic manipulation environment using the NAO humanoid robot, seated on a chair facing a table with simple objects such as cubes, spheres, and cylinders of varying colors and sizes, as shown in Figure 4.3. The NAO's arms possess full access to the tabletop, and within the simulation, we retain the flexibility to add, remove, or modify any object within the scene. The robot's legs and torso are fixed in a default position, while control is exerted over the arms, head, hands, and fingers, corresponding to 28 joints. Each joint is bounded by maximum and minimum angular amplitudes, detailed in Table 4.1. The NAO's hand comprises three fingers, each capable of movement across three joints, except the thumb, which operates across two joints.

In our experiments, we utilized a variety of sensory inputs to represent the state of the environment. These inputs are scene images, detected collisions between the robot's fingers and objects, and proprioceptive feedback from the robot. The PyRep library

Table 4.1: Joint limits of the NAO robot.

Joint name	Type	Motion	Range (degrees)
HeadYaw	Head joints	Head joint twist (Z)	-119.5 to 119.5
HeadPitch	Head Joints	Head joint front and back (Y)	-38.5 to 29.5
LShoulderPitch	Left Arm	Left shoulder joint front and back (Y)	-119.5 to 119.5
LShoulderRoll	Left Arm	Left shoulder joint right and left (Z)	-18 to 76
LElbowYaw	Left Arm	Left shoulder joint twist (X)	-119.5 to 119.5
LElbowRoll	Left Arm	Left elbow joint (Z)	-88.5 to -2
LWristYaw	Left Arm	Left wrist joint (X)	-104.5 to 104.5
LHand	Left Arm	Left hand	Open and Close
RShoulderPitch	Right Arm	Right shoulder joint front and back (Y)	-119.5 to 119.5
RShoulderRoll	Right Arm	Right shoulder joint right and left (Z)	-76 to 18
RElbowYaw	Right Arm	Right shoulder joint twist (X)	-119.5 to 119.5
RElbowRoll	Right Arm	Right elbow joint (Z)	2 to 88.5
RWristYaw	Right Arm	Right wrist joint (X)	-104.5 to 104.5
RHand	Right Arm	Right hand	Open and Close
LHipYawPitch	Pelvis	Left hip joint twist (Y-Z 45°)	-65.62 to 42.44
RHipYawPitch	Pelvis	Right hip joint twist (Y-Z 45°)	-65.62 to 42.44
LHipRoll	Left leg	Left hip joint right and left (X)	-21.74 to 45.29
LHipPitch	Left leg	Left hip joint front and back (Y)	-88.00 to 27.73
LKneePitch	Left leg	Left knee joint (Y)	-5.29 to 121.04
LAnklePitch	Left leg	Left ankle joint front and back (Y)	-68.15 to 52.86
LAnkleRoll	Left leg	Left ankle joint right and left (X)	-22.79 to 44.06
RHipRoll	Right leg	Right hip joint right and left (X)	-45.29 to 21.74
RHipPitch	Right leg	Right hip joint front and back (Y)	-88.00 to 27.73
RKneePitch	Right leg	Right knee joint (Y)	-5.90 to 121.47
RAnklePitch	Right leg	Right ankle joint front and back (Y)	-67.97 to 53.40
RAnkleRoll	Right leg	Right ankle joint right and left (X)	-44.06 to 22.80

facilitated the transmission of all sensory signals and actions to the robot's joints at each simulation step. The experimental setup enabled comprehensive training of the robot for manipulation tasks spanning a wide range of complexities, including grabbing, lifting, touching, grasping, and assembling.

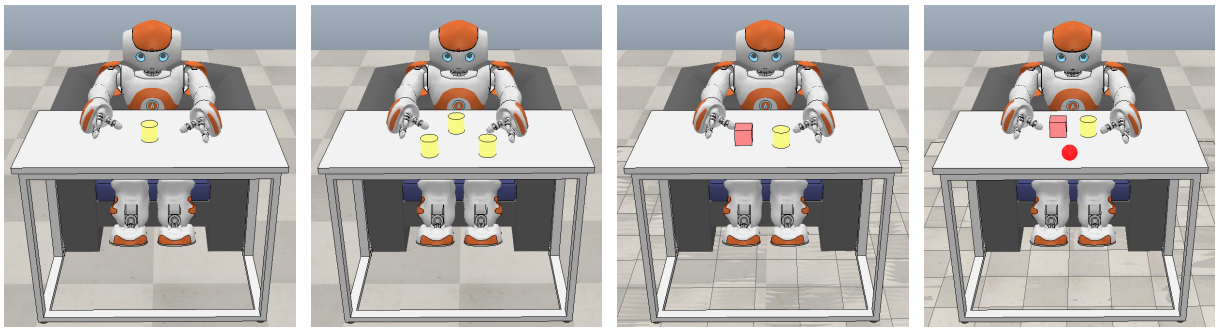


Figure 4.3: Scene samples from the simulator with different objects, such as cubes, spheres, and cylinders.

4.1.3 Metrics

We evaluate our agent performance using the following metrics:

- **Mean episodic return:** The total accumulated reward obtained by an agent throughout an episode, averaged across N episodes:

$$\text{Mean episodic return} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T_i} r_t^{(i)} \tag{4.1}$$

  where $r_t^{(i)}$ is the reward at time t in episode i, and T_i is the length of episode i.

- **Mean interaction intensity**: The total accumulated of tactile interactions (e.g., fingertip-object touches) by agent throughout an episode, averaged across N episodes:

$$\text{Mean interaction intensity} = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=0}^{T_i} \text{touch}_t^{(i)} \tag{4.2}$$

  where $\text{touch}_t^{(i)}$ is 1 if contact occurs at time t in episode i, and 0 otherwise.

- **Qualitative metrics:** These include visual inspection and human judgment of emergent behaviors during both training and testing phases. Evaluators may analyze video frames that display motor coordination or object interactions throughout the training and testing processes.

- The number of neural network parameters.

We evaluate the accuracy of the agent's world model prediction using the following metrics:

- **Mean Squared Error (MSE) [41]:** This metric quantifies the average squared difference between the predicted values and the ground truth. It emphasizes larger errors more strongly and is widely used for measuring prediction accuracy in regression tasks:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.3}$$

  where y_i is the ground truth and $\hat{y}_i$ is the prediction.

- **Sum of Absolute Differences (SAD) [41]:** This metric computes the total absolute deviation between the predicted values and the ground truth. It provides a direct measure of discrepancy that is less sensitive to outliers than MSE:

$$\text{SAD} = \sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.4}$$

  where y_i is the ground truth and $\hat{y}_i$ is the prediction.

- **Structural Similarity Index (SSIM) [151]:** This metric evaluates the perceptual similarity between two images by comparing local patterns of pixel intensities normalized for luminance, contrast, and structure. It is beneficial for image prediction tasks due to its alignment with human visual perception:

$$\mathrm{SSIM}\left(\mathbf{x},\mathbf{y}\right) = \left[l\left(\mathbf{x},\mathbf{y}\right)\right]^{\alpha} \cdot \left[c\left(\mathbf{x},\mathbf{y}\right)\right]^{\beta} \cdot \left[s\left(\mathbf{x},\mathbf{y}\right)\right]^{\gamma}, \tag{4.5}$$

  where $l(\cdot)$, $c(\cdot)$, and $s(\cdot)$ represent the luminance, contrast, and structure similarity maps between the ground truth x and the prediction y, respectively. The positive exponents α, β, and γ control their relative importance in the final similarity score.

- **L1 Distance [41]:** This metric measures the average absolute difference between predicted and true values. It is a robust alternative to MSE and is less influenced by large outliers:

$$\text{L1 Distance} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.6}$$

  where y_i is the ground truth and $\hat{y}_i$ is the prediction.

4.1.4 Software and Libraries

The code was developed in the Python programming language. The main libraries used are:

- Matplolib - for data ploting;
- Pandas - for data manipulate;
- Numpy - for matrix calculations;
- PyTorch 1.3.1 - for building neural networks;
- Scipy - for mathematical calculations;
- Scikit-Learn - for machine learning.

4.1.5 Hardware Specification

The hardware used for training the models has the following specifications:

- Motherboard: Asus Rog Strix Z790-A Gaming;
- CPU: Intel Core i7-13700KF @ 5.4GHz;
- RAM: Corsair DDR4 1x32 Gb @ 2666MHz;
- GPU: Nvidia RTX 4090 with 24Gb, and Cuda v12.2;
- Disk: Seagate Barracuda 2Tb;
- Operating System: Ubuntu v22.04.

4.2 Methodology

The methodology adopted in this work is divided into three main stages: (i) simulator and environment setup; (ii) development, training, and evaluation of the DreamerRL framework to enable the autonomous development of the agent; and (iii) task adaptation evaluation.

In the first stage, we defined and configured the training environments within CoppeliaSim. Several object manipulation environments were created manually and, in some cases, using PyRep's programming capabilities. All environments include an NAO humanoid robot seated in front of a table with objects available for manipulation. The types, arrangements, and positions of these objects vary depending on the objective of each experiment, but all were placed within the reachable workspace of the robot's arms. We used a variety of objects — including cylinders, cones, cubes, and spheres — with different colors, sizes, and quantities. The initial joint positions of the robot were set to the same configuration in all environments to ensure consistency across experiments.

In the second stage, the focus was on developing the DreamerRL framework for creating a complex robotic agent capable of autonomously developing motor skills. Initially, all the components of the framework were implemented using simple elements. These components were gradually enriched as the experiments progressed to enhance the agent's autonomous development. In all experiments conducted in this stage, the agent was trained using reinforcement learning, specifically employing PPO algorithm, to learn a world model. In the first experiment, the agent's task was to understand how the environment functions by predicting only the next visual observation. In this case, curiosity was single-modality, and the agent's actions were driven solely by the goal of discovering novel and challenging visual states. The architecture used to build the world model and generate motor actions consisted of a monolithic and hierarchical structure based on classical artificial neural networks, such as convolutional neural networks (CNNs) and linear layers.

In the subsequent experiments, the complexity of the framework components was progressively increased. The agent's predictions became multimodal, and the system incorporated new sensors and actuators, allowing for greater environmental immersion. Curiosity also evolved into a multimodal mechanism, increasing the complexity of the policy to be learned. Next, we evaluated how multimodality and the increased sensorimotor complexity influenced the agent's ability to develop autonomously through qualitative and quantitative comparisons between the current agent and the baseline agent from the initial experiments. Furthermore, we drew parallels between the agent's emergent behaviors and early childhood development, discussing observed similarities and differences throughout training. Finally, following the training phase, we assessed the generalization capability of the agent's learned world model by exposing it to novel, previously unseen environments and analyzing the quality of its predictions in these test environments.

In the final stage, we investigated the agent's ability to transfer and adapt previously acquired skills to a new task. This phase focused on testing how well an intrinsically trained agent initially developed to learn a world model could be adapted to solve a downstream task with an extrinsic objective. Specifically, we created the CapturingBall

task, in which the agent had to intercept a moving ball using a predefined extrinsic reward signal. To evaluate this, we froze the previously trained world model weights and fine-tuned only a few layers of the policy network to adapt it to the new task. As a baseline comparison, we trained a separate agent from scratch using a standard extrinsic reward-driven reinforcement learning approach for the same task. The performance of both agents was compared using the mean and standard deviation of the extrinsic rewards obtained, allowing for a robust statistical analysis.

Subsequently, we enhanced the agent's architecture in task adaptation by integrating neocortical-inspired structural biases, such as modularity, sparsity, and hierarchical organization with bidirectional information flow. We first trained a new agent to learn a world model using this enriched architecture and then transferred its policy to the Capturing-Ball task through fine-tuning, as done previously. This setup allowed us to investigate whether such biases could improve the agent's ability to adapt. Finally, we conducted a comparative analysis between the fine-tuned modular policy and the monolithic policy. We evaluated their performance in the new task to assess whether incorporating structural biases provided measurable benefits regarding adaptability and learning efficiency.

Finally, we discussed the broader implications of our findings in the context of autonomous agent development and adaptability reporting main results.

Chapter 5

DreamerRL

This chapter presents the **DreamerRL**, a framework to learn how the world works. The initial sections present the construction and validation of our approach (Sections 5.1 to 5.2). We detail the initial construction of a baseline model, including the main components and the integration of curiosity into the learning process. Subsequently, we assess the impact of multimodality on curiosity and exploration, discussing how different modalities enrich the curiosity-driven learning process (Section 5.3). Additionally, we incorporate further sensing aspects to enhance embodiment and facilitate more complex and realistic interactions with the environment. This evolution will demonstrate how various embodied elements influence autonomous development and adaptability in our artificial robot agent (Section 5.3).

Posteriorly, we analyzed if our approach constructed a world's internal representation that enabled the robotic agent to learn abstract concepts about how the world functions. To this end, we conducted generalization tests in novel scenarios distinct from those encountered during intrinsic training (Section 5.4). Finally, we conduct adaptation tests by applying our intrinsic agent to a new extrinsic task and comparing its performance with an identical agent trained exclusively through extrinsic rewards (Section 5.5). In this point, we also examined modularity, hierarchy with bidirectional flow, and sparsity biases in the adaptation's quality.

5.1 The Framework Definition

In designing our framework, we developed a structure inspired by key research on the theoretical aspects of world models and the elements that enhance their construction, contributing to human intelligence development [55, 54, 57, 62, 6, 82, 71, 70, 136]. Our framework is designed to enable autonomous development and task adaptation in robotic agents through the integration of embodied perception, motivation (intrinsic and extrinsic), neocortical circuit, and sensorimotor integration to construction and continuous refinement of an internal model of the world world, which allows the agent to predict, evaluate, and act based on internal representations rather than relying solely on external supervision. The framework is composed of six interconnected modules (Figure 5.1):

1. **Perception Module:** The perception module processes raw sensory data from the

external environment, transforming it into structured representations. An **embodied agent** provides the raw sensory data with sensory modalities and a body that is more similar to humans to demonstrate the importance of body restrictions in building world models and autonomous development. These representations serve as input for the internal environment, particularly the world model, extrinsic motivation, intrinsic motivation, and critic.

2. **Internal Environment (World Model):** The internal environment is the cognitive core of the agent and contains the world model, a predictive system responsible for learning the latent dynamics of the environment and future predictions about it. By observing sequences of actions and sensory states, the world model enables the agent to simulate future states of the real world internally. The world model allows the agent to act reactively and based on imagined consequences of actions. Importantly, the world model can implement neocortical circuit-inspired structures, including modularity, hierarchy, and sparsity, providing the architectural flexibility necessary for adaptive internal representations.

3. **Intrinsic Motivation:** This module evaluates the agent's internal states to generate intrinsic rewards that guide exploration. These rewards can be trainable or hard-coded and are designed to capture internal drives. Our work focuses only on curiosity-driven rewards derived from novelty and unpredictability in the world model's predictions mainly because these self-supervised signals encourage the agent to attend to parts of the environment that are still uncertain or poorly understood, supporting constructing a more complete world model. The curiosity signal is passed to the actor, shaping general-purpose behaviors unrelated to specific extrinsic goals. While we emphasize curiosity due to its direct link to discovery, the framework supports the integration of other intrinsic motivations, such as homeostatic regulation, affective dynamics, and social drives, to guide behavior across varied contexts.

4. **Extrinsic Motivation:** In addition to internal motivation, the framework supports extrinsic motivation through task-specific rewards. These are provided during goal-directed training phases and are used to evaluate and refine the agent's performance in downstream tasks. These rewards influence the actor and the critic, enabling reinforcement learning in classical RL setups when needed.

5. **Actor:** The actor is responsible for selecting actions based on latent state representations from the world model. It learns a policy that maximizes expected returns from intrinsic and/or extrinsic rewards. In addition to receiving state representations, the actor sends its chosen actions back to the world model, allowing it to anticipate future environmental states before real-world execution. The actor can also incorporate neocortical-inspired circuits, where modular and hierarchical structures support flexible and transferable policies across tasks.

   Critic: The critic estimates how valuable a current state is for the agent's intrinsic or extrinsic goals, guiding the actor's policy updates. Like the world model and actor, the critic can be designed using neocortical circuit principles.

Our framework is highly flexible, allowing for the integration and simultaneous use of multiple intrinsic and extrinsic reward signals to train the agent. The actor and critic components are trained via reinforcement learning. At the same time, the world model consists of multiple neural networks, some of which are optimized jointly with the actor and critic through RL, and others via self-supervised learning using real environmental observations or internal signals as ground truth. In our implementation, the agent is initially trained using only intrinsic motivation, specifically a curiosity-based reward. This phase encourages exploration and leads the agent to develop task-agnostic motor and cognitive skills while actively building a predictive model of the world. We also investigate how embodiment and multimodality richness influence autonomous development during this stage.

Subsequently, the skills acquired during intrinsically motivated exploration are transferred to downstream tasks that rely solely on extrinsic rewards. Our framework enables this transfer straightforwardly and intuitively, bridging self-supervised exploration with goal-directed behavior. Additionally, all trainable components of the system can be implemented using neural architectures inspired by neocortical circuits, thereby enabling a biologically grounded approximation of predictive coding principles found in the brain.

To implement our modules in a functional structure, we use two neural networks in the world model, comprising a **StateNet** and the **StatePredictor**, as shown in Figure 5.2. At each time step t, the StateNet processes the environment's current state $\mathbf{s_t}$ and produces a latent space representation $\mathbf{h_t}$, which is then provided to the actor. Based on this representation, the actor chooses an action $\mathbf{a_t}$, while the critic assesses the state value

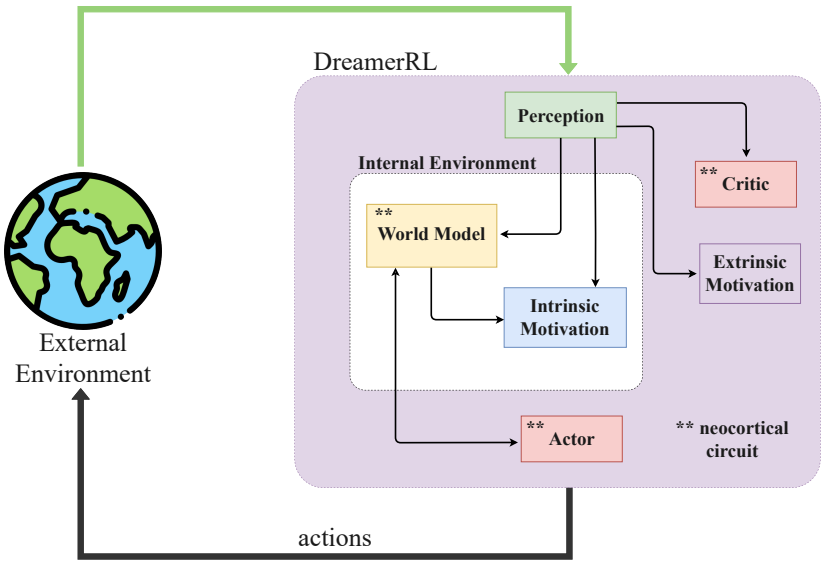


Figure 5.1: The framework consists of six main modules interconnected: (i) Perception, responsible for processing raw sensory data; (ii) Internal Environment, which includes the world model to predict future states of the environment and encode internal representations; and (iii) Intrinsic Motivation, which computes intrinsic rewards; (iv) Actor, which selects actions based on the current latent state representation; (v) Critic, which estimates the value of the current state; and (vi) Extrinsic Motivation, which provides task-specific rewards when available.

$V(\mathbf{s_t})$. Once the action $\mathbf{a_t}$ is selected along with the current state $\mathbf{s_t}$, they both are passed to the StatePredictor, which predicts the next environmental observation $\hat{\mathbf{x}}_{\mathbf{t+1}}$ before the agent interacts with the environment. The selected action is then executed in the environment, yielding a new observation $\mathbf{x_{t+1}}$. This new observation, together with the prediction $\hat{\mathbf{x}}_{\mathbf{t+1}}$ are used in the intrinsic motivation module to determine the intrinsic reward r_{int} and then to train the policy $\pi(\mathbf{a}|\mathbf{s})$.

The StatePredictor generates future predictions, anticipating the next environment's observation based on the current state and the action to be executed by the agent. This prediction generates an internal reward incentivizing the actor to explore new states. The actor is encouraged to promote exploration and learning of new behaviors as the StatePredictor is satisfied with its world predictions. This results in a continuous cycle of discoveries, driving the agent to explore new states in search of knowledge of how the world works. At the same time, StateNet assimilates a representation of all states observed by the agent during exploration, which may be valuable in the future. We hypothesize that this process establishes an iterative cycle of learning and adaptation, where exploration driven by the agent's internal objectives promotes the emergence of complex autonomous behaviors in the environment and more robust and generalizable representations.

Our approach leverages multimodality to enhance sensorimotor integration. The agent's input is the state $\mathbf{s_t}$, comprising a stack of past and present observations, which may originate from a single modality or multiple modalities. All modules receive $\mathbf{s_t}$, processing it through their respective encoders to produce multimodal one-dimensional feature vectors. The StatePredictor is equipped with multiple prediction modules, each corresponding to a specific sensory modality and tasked with predicting the subsequent environmental observation for that modality. The outputs from these modules are concatenated to generate the predicted next observation $\hat{\mathbf{x}}_{\mathbf{t+1}}$. In multimodal settings, the cost function incorporates each modality's predictions to compute the agent's overall reward. We employ a weighted composition strategy to combine the rewards across different modalities, promoting a curious policy that prioritizes the relevance and accuracy of predictions across sensory inputs. This weighted approach to reward composition is designed to encourage the agent to integrate information from multiple modalities effectively. The intrinsic reward function is then given by

$$r_{\text{int}} = \sum_{i=1}^{N} w_i \mathcal{L}_i\Big(\hat{\mathbf{x}}_{\mathbf{i,t+1}}, \mathbf{x}_{\mathbf{i,t+1}}\Big), \tag{5.1}$$

where N is the number of predictor modules present in the StatePredictor, w_i is the weight associated with the i-th predictor module, reflecting its importance in the overall reward such that $\sum_{i=1}^{N} w_i = 1.0$, $\mathcal{L}_i$ is any measure of discrepancy between the prediction and the actual observation for the i-th sensory modality, and $\hat{\mathbf{x}}_{\mathbf{i,t+1}}$ and $\mathbf{x}_{\mathbf{i,t+1}}$ represent the predicted next observation and the actual observation from the environment for the i-th sensory modality, respectively. Throughout our experiments, we will assess whether this reward function promotes richer learning from the environment.

Therefore, the intrinsic reward r_{int} is used for learning the actor's policy. The loss function $\mathcal{L}$ for intrinsic reward is derived from the same loss metric used by the StatePre-

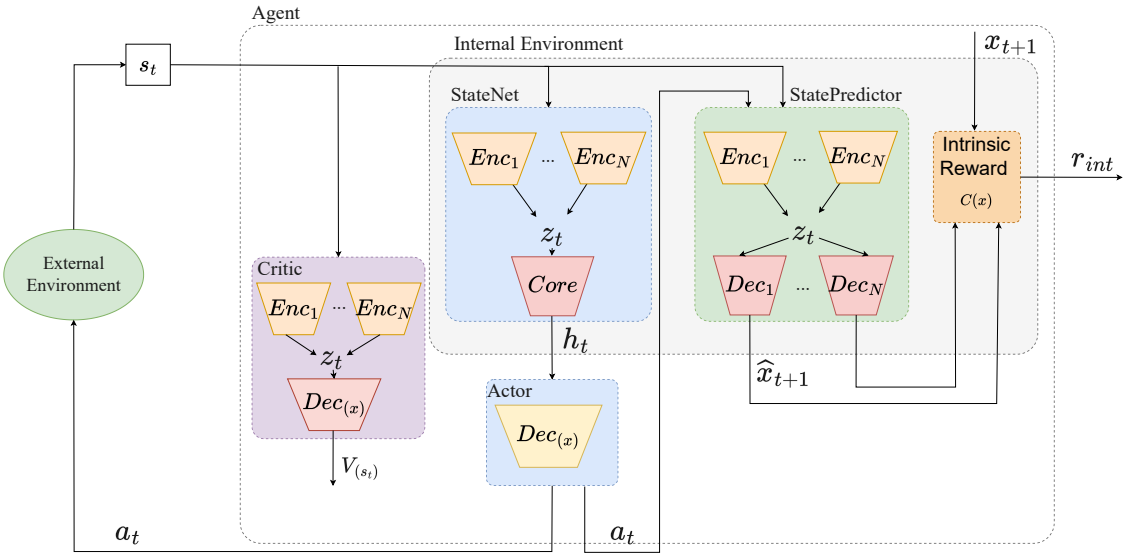


Figure 5.2: Our framework implementation. The world model contains two neural networks: the StateNet and the StatePredictor. At each time step t, the StateNet receives the current state $\mathbf{s_t}$ from the environment and generates a latent space vector $\mathbf{h_t}$, which is passed to the actor. The actor then selects an action $\mathbf{a_t}$, while the critic evaluates the state value $V(\mathbf{s_t})$ reached by the curiosity-driven policy. When the actor selects the action $\mathbf{a_t}$, it is sent along with the current state $\mathbf{s_t}$ to the StatePredictor, which generates predictions $\hat{\mathbf{x}}_{\mathbf{t+1}}$ about the next environmental observation before the agent takes action. Subsequently, the action is executed in the environment, resulting in a new observation $\mathbf{x_{t+1}}$. This new observation $\mathbf{x_{t+1}}$, along with the prediction $\hat{\mathbf{x}}_{\mathbf{t+1}}$, is used to calculate the intrinsic reward that adjusts the agent's curiosity-driven policy. In this way, the predictions made by the StatePredictor guide the agent's curiosity. When the StatePredictor makes accurate predictions, the agent's intrinsic reward decreases, signaling that the environment is well understood. This process encourages the agent to explore new actions that challenge the predictions of the StatePredictor, promoting continuous learning and enhancing the agent's understanding of the environment.

dictor. The intrinsic reward decreases as the StatePredictor improves its world prediction. Simultaneously, the actor, critic, and StateNet are updated to maximize the intrinsic reward gain. As the agent learns about the environment, the actor generates novel actions to surprise the StatePredictor with unpredictable situations, increasing its intrinsic curiosity. This process generates a cycle of discovering complex behaviors for the agent. While the StatePredictor operates to make the best possible prediction of the next environmental observation, the actor adjusts to perform actions that maximize the given intrinsic reward; consequently, our agent employs two types of losses. The StatePredictor modules are parameterized by $\theta_{\mathbf{SP}}$ and minimized in a self-supervised manner using regression loss, as

$$\mathcal{L}_{SP} = \min_{\theta_{SP}} \sum_{i=1}^{N} \mathcal{L}_i \Big(\hat{\mathbf{x}}_{\mathbf{i,t+1}}, \mathbf{x}_{\mathbf{i,t+1}} \Big), \tag{5.2}$$

where $\mathcal{L}_{SP}$ measures the discrepancy between the predicted and actual observation of the

environment. The objective of this loss is to enable the StatePredictor to learn the best possible prediction before the agent takes the chosen action. This prediction encompasses aspects of causality, dynamics, and visual aspects of the environment.

Meanwhile, the StateNet, actor, and critic are trained via reinforcement learning to maximize the sum of the expected intrinsic rewards as

$$\max_{\theta_{SN},\theta_C,\theta_A} E_{\pi(\mathbf{s_t};\theta_{SN},\theta_C,\theta_A)} \left[\sum_t r_{\text{int}} \right], \tag{5.3}$$

where $\theta_{\mathbf{SN}}$, $\theta_{\mathbf{C}}$, and $\theta_{\mathbf{A}}$ are the parameters of the StateNet, critic, and actor, respectively. And r_{int} is the intrinsic reward.

To illustrate the DreamerRL functioning, we present the developed Algorithm 3. This algorithm outlines the agent's learning process using PPO. The process begins by initializing the parameters of the involved networks: StateNet, StatePredictor, Actor, and Critic. At the beginning of each episode, the agent observes the environment's initial state and, at each time step, selects an action based on the policy learned by the actor network. This action is executed in the environment, resulting in a new observation. Simultaneously, the intrinsic reward is calculated based on the prediction error of the next observation by the StatePredictor, reflecting the agent's curiosity in unexpected situations. At the end of each episode, state transitions and rewards are stored in a buffer, which is used to update the policy. During this stage, the StatePredictor's supervised loss and the policy's loss are minimized using PPO, adjusting the actor's policy and updating the critic network to improve the state evaluation.

Despite the agent's being driven by curiosity, the proposed approach converges toward learning stable policies, as predicted by reinforcement learning theory. This convergence is feasible for several reasons. First, although the agent operates in a stochastic environment and follows a novelty-driven policy, the environment contains stationary and stable elements that can be reliably learned, such as spatial regularities, fixed objects, physical laws, and motor constraints that remain invariant over time. These components enable the agent to rapidly construct compact and consistent representations of the environment within the early training episodes, stabilizing key components of the value function and the learned policy, and the curiosity signal transitions from acting as a purely exploratory force to functioning as a selective mechanism, targeting only partially unknown regions of the environment. As a result, exploratory behavior becomes increasingly focused and efficient. Additionally, a neural network trained in a self-supervised manner predicts future environmental states. This neural network is not directly influenced by policy gradients, being only indirectly shaped by the actions selected by the actor. This decoupling ensures the stability of latent state learning, even in an adaptive exploratory policy.

5.2 The Intrinsic-motivated Agent

In this section, we conducted experiments in a controlled manipulation environment to create and validate the intrinsic-motivated agent and experimental setup. The environment consists of a table with three differently colored cubes (i.e., red, green, and blue)

Algorithm 3 The DreamerRL's learning procedure

Initialize StateNet parameters θ_{SN}, StatePredictor parameters θ_{SP}, actor parameters θ_A, and critic parameters θ_C;
Initialize environment and observation buffer $\mathcal{B}$;
for each episode **do**
  Reset environment, observe initial state $\mathbf{s_0}$;
  for each time step t **do**
    Sample action $\mathbf{a_t}$ from actor $\pi(\mathbf{a_t}|\mathbf{s_t};\theta_\mathbf{A})$;
    Given action $\mathbf{a_t}$, predict the next observation $\hat{\mathbf{x}}_{\mathbf{t+1}}$ from StatePredictor;
    Execute action $\mathbf{a_t}$, observe the next state $\mathbf{s_{t+1}} = (\mathbf{x_{t-1}}, \mathbf{x_t}, \mathbf{x_{t+1}})$ from the environment;
    Compute intrinsic reward r_{int} using the intrinsic loss function $\mathcal{L}_{SP}$;
    Store $(\mathbf{s_t}, \mathbf{a_t}, r_{int}, \mathbf{s_{t+1}})$ in buffer $\mathcal{B}$;
  end for
  Perform PPO update:
  for each batch from buffer $\mathcal{B}$ **do**
    Compute advantage estimates $\mathbf{A_t}$ using the Critic;
    Update Actor by maximizing $\mathcal{L}_{actor} = \mathbb{E}_t \left[\frac{\pi(a_t|s_t;\theta_{AC})}{\pi_{old}(a_t|s_t;\theta_{AC})} \mathbf{A_t} \right]$;
    Update Critic by minimizing $\mathcal{L}_{critic} = \mathbb{E}_t \left[(r_{\text{int}} + \gamma V(\mathbf{s_{t+1}};\theta_\mathbf{C}) - V(\mathbf{s_t};\theta_\mathbf{C}))^2 \right]$;
    Update StatePredictor by minimizing $\mathcal{L}_{SP} = \sum_{i=1}^{N} \mathcal{L}_i(\hat{\mathbf{x}}_{\mathbf{i,t+1}}, \mathbf{x}_{\mathbf{i,t+1}})$;
    Update $\theta_\mathbf{SN}$, $\theta_\mathbf{SP}$, $\theta_\mathbf{A}$ and $\theta_\mathbf{C}$ using gradient descent;
  end for
end for

and the NAO robot seated in front of it. In this stage, the robot was allowed to freely explore the objects, relying solely on intrinsic curiosity-based rewards to guide its actions. Throughout the training process, the robot was driven by uncertainty and the desire to minimize the gap between its internal predictions and the real-world observations made during its interactions with the environment.

To track the curiosity level developed by the robot about the objects, we utilized the average number of touches made by the fingertips across all episodes prior to the PPO update, called mean interaction intensity. This parameter was essential for computing the frequency of the robot's interactions with the cubes, serving as an indirect metric of its curiosity. As the robot engages more with the objects, this metric is expected to increase, indicating a rise in the number of touches and manipulations, suggesting that the robot is increasingly interested in exploring and understanding the properties of the cubes in front of it. We expect that the robot will initially exhibit few interactions with the objects, focusing on learning the static and structural aspects of the environment. Once these aspects have been assimilated, we hypothesize that the robot will begin to interact more frequently with the cubes on the table, as their dynamic nature has the potential to stimulate curiosity.

Since our first experiment, DreamerRL has already contained all the theoretical elements, but they were implemented simply. We will enrich these elements and make them more complex during the work. Therefore, we implement multimodality through vision and proprioception signals, intrinsically motivated reinforcement learning with a curious

reward function from only a single modality, and neocortical circuitry through monolithic hierarchical artificial neural networks.

Observation and State Space. Our experiment utilizes a multi-modal observational space composed of images and proprioception signals. Therefore, each observation $\mathbf{x_t}$ is given by

$$\mathbf{x_t} = \left[\mathbf{p}, \mathbf{i^t}, \mathbf{i^f}\right], \tag{5.4}$$

and comprises the proprioception vector $\mathbf{p}$, two RGB images of 512×256 pixels, one seen from the tabletop at a 90-degree angle $\mathbf{i^t}$ and the other from the front at a 45-degree angle $\mathbf{i^f}$. All images are normalized before leaving the environment. We employed z-normalization on the images by randomly sampling the states 500 times. The mean and standard deviation were computed to normalize the images. For the joints, the joint space was mapped to a range of -1 to 1.

From this perspective, the RL state $\mathbf{s_t}$ is composed of 3 consecutive observations of the scene as

$$\mathbf{s_t} = [\mathbf{x_t}, \mathbf{x_{t-1}}, \mathbf{x_{t-2}}], \tag{5.5}$$

stacking observations to compose a state is mainly helpful in avoiding perceptual aliasing, where multiple states may give rise to the same perception. We use skip-frame 1 to ensure a better representation of the state.

Reward Function. We employed an intrinsic reward function, utilizing the normalized mean squared error (MSE), which quantifies the average squared difference between predicted and actual values while being minimally sensitive to outliers as

$$r_{\text{int}} = 1 - \frac{1}{1 + \text{MSE}(\hat{\mathbf{x}}_{\mathbf{t+1}}, \mathbf{x_t})}, \tag{5.6}$$

where r_{int} is normalized between 0 and 1; when the reward is 0, the agent can accurately predict the environment's next observation. When it is 1, the agent cannot predict anything correctly about the environment. This reward encourages the agent to strive to learn all static and dynamic characteristics of the environment, thereby stimulating the actor to generate actions that elicit increasingly novel and creative situations. In this work, we normalize the reward between 0 and 1 to facilitate the comparison of different intrinsic rewards and ensure that all rewards are on the same scale.

Action Space. At each time step t, the action $\mathbf{a_t}$ corresponds to 26 angular joint values of the NAO robot. These joints pertain to the hands and arms, as the head is fixed. At this stage, we decided to introduce the minimum possible embodiment and sensorimotor integration in the agent, aiming to evaluate intrinsic learning in a more isolated manner relative to other variables. Furthermore, including head movements at this point would significantly increase the experiment's complexity, complicating the analysis of the results. This approach allows us to focus our investigation on the effects of curious intrinsic reward on exploration.

Agent Architecture. In the StatePredictor, we employ a UNET encoder-decoder [120] to generate predictions of the next frame, as shown in Figure 5.3. Specifically, we utilize

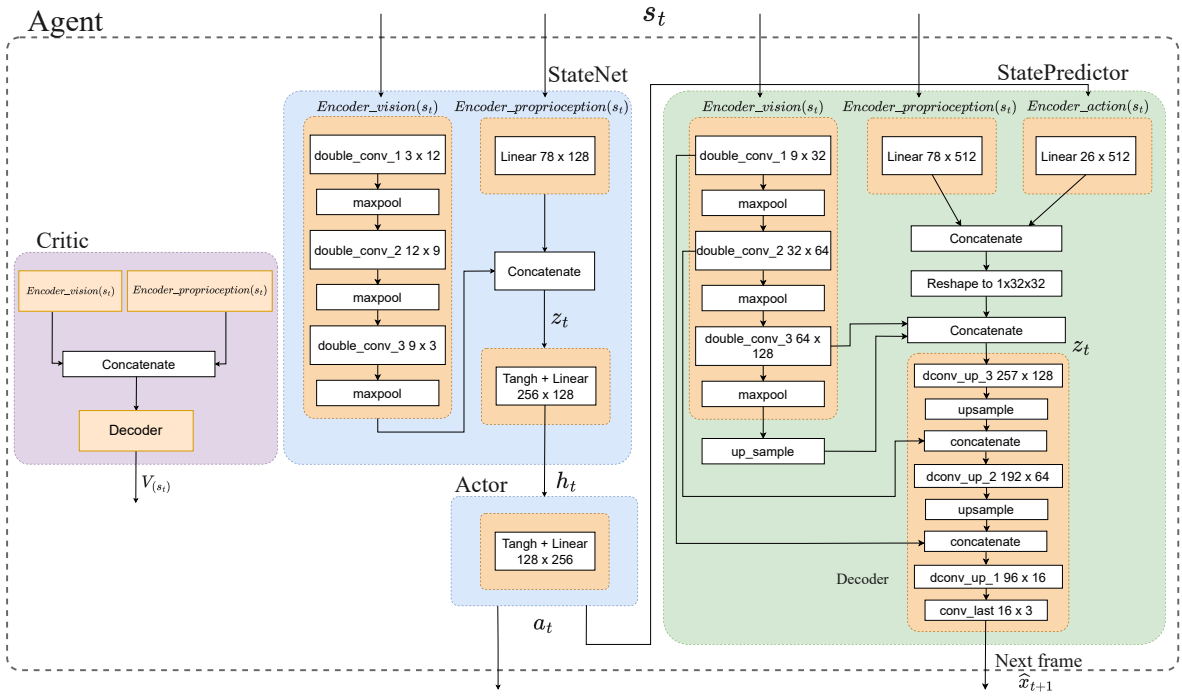


Figure 5.3: The agent details. The agent uses a UNET encoder-decoder in the StatePredictor to next frame prediction $\hat{\mathbf{x}}_{\mathbf{t+1}}$, employing convolutional layers for vision and linear layers for proprioception and actions. The encoded information $\mathbf{z_t}$ is passed through the decoder, producing an image of $128 \times 128 \times 3$ pixels. The StateNet utilizes the UNET encoder and linear layers to process vision and proprioception, with distinct Actor and Critic networks.

an Encoder_vision($\mathbf{s_t}$) with three layers of double convolutional operations with filter sizes of 9×32, 32×64, and 64×128 to encode the stack of frames from the state $\mathbf{s_t}$. A linear layer with 512 units in Encoder_proprioception($\mathbf{s_t}$) encodes proprioception, while another linear layer with 512 units in Encoder_action($\mathbf{a_t}$) encodes the action $\mathbf{a_t}$ chosen by the agent. Subsequently, we concatenate all this information $\mathbf{z_t}$ and pass it through the Decoder($\mathbf{z_t}$), which contains three double convolutional layers and two residual upsample layers, ultimately generating the image of size $128 \times 128 \times 3$ pixels $\hat{\mathbf{x}}_{\mathbf{t+1}}$.

In the StateNet, we employ a UNET encoder to process visual input, a linear layer with 128 units to process proprioception, and a linear layer with 256 units to process concatenated information from vision and proprioception. For the Actor, we utilize only one linear layer with 128 units. Finally, for the Critic, we adopt a structure identical to the StateNet. Hyperbolic tangent activation functions are employed between all linear layers. The StateNet does not share parameters with the StatePredictor. The StateNet is tuned based on iterations of the Actor using the reward derived from the intrinsic curiosity reward, thus indirectly connecting them through the reward signal. Our agent consists of 4.94 million trainable parameters with a training time of approximately 72 hours.

Training. We trained the agent for 3×10^6 steps, collecting 12 rollouts with a trajectory length of 32. Considering our moderately small batch size, we adopted a learning rate of 1×10^{-4} for all trained networks. Additionally, we refined the policy every 20

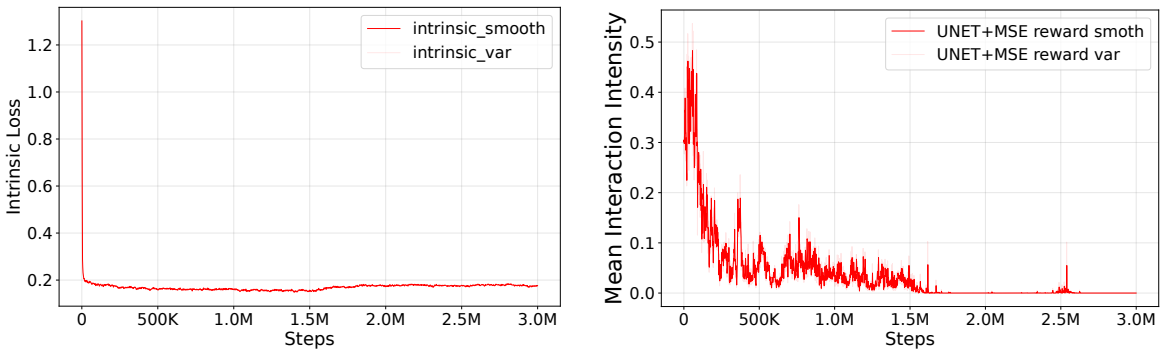
Table 5.1: Hyperparameters employed for training our agent.

Roll-outs	12
Trajectory Length	32
Learning Rate Internal Environment	1×10^{-4}
Learning Rate Actor	1×10^{-4}
Learning Rate Critic	1×10^{-4}
Epochs	20
Discount Factor	0.99
PPO Clip	0.2
Policy Std	0.5

epochs but did not utilize the Generalized Advantage Estimation (GAE) [128] at this stage. Reward normalization was also employed to reduce variance. Table 5.1 summarizes the parameters used for training. We kept the policy standard deviation fixed during training at a value of 0.5.

During training, we assessed the agent quantitatively and qualitatively. We evaluated the behavior of the StatePredictor loss $\mathcal{L}_{SP}$ to ensure that the agent is learning world predictions as expected, as depicted in Figure 5.4 (a). To assess the degree of interaction with objects, we also computed the mean interaction intensity given to any cube by the robot's fingertips. The result is the average sum of collisions with the fingers on all three cubes, such that each collision between a phalange and a cube counts as one collision point. If the robot perfectly grasps a cube with both hands, there is a potential to score 16 points, and if this occurs in all collected rollouts, we would have an average score of 16 per episode. To qualitatively assess the behaviors explored by the agent and the environmental aspects it learned, we saved a sequence of frames from randomly selected rollouts at each training iteration. We sampled these rollouts at six distinct steps.

The loss $\mathcal{L}_{SP}$ indicates convergence, and the mean interaction intensity, illustrated in



(a) Intrinsic loss

(b) Mean interaction intensity

Figure 5.4: Our agent's loss $\mathcal{L}_{SP}$ and the mean interaction intensity. In (a), the $\mathcal{L}_{SP}$ converged, presenting only a small residual error at the end due to aspects of the environment that could not be predicted. In (b), we use the mean interaction intensity to evaluate the agent's number of iterations with the objects during training. The mean interaction intensity demonstrates that at the beginning of training, the agent iterated much more with the objects, and after 1 million steps, these iterations became increasingly sparse.

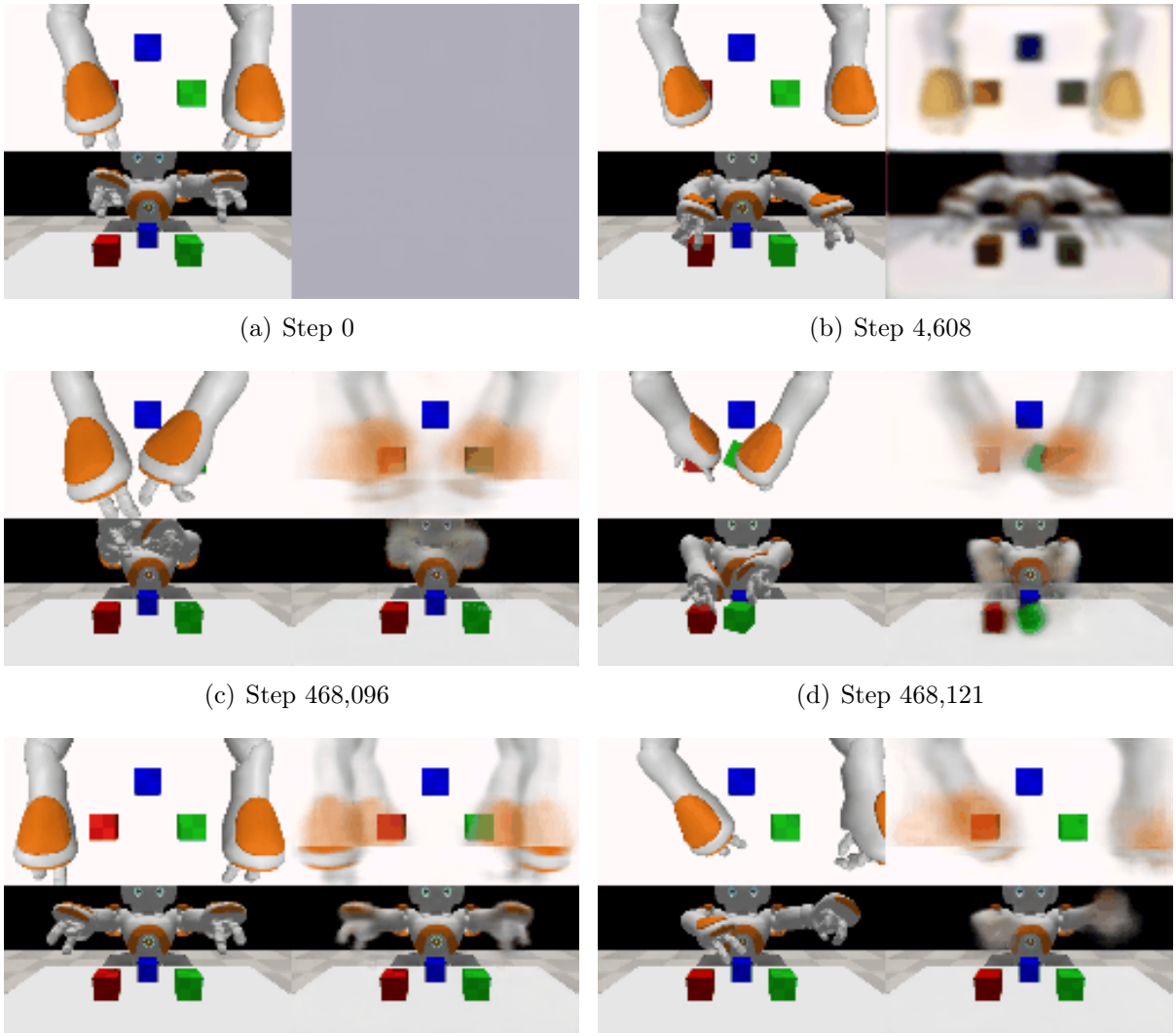
Figure 5.4 (b), shows that the agent interacted with objects through some touches but did not engage in curiosity or much interaction with the objects as we had hypothesized. From 1.5 million steps onward, the agent's interactions with the objects became more sparse. However, the loss $\mathcal{L}_{SP}$ shows that the agent learned various aspects of the environment, especially in the early stages of training, where the loss decreases abruptly. This behavior suggests that static aspects of the environment were rapidly assimilated, with a slight remaining error, possibly corresponding to dynamic aspects, such as the relationship between the robot, its joints, and the surrounding objects. At the beginning of training, the agent perceives the environment as a static gray image, as shown in Figure 5.5 (a). However, after only 4,000 steps, it learned to identify the shape and location of static objects, with limited knowledge of the colors and dynamic objects. In the first five hundred thousand steps, the focus shifts towards improving the perception of object colors and understanding the dynamics of its arms and hands, which appear to be incompletely learned, as shown in Figures 5.5 (b) and (c). By around 500,000 steps, the robot is concentrated on understanding inverse kinematics and the dynamics of its limbs to interact with dynamic elements of the scene, thereby enhancing prediction and internal understanding of its own body, as evidenced in Figure 5.5 (d).

Later, towards the end of training, we observed the agent's focus on understanding its arms and how this influences scene prediction, still struggling to predict arm and hand movements accurately. This results in a behavior where the robot moves its hands from right to left until the end of training. The mean interation intensity plot, depicted in Figure 5.4 (b), confirms this interpretation, revealing that the number of interactions of the finger phalanges with objects on the table gradually decreases after approximately 1.5 million steps, as the robot focuses more on understanding how the movement of its arms affects prediction, given that arm predictions in the upper frame have a more significant impact on loss. The mean values of the number of touches also indicate that the robot interacted more with objects at the beginning of training, resulting in learning the world's dynamic aspects. However, compared to the maximum possible values, the low interaction values suggest that the robot has developed only simple interaction behaviors with objects, such as touching with the fingertips, as illustrated in some frames in Figure 5.6.

The results indicate that the agent is learning a curiosity-driven policy, gradually acquiring knowledge about various aspects of the environment, though with a progression slightly different from what was expected. Initially, the agent learns static aspects and quickly explores the objects' dynamics, shape, and form. By the end of training, the agent consistently focuses on investigating its hands and arms through the top camera. This behavior suggests that once the agent fully learns to predict its body's movements, there is more interest in interacting with external objects. This outcome is particularly intriguing as it mirrors the infants' early developmental stages, where self-exploration, especially with the hands, takes precedence over external engagement. Since the agent has yet to complete the self-exploration phase, as evidenced by its inability to precisely predict its arms and fingers' movements, external objects have not become its primary focus.

To further enhance the agent's curiosity-driven learning and improve its ability to complete the self-exploration phase, we hypothesize that modifying the $\mathcal{L}_{SP}$ loss function

from Mean Squared Error (MSE) to the Structural Similarity Index (SSIM) [151] could lead to higher-quality visual predictions, particularly for the dynamic regions of hands and

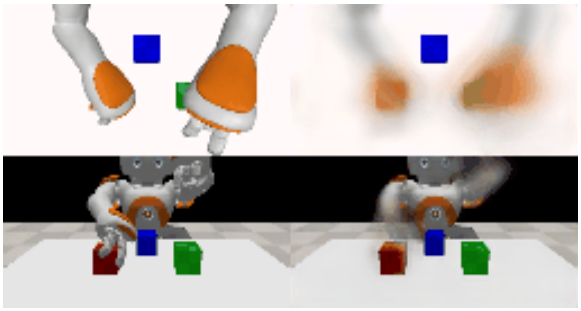


(a) Step 0 (b) Step 4,608

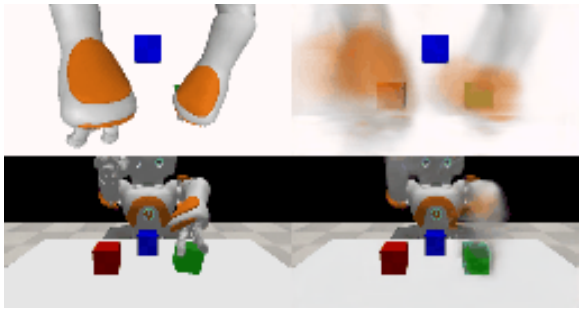
(c) Step 468,096 (d) Step 468,121

(e) Step 2,000,000 (f) Step 2,000,028

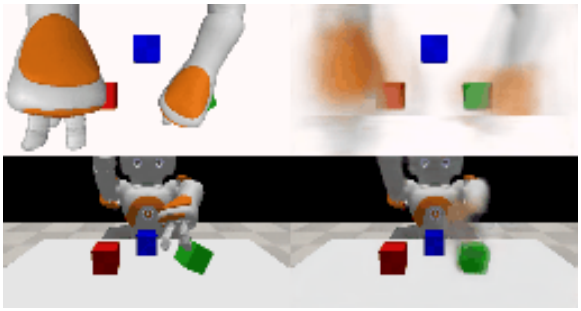
Figure 5.5: Pairs of episode samples at different training steps demonstrate what the agent has learned. In each case, the frame on the left represents the scene provided by the external environment at time $t + 1$, while the frame on the right shows the agent's prediction for the scene at time t. The two frames should be identical when the learning process is completed. In (a), we observe the agent at step 0 of learning, where it is evident that the agent's prediction of the environment is still unclear, leading to a gray frame prediction. In (b), at step 4,068, the agent has learned the position and shape of static elements in the scene but struggles to predict dynamic aspects, such as arm movements correctly. In (c) and (d), the agent successfully differentiates the color of all elements in the scene and begins interacting with the objects around it. In (d), the agent's learning of the effects of its fingers on the objects becomes noticeable, as shown through its interactions with the green cube. In (e) and (f), the agent still faces challenges in predicting the movements of its arms, which remains one of its most significant obstacles. For more details, watch the video.



(a) Step 181,632



(b) Step 1,088,256



(c) Step 1,088,257

Figure 5.6: Pairs of episode samples at different training steps demonstrate the agent interacting with objects on the table. The agent can make subtle touches on the objects.

fingers. While MSE measures the average of the squared differences between predicted and actual pixel values, it fails to capture more complex image structure aspects crucial for accurate predictions in dynamic environments. In contrast, SSIM is more appropriate for our agent as it evaluates image quality by considering factors beyond mere pixel values, such as luminance, contrast, and structural details, which are essential for understanding the overall composition of a scene. Moreover, SSIM has been widely used in contexts involving high-resolution image prediction and visual odometry, making it a more robust choice for tasks where preserving structural information is critical. By adopting SSIM, we expect the agent to achieve more accurate predictions of its body, thus accelerating its self-exploration learning and enabling it to progress to more advanced exploration in the external environment.

5.2.1 Visual Predictions with Structural Similarity Index

The Structural Similarity Index (SSIM) [151] is a widely used metric for evaluating image quality by measuring the perceptual similarity between two images. As an alternative to the Mean Squared Error (MSE), which often poorly correlates with human perception, SSIM compares three similarity maps: luminance, contrast, and structure. These maps are generated from local regions of the input images and assessed using functions like the Pearson correlation coefficient. The final SSIM metric is derived by combining the luminance $l(\mathbf{x}, \mathbf{y})$, contrast $c(\mathbf{x}, \mathbf{y})$, and structure $s(\mathbf{x}, \mathbf{y})$ comparison functions, as follows

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \tag{5.7}$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \tag{5.8}$$

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \tag{5.9}$$

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma, \tag{5.10}$$

where C_1, C_2, e $C_3 = \frac{C_2}{2}$ are constants to ensure stability when the denominator becomes 0; μ_x and μ_y represents the mean of the given images $\mathbf{x}$ and $\mathbf{y}$, respectively; σ denotes the standard deviation of the given images; $\alpha > 0$, $\beta > 0$, $\gamma > 0$ denote the relative importance of each of the metrics. We assume $\alpha = \beta = \gamma = 1$ in our implementation. The metric values were adjusted within the range $[0, 1]$, where zero indicates that the two images are entirely different, and one indicates that the two images are identical.

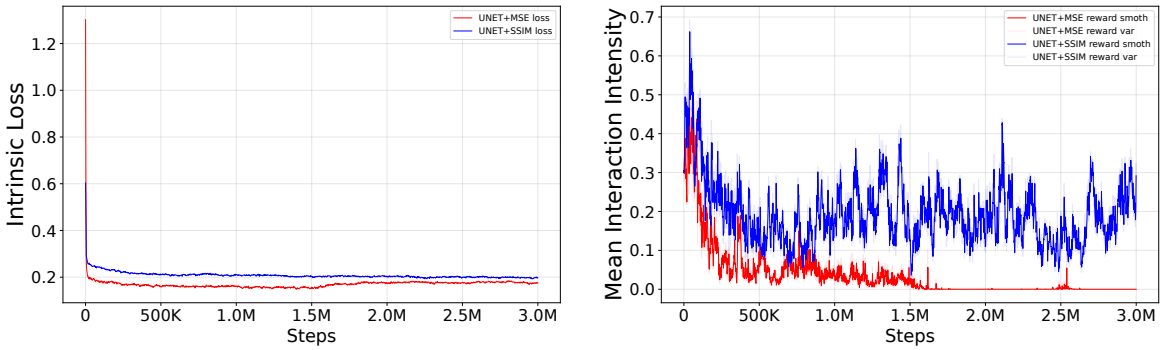
Since our intrinsic reward function is designed to encourage the agent to explore new states when the current state is already known, r_{int} is defined as

$$r_{\text{int}} = 1 - \text{SSIM}(\mathbf{x}, \mathbf{y}), \tag{5.11}$$

when $r_{\text{int}} = 0$, the agent can predict the next frame completely; this low reward encourages the agent to explore new states. Conversely, when $r_{\text{int}} = 1$, the agent has yet to learn the current state and should persist until it is adequately learned.

For training, we maintained unchanged training parameters and agent configurations to assess whether SSIM positively affects the agent's predictions, adhering to the same settings described in Section 5.2. The results shown in Figure 5.7 indicate that the StatePredictor's loss $\mathcal{L}_{SP}$ decreases at a more consistent rate, reaching a residual error similar to that of the agent using MSE. However, an evolution in the agent's self-exploration is noted, as evidenced in Figure 5.7 (b), where interactions with objects became more consistent over time, indicating that the agent has learned to predict relevant aspects of its body dynamics more efficiently, now being able to explore and discover new interactions with the dynamic objects in the scene. This progress is analogous to the exploratory behavior of infants, who, as they become familiar with the movements and limits of their bodies, begin to interact with their environment more actively and continuously. Posteriorly, infants gradually develop motor and perceptual skills, allowing them to explore and manipulate objects with increasing curiosity and intentionality.

The quantitative results are also qualitatively confirmed. In Figure 5.8 (a), both agents are very similar. However, as the training progresses, in Figure 5.8 (b), at step 6,528, the SSIM-trained agent demonstrates some understanding of its arm's movement, something the MSE-trained agent still cannot predict with the same accuracy at this time. In Figure 5.8 (c) and (d), the SSIM-trained agent can predict high-quality visual images, demonstrating accuracy in both the objects' dynamics and arm movement. In the tabletop image, the agent precisely predicts the shape and position of its arms, accurately



(a) MSE and SSIM losses

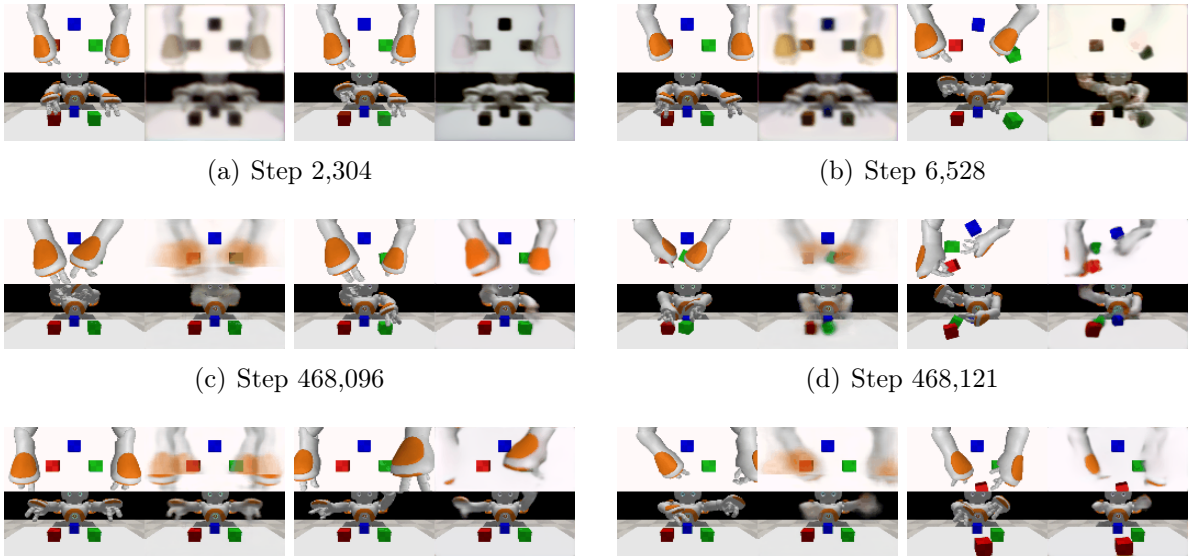
(b) Mean interaction intensity

Figure 5.7: Training results of our agent, using the SSIM metric as the $\mathcal{L}_{SP}$ loss and as part of the reward function. In (a), we observe the losses using SSIM and MSE, which converge with only a small residual error at the end, reflecting aspects of the environment that could not be predicted. In (b), we see the curve of the agent's interactions with the objects on the table; it is evident that the agent using SSIM learned more easily about various aspects of the environment and its body, resulting in more frequent interactions with the objects.

estimating their speed. Even in the final training steps, in Figure 5.8 (e) and (f), the agent maintains high-quality arm predictions, enhancing its ability to interact with the objects. As a result, the agent reduces the need to frequently touch its arms to the tabletop camera to understand the dynamics of its upper limbs, a behavior commonly observed in the agent trained with MSE. Additionally, the agent concludes its training by actively exploring and throwing the cubes into various positions, indicating an expanded phase of curiosity compared to the MSE-trained agent. In this phase, it experiments with new ways of manipulating objects, exploring the scene through more reactive actions, such as striking the cubes with its hands more frequently and intentionally.

Despite the advances achieved using SSIM, the agent still encounters difficulties in simultaneously predicting the dynamics and form of the arms in some frames, as shown in Figure 5.8, (e) and (f). This observation led us to hypothesize that replacing the UNET with an architecture more specialized for dynamic image generation could yield significant improvements. The UNET encoder-decoder we implemented was initially designed to segment and reconstruct static images; however, in our context, it may not be the most suitable for predicting complex dynamics, such as the articulated motion of upper limbs.

To address this limitation, we propose using Generalized Divisive Normalization (GDN) [7] between conventional convolutional layers. GDN is widely used in convolutional neural networks to stabilize and enhance the quality of image generation. In similar tasks, such as video enhancement and high-fidelity geometric image synthesis, architectures incorporating GDN have proven effective at capturing spatial details and reducing undesirable artifacts during prediction [66]. Previous research also indicates that GDN helps regularize neural networks, making them less sensitive to minor variations in input data, which is an essential factor for improving training stability, image consistency, and generalization [7]. We believe incorporating GDN between convolutional layers, alongside the SSIM loss function already in use, will help stabilize the StatePredictor's training and enhance



(a) Step 2,304 (b) Step 6,528

(c) Step 468,096 (d) Step 468,121

(e) Step 2,000,000 (f) Step 2,000,028

Figure 5.8: Comparison of the agent's predictions using MSE and SSIM across various training steps. In each alternative, the first pair of images on the left refers to MSE-based agent results, and the pair on the right refers to SSIM-based agent results. In (a), at training step 2,304, both agents have already learned the objects' shape, the grayscale differences of the environment's colors, and the shape of their bodies. However, they still face difficulty in fully predicting the dynamic of their arms. In (b), at step 6,528, the SSIM-based agent has begun learning the dynamics of the scene elements. The predicted movement of the green cube still shows a slight shift compared to its actual position, but the main challenge remains to predict the movements of the arms and hands. In (c) and (d), the SSIM-based agent has almost perfectly learned some arm movements and provides an accurate prediction of the objects' movement on the table, something the MSE-based agent has not yet achieved. In (e) and (f), the SSIM-based agent exhibits minor difficulties predicting the fingers and arm shapes. It already provides a more accurate prediction of the upper limbs compared to the MSE-based agent, which depicts the arms in multiple movement positions within a single frame. Thus, the SSIM-based agent has improved the quality of predictions for the image's dynamic parts.

prediction accuracy, particularly in the regions of the agent's hands and fingers.

5.2.2 Visual Predictions with Generalized Divisive Normalization

The primary idea behind Generalized Divisive Normalization (GDN) [7] is to transform the input data such that its distribution becomes closer to a normal distribution or any other desired distribution. This transformation is achieved by applying a series of operations, including mean subtraction, division, and a nonlinear activation function, to normalize the activations of intermediate convolutional layers. In summary, the GDN that we apply between convolutional layers is

$$y_i = \frac{z_i}{\left(\beta_i + \sum_j \gamma_{ij} |z_j|^{\alpha_{ij}}\right)^{\varepsilon_i}}, \tag{5.12}$$

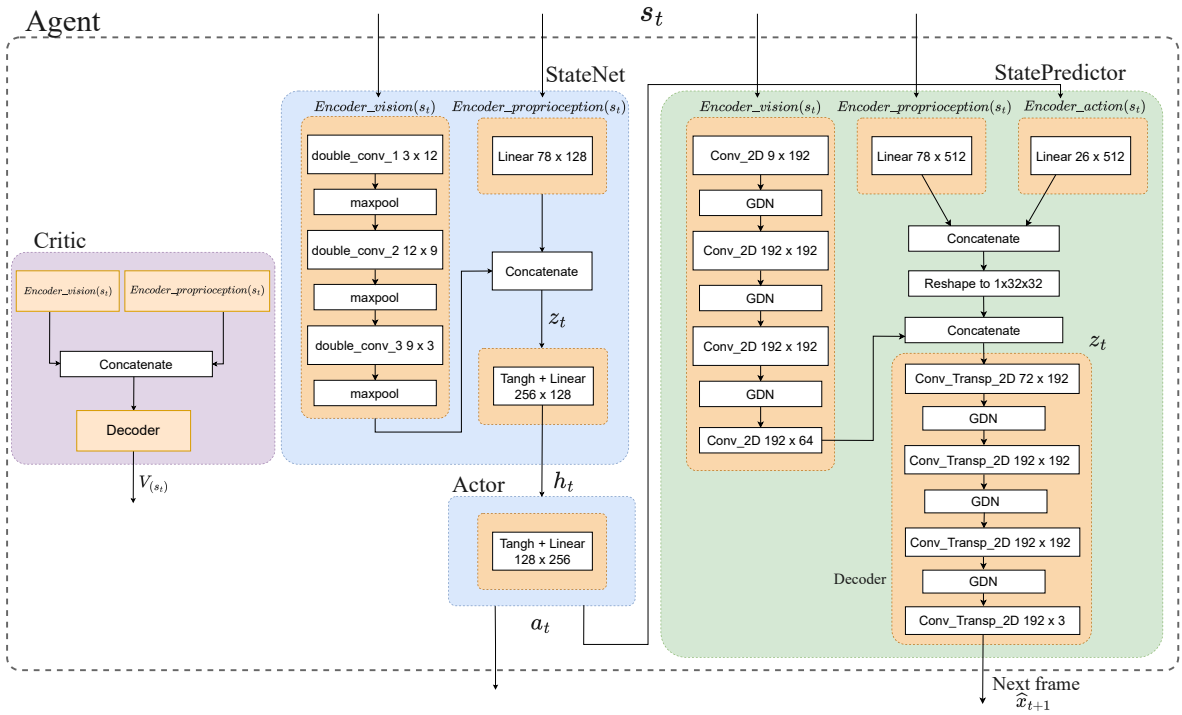


Figure 5.9: The agent implemented a convolutional encoder-decoder with GDNs in the StatePredictor to predict the next observation $\hat{\mathbf{x}}_{\mathbf{t+1}}$. Then, the encoded information is passed through the decoder, producing an image of $128 \times 128 \times 3$ pixels.

where β_i defines a normalization base value to ensure stability, γ_{ij} regulates the contribution of each input to the normalization of each output, α_{ij} adjusts the intensity of each channel before normalization, and ε_i modulates the division's intensity, thereby controlling the compression of each channel's response. We set $\varepsilon = 0.5$, $\alpha_{ij} = 2.0$, and $\beta_i \approx 1.0$, while γ is a diagonal matrix initialized as $\sqrt{\gamma_{init} \cdot I + \text{pedestal}}$, with $\gamma_{init} = 0.1$, $pedestal = 1.45^{-11}$, and I as an identity matrix whose dimension depends on the number of input features. Both parameters, β and γ, undergo reparameterization during the forward pass, which ensures values remain within defined limits, preventing them from becoming negative.

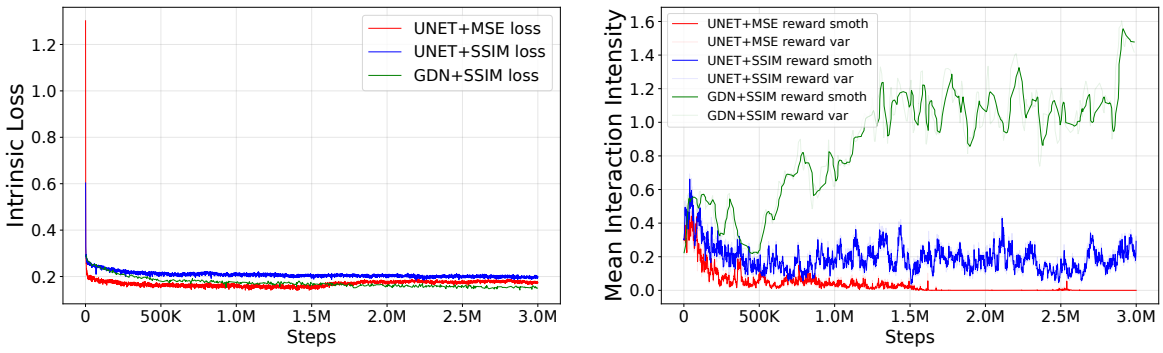
In this experiment, we trained the agent using the SSIM metric as the intrinsic reward and the loss functions for the StatePredictor. We replaced the UNET encoder-decoder in the StatePredictor with a classic convolutional encoder-decoder structure, composed of four convolutional layers with GDN between them, using filters of sizes 9×192, 192×192, 192×192, and 192×64. In the decoder, we included four 2D transposed convolutional layers, also with inverse GDN between them, with filters of sizes 72×192, 192×192, 192×192, and 192×3. The complete architecture is shown in Figure 5.9. We used the same training parameters described in Section 5.2 and maintained approximately the same number of trainable parameters, resulting in an agent with approximately 4.9 million parameters. The training process takes approximately 72 hours to complete.

Our training results demonstrate that the agent modifications significantly improved the prediction in dynamic regions such as the arms and hands. Beyond this, these en-

hancements also increased the agent's level of engagement with objects in the scene. These adjustments enabled more comprehensive self-exploration, allowing the agent's curiosity to expand from exploring its arms to interacting with dynamic external elements in the environment, which are components that offer several opportunities for discoveries, unlike the fixed table and background. As shown in Figure 5.10 (b), the mean interaction intensity curve grew almost linearly over time, reflecting the agent's increasing interest in exploring objects in novel ways and resulting in extended interaction time with them compared to previous agents.

The agent's increased engagement with objects in the environment extends beyond a simple rise in touch frequency or an improvement in self-exploration; it also signals a significant advancement in the agent's autonomous development. The agent progresses from simple, random interactions to more intentional and coordinated behaviors (Figure 5.11). In previous implementations, agents only performed brief, seemingly random touches on objects. However, in the current agent, we observe a progression that begins with these initial touches and gradually evolves into more complex actions, such as holding objects for longer, lifting them, and throwing them upward or off the table in repetitive motions by the end of training (Figure 5.12). This progression of behaviors suggests that it is exploring the environment more intentionally to understand the effects of its actions. Additionally, we observe that the agent's behavior of throwing objects upward is highly repetitive, indicating the challenges it faces in imagining the trajectory of objects under the influence of gravity. As shown in Figure 5.12 (b) and (c), the agent takes considerable time to predict the details of object trajectories when thrown into the air, and even at the end of the training, it has not yet perfected its ability to anticipate the trajectories of all objects.

The current agent's behavior of throwing cubes into the air and the progress of autonomous development reveal a crucial insight: computational curiosity is a fundamen-



(a) Intrinsic loss

(b) Mean interaction intensity

Figure 5.10: Training results of our agent, using the GDN transform in the StatePredictor. In (a), we observe the $\mathcal{L}_{SP}$ loss using SSIM and MSE, which converge with only a small residual error at the end, reflecting aspects of the environment that could not be predicted. In (b), we see the curve of the agent's interactions with the objects on the table; it is evident that the agent using GDN and SSIM learned more easily about various aspects of the environment and its body, resulting in more frequent interactions with the objects.

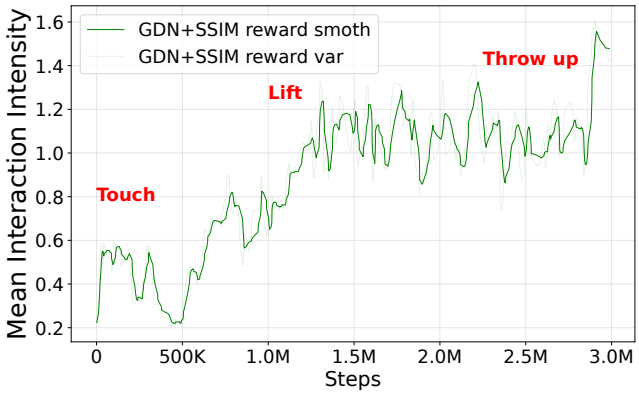


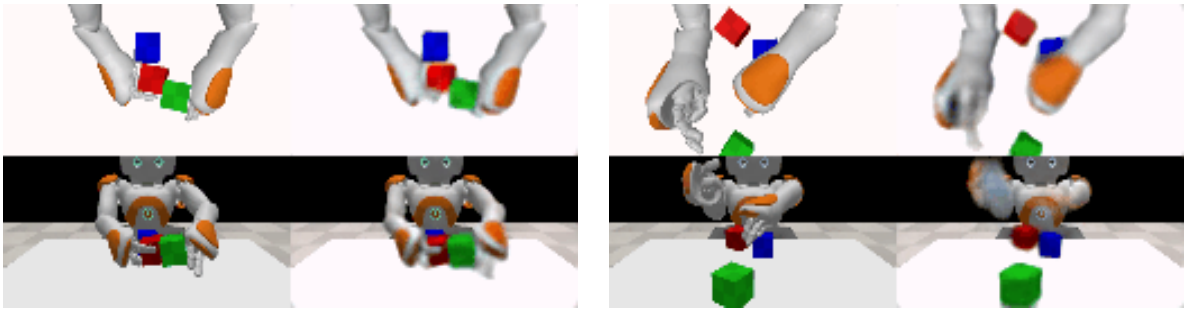
Figure 5.11: Evolution of the agent's behaviors over the course of training. We observed a clear progression toward more intentional interactions. The agent exhibits seemingly random touches on objects during the first million steps. From this point, between 1 and 2 million steps, the agent begins to lift objects, and between 2 and 3 million steps, it progresses to throwing them upward. These behaviors suggest increasing complexity and intentionality in the agent's exploration of the environment.

tal driver in autonomous development. We have computationally demonstrated, albeit rudimentary, a curious agent whose exploration cycle closely mirrors early childhood development. Our agent evolves within the environment through a progressive curiosity, intensifying over time and leading to more complex states. Furthermore, in all agents, at a certain point during training, behavior tends to converge into repetitive motor actions. In this case, the action of tossing cubes upward resembles exploratory behavior in children between the ages of 1 and 3, who repeatedly throw objects to the ground, observing them to investigate physical variables of the world, such as gravity. In this way, we observe that the predictive accuracy of the StatePredictor limits the motor development potential of the agent: the greater the ability of this network to accurately anticipate the world's future state, the higher the likelihood that the agent will exhibit more complex, goal-directed manipulative behaviors, highlighting the critical role of imagination and curiosity in the development of structured actions within the environment.

Our agent could have developed more complex behaviors if it had been able to predict finger movement accurately. However, we were unable to improve this prediction solely through architectural modifications. We attempted to adjust the configurations of the convolutional networks to capture finer details, but unfortunately, this approach was insufficient. In contrast, improving finger prediction may be linked to incorporating tactile information into the observation prediction. In addition to predicting the next frame, the agent could also begin to estimate tactile information, which would help improve the accuracy of finger movement prediction. However, before proceeding with experiments along this line, we will first compare all agents tested so far to establish our baseline.

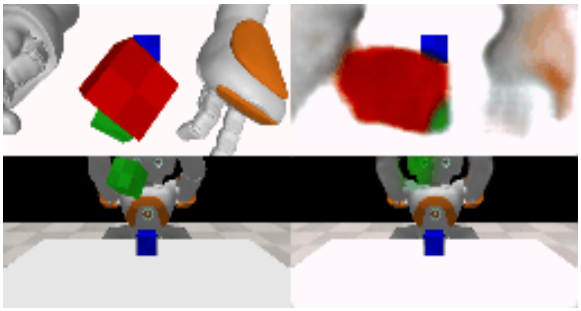
5.2.3 Baseline Agent Selection

In this section, we compare the performance of all developed agents and include the GDN+MSE agent to provide a comprehensive analysis covering all investigated parameter



(a) Step 2,000,000

(b) Step 2,999,040



(c) Step 2,999,051

Figure 5.12: Sample pairs from episodes at different training steps of the GDN+SSIM agent. In each pair, the left image represents the scene provided by the external environment at time $t + 1$, while the right image shows the agent's prediction of this scene at time t. Under ideal conditions, the two images would be identical. In (a), at the 2-million steps, the agent has already learned the shapes of static and dynamic elements and the objects' colors in the environment. Besides, it can accurately predict its body movements, including the arms and parts of the hands, which previously posed challenges. At this stage, the agent focuses on interacting with objects in various ways. In (b) and (c), at the 2.9-million steps, the agent begins to throw objects upward and encounters some difficulty in fully predicting the downward trajectories of the red and green cubes, suggesting that the effect of gravity presents a more complex learning challenge. For more details, watch the video.

combinations. To establish the baseline agent, we evaluated the learned predictions based on criteria related to the accuracy of environment element predictions, the learning of body movements and structure, and the emergence of object manipulation behaviors during exploration. These elements were essential for demonstrating the agent's motor development and autonomy. In our analysis, we constructed Table 5.2, highlighting the topics fully learned by the agents, those in which predictions consistently matched the real environment across all frames of the last fifty training steps. We also classified partial learning cases where predictions were only accurate in specific frames. Moreover, we considered the entire training period for the behavior analysis, as the agent can develop different behaviors over time.

We identified four primary dimensions as the most relevant for the agent's learning process: environmental aspects, including object color, shape, and dynamics; body-related aspects, encompassing motor control and accurate perception of body shape in resting and

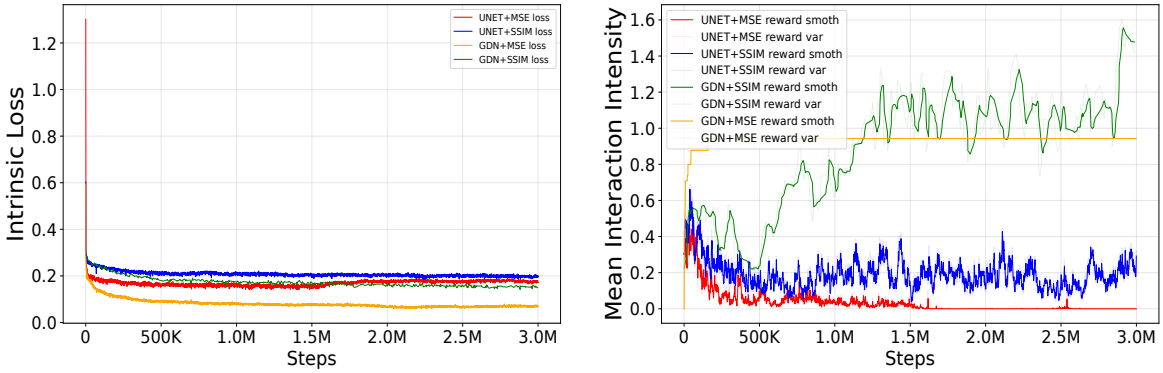
moving states; and emergent behaviors observed during exploration. Regarding behaviors, we analyzed the presence of six essentials for performing more complex tasks: touch, hold, lift, drag, throw, and put. We define *touch* as any contact made by the agent using finger phalanges in a single time step t. The *hold* behavior involves the agent touching the object using finger phalanges for more than one time step t. The *lift* behavior involves lifting the object from the table. *Drag* refers to transporting an object from one point to another across the table surface after touching it and holding it for more than one time step t. In the *throw* behavior, the agent throws object off the table. Finally, the *put* behavior occurs when the agent touches the object, lifts it, keeps it suspended for a moment, and then places it back at another point on the table.

Comparing the obtained results, we observed that the tested agents, rewards, and loss functions significantly influenced the agent's prediction capabilities and exploratory behavior. As shown in Table 5.2, all agents fully learned object-related features such as color and shape. Regarding object dynamics, all agents achieved complete learning except for the GDN agents, whose learning was partial. This classification was not due to a weaker understanding of dynamics compared to the other tested agents but rather because these agents were the only ones to explore the z-axis. By throwing the cubes upward, they occasionally failed to accurately predict the entire trajectory of all objects during their descent. However, concerning the x and y axes, extensively explored by the other agents, the GDN agents demonstrated even more precise dynamic trajectory predictions. From this perspective, they can be considered more complete than the others.

Additionally, the GDN agents were the only ones who exhibited more behaviors during exploration. It not only performed touches on the cubes but also lifted them off the table using both hands and throwing them upward. In contrast, the UNET+MSE agent was limited to making brief touches with their fingertips. The graph in Figure 5.13 (b) confirms this behavior, showing that the GDN+SSIM agent displayed the highest level of object engagement during training, with GDN+MSE in second place. We believe that the GDN+SSIM agent could develop more complex behaviors and the highest mean interaction intensity because it learned other aspects of the environment and its own body more efficiently than other agents, suggesting that incorporating the GDN mechanism and the SSIM metric benefited our approach in multiple ways. All agents faced challenges in predicting the shape and movement of their fingers. However, using GDN proved beneficial, as only agents configured with GDN could partially predict the dynamics and shape of

Table 5.2: A summary of the main aspects learned by the agent throughout training is organized into three primary categories: environmental elements (objects), internal agent features (body movements and shapes), and exploratory behaviors observed during object interactions. Items marked with a check (✓) represent fully learned aspects, those indicated with a circle (○) are considered partially learned, while empty items were not learned during training.

Agent	**Objects**			**Body Movements**		**Body Shape**				**Behaviors**					
	Colors	Shape	Dynamics	Arms	Hands	Head	Torso	Arms	Hands	Touch	Hold	Lift	Drag	Throw	Put
UNET+MSE	✓	✓	✓	○	○	✓	✓	✓		✓					
UNET+SSIM	✓	✓	✓	✓	○	✓	✓	○		✓	✓		✓		
GDN+SSIM	✓	✓	○	✓	○	✓	✓	✓	○	✓	✓	✓	✓	✓	
GDN+MSE	✓	✓	○	✓	○	✓	✓	○	○	✓	✓	✓	✓	✓	



(a) Intrinsic loss

(b) Mean interaction intensity

Figure 5.13: Training results of our agents, using the UNET, GDN in the StatePredictor architecture; and MSE, SSIM in the loss function and reward. In (a), we observe the $\mathcal{L}_{SP}$ losses, which converge to a small residual error, reflecting aspects of the environment that cannot be predicted. All losses converge to similar residual values. In (b), we see the curve of the agent's interactions with the objects on the table; it is evident that the agent using GDN+SSIM learned more easily about various aspects of the environment and its body, resulting in more frequent interactions with the objects.

finger phalanges. Conversely, employing the MSE loss, particularly when combined with the UNET architecture, destabilized the agent's body predictions. This configuration tended to blur all images of the top-down view of the table, placing the agent's arms in multiple locations simultaneously.

The most frequent behavior exhibited by all agents was touching objects, indicating that they developed simple environmental exploration skills but still need to acquire advanced motor manipulation skills. Behaviors such as lifting, throwing, or putting objects on the table were absent in half of the agents. Only the GDN agents showed evidence of learning the lift and throw behavior, suggesting an intriguing synergy between GDN and different losses, providing more training stability and facilitated learning. The agent's interactions, as illustrated in Figure 5.13 (b), reveal that the GDN+SSIM agent achieved the highest number of interactions, resulting in better prediction performance and higher-quality generated images. However, despite this agent's ability to develop the throwing behavior, its execution remains primitive, as it can only perform the action when the cubes are positioned at the center of the table and throws all cubes simultaneously, lacking the ability to select a specific cube.

Based on the results obtained, we conclude that the GDN+SSIM agent is the most suitable candidate for our baseline, making it beneficial to retain all its components in subsequent sections of this work. Besides, the results confirmed our first hypothesis, which proposes that a complex robotic agent trained to model the world in an object manipulation environment can accurately predict both the dynamics of the external environment and its behavior.

5.3 Enriching the Agent's World Model

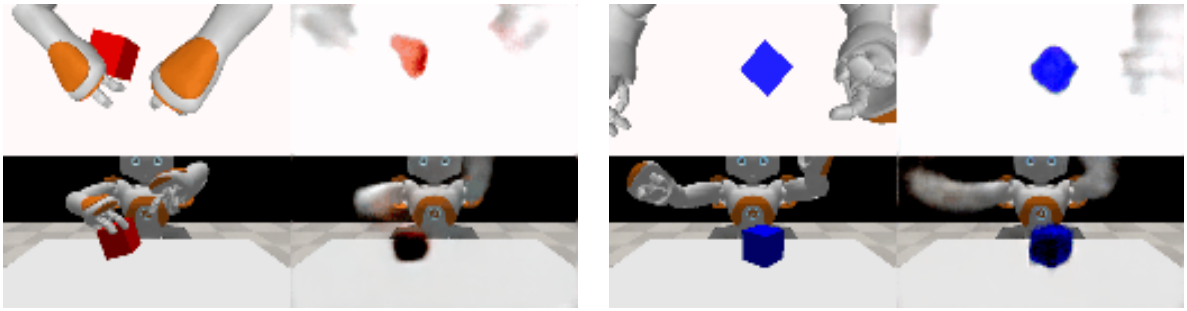
Despite the promising baseline results, the agent was trapped in a local minimum. It cannot completely predict the fingers' movement and the objects' trajectory on the z-axis. As a result, it repeats the same movements for several steps, encouraged by the curious reward function that keeps it exploring the same states repeatedly. Without being able to learn the movements of its fingers properly, the agent spent the last 1 million steps of training repeating the movement of throwing the cubes upwards and, even so, he reached the end of the training without being able to completely predict all the effects that this action cause on objects and its hands. This behavior prevents progress to states requiring more complex motor coordination, as the curious reward function changes the agent's focus only when it can predict the current states appropriately.

The observed behavior suggests that, with the resources used so far, the agent has reached the limit of its exploration. To check whether these difficulties were related to the scenario's complexity, we executed a new training on a scene containing just one cube on the table. We hypothesized that by reducing the number of elements in the scene, finger learning would be facilitated, as would object selection, as there would be fewer elements to predict, allowing the agent to evolve towards more sophisticated manipulation behaviors. Furthermore, to stimulate the agent's curiosity regarding the object, we configured the cube to appear randomly and assigned it one of the colors: red, green, or blue.

To evaluate whether changing the scene impacted the agent's learning, we kept all hyperparameters identical to those used in the baseline training. The results indicated that, at the beginning of training, the agent interacted with the cube simply, making rapid touches. Over time, it learned to predict the cube's trajectory on the x and y axes. However, after mastering these dynamics, the agent gradually lost interest in the object, interacting with it less and less as training progressed, as shown in Figure 5.14. Despite mastering the cube's dynamics, the agent could not lift it to explore the z-axis due to a lack of dexterity. This task was more manageable in the baseline scenario, where three cubes on the table allowed the agent to bring them together and lift them using arm support. Manipulating a single cube, however, required greater precision and dexterity in hand movements, which the agent had not yet developed.

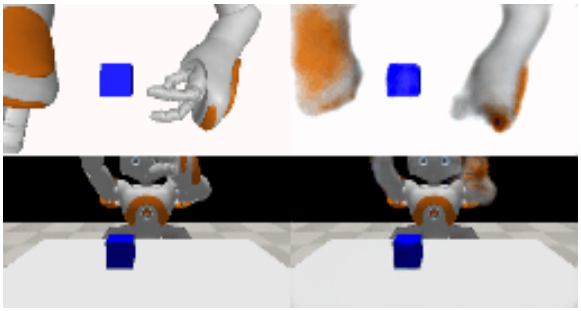
Simplifying the scene further emphasized the challenges associated with predicting finger movements. Once the agent had learned the cube's dynamics and shape, it lost interest and shifted its focus to exploring its own hands. During the second half of the training, the agent repeatedly raised their hands, prioritizing the refinement of finger movement predictions while neglecting interactions with the cube. This behavior demonstrated that the agent became trapped in a local minimum of curiosity, hindering its ability to progress toward more complex behaviors. While this strategy led to a modest improvement in the accuracy of finger movement predictions, it was insufficient for the agent to develop enhanced hand dexterity or to perform more advanced manipulations involving the cube.

As Section 5.2.2 mentioned, adjusting the parameters and layers of the model's CNN was insufficient to improve these predictions. Rather than the architecture itself, we believe that using third-person vision and a single sensory modality significantly impacted



(a) Step 102,528

(b) Step 748,416



(c) Step 1,499,136

Figure 5.14: Sample episode pairs at different training steps. In each pair, the left image represents the scene provided by the external environment at time $t + 1$, while the right image shows the agent's prediction of this scene at t. Under ideal conditions, the two images would be identical. In (a), the agent interacts with the cube and has already learned the environment's shapes, dynamics, and colors of static and dynamic elements. However, it can not accurately predict its hands and fingers' form and movements. In (b), the agent stops interacting with the object and raises its hands to improve the prediction of hands and fingers. The difference between the images in (b) and (c) shows that the prediction quality improves; however, it is still insufficient for the agent to fully learn this state and advance in exploring more challenging states.

intrinsic curiosity, leading to the observed results. Third-person vision caused the agent to focus primarily on predicting its own body. However, humans do not learn about themselves using third-person vision. Instead, they rely on first-person vision, are oriented toward observing the external environment, and learn about themselves through direct interactions with their surroundings, exploring how their movements affect the world around them. Another factor is that with vision as the sole sensory modality, objects occupy only a small portion of the image pixels compared to the arm movements, directing curiosity toward other distracting elements.

Vision is an essential sensory modality, providing a broad range and variety of sensory information for learning dynamics and spatial relationships among elements. However, vision does not encourage more advanced manipulative actions. In the manipulation objects scenario, tactile information plays a critical role, as tactile senses are intrinsically tied to the interest in exploring objects. For example, during infancy, humans instinctively touch various surfaces to learn to differentiate materials and textures. Incorporating first-person vision, integrating tactile information, and redefining the reward function

as a multimodal reward based on visual, tactile, and proprioceptive information could significantly enhance our agent's finger predictions and autonomous development. These changes could enable the development of more sophisticated manipulation behaviors, as they would make the agent more embodied and immersed in the physical environment.

5.3.1 Multimodal Curiosity

To implement the multimodal curiosity, we start by adding the collision information in the observation $\mathbf{x_t}$ as follows

$$\mathbf{x_t} = \Big[\mathbf{p}, \mathbf{c}, \mathbf{i^t}, \mathbf{i^f}\Big]. \tag{5.13}$$

As a result, the agent's observation space now consists of a 16-position collision vector $\mathbf{c}$, the proprioception vector $\mathbf{p}$, and the top and frontal images $\mathbf{i^t}$ and $\mathbf{i^f}$. The collision vector is binary, and each position indicates the occurrence of a collision on one of the finger's phalanges. The state $\mathbf{s_t}$ remains represented by a stack of three observations, with one simulation step skipped.

To decode the multimodal encoder into information for the proprioception and collision decoders, we add two 1D convolutional layers of 72 × 12 channels and 12 × 1, then a flattened layer, and two linear decoders, one for each sensory modality. The proprioception decoder has a linear layer with 66 neurons at the input and 26 at the output. The collision decoder has 66 neurons at the input and 16 at the output, followed by a sigmoidal activation function that formats the output between continuous values from 0 to 1.

We also modified the intrinsic reward function, which is now a weighted function composed of three types of rewards, each type corresponding to a sensory modality as

$$R_{\mathrm{i}} = 1 - \mathrm{SSIM}(\hat{\mathbf{x}}_{\mathbf{i,t+1}}, \mathbf{x}_{\mathbf{i,t+1}}), \tag{5.14}$$

$$R_{\mathrm{p}} = 1 - \frac{1}{1 + \mathrm{MSE}(\hat{\mathbf{x}}_{\mathbf{p,t+1}}, \mathbf{x}_{\mathbf{p,t+1}})}, \tag{5.15}$$

$$R_{\mathrm{c}} = 1 - \frac{1}{1 + \mathrm{SAD}(\hat{\mathbf{x}}_{\mathbf{c,t+1}}, \mathbf{x}_{\mathbf{c,t+1}})}, \tag{5.16}$$

where R_i is the negated SSIM structural similarity metric, R_p is the normalized MSE error between 0 and 1, and R_c is the normalized sum of absolute differences (SAD) between 0 and 1. Then, the intrinsic reward function r_{int} is defined as

$$r_{\mathrm{int}} = w_i R_{\mathrm{i}} + w_p R_p + w_c R_c, \tag{5.17}$$

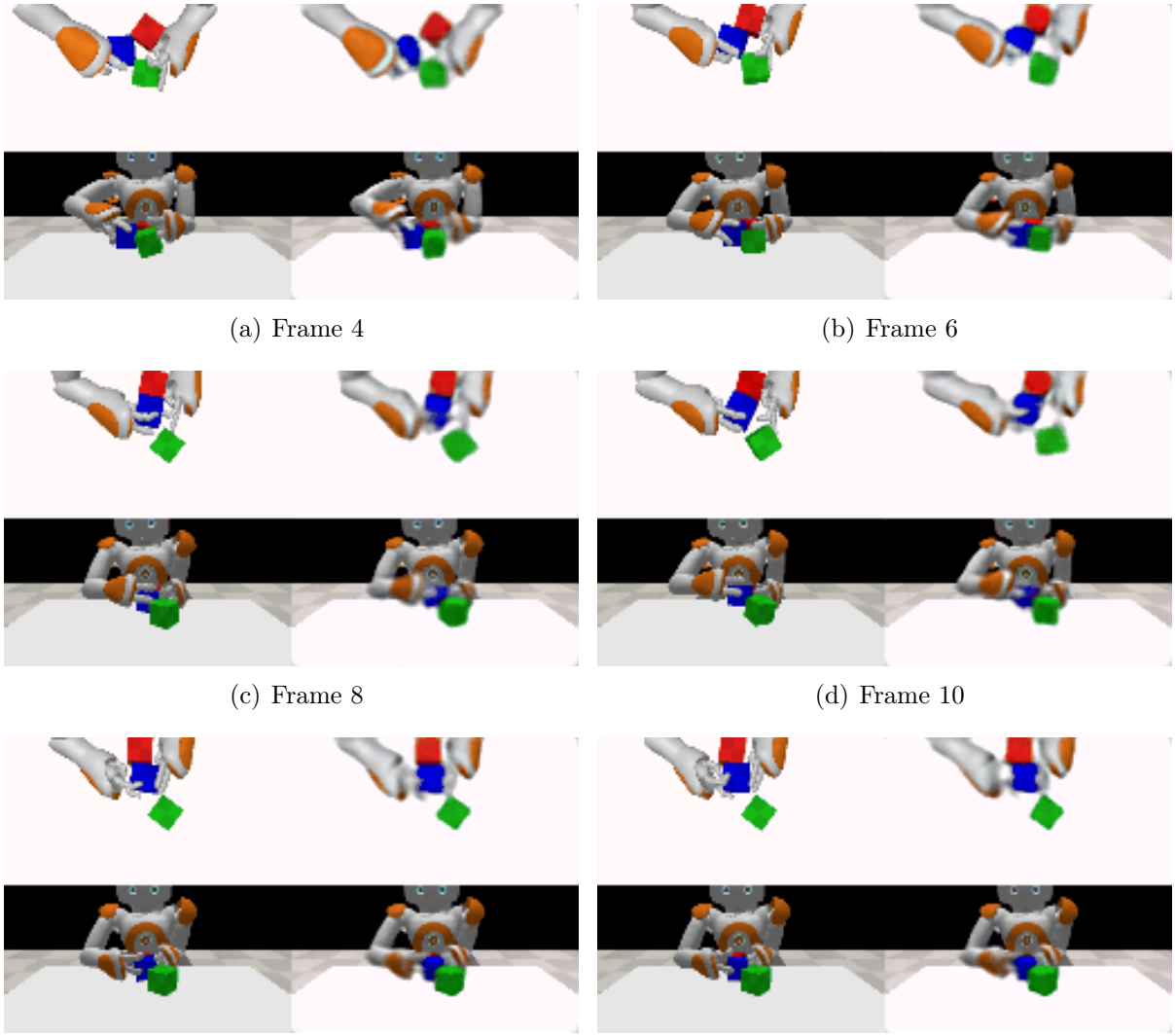
where w_i, w_p, and w_c are weights for image, proprioception, and collision reward functions, respectively. For w_i and w_p, we assigned 0.25; for w_c, we assigned 0.5. We assigned greater weight to the collision reward function to encourage the agent to explore objects more intensely and reflect aspects of children's natural curiosity that constantly rubs their tactile senses on objects to learn about shape, hardness, and texture. Furthermore, we consider collisions the most challenging modality to learn to predict correctly, as they are binary

and change drastically between one step and another, making them instantaneous events and difficult to predict. In addition, there are frequent occlusions, as some phalanges' collisions are not entirely seen in the images. Regarding the loss functions, we continue using SSIM for image prediction and employ MSE and L1 metrics for proprioception and collision predictions, respectively. At this point, we are treating collision prediction as a regression problem.

We trained the agent in the scene containing three cubes, as switching to a single-cube scene did not improve the agent's performance. The results demonstrated that adding tactile information to the state significantly enhanced the visual prediction of finger shape and movement, which had been a challenge for the agent in previous training sessions. This improvement made the learning of the agent's body elements more comprehensive. Moreover, the agent could develop more precise and skillful actions with better hand and finger movement predictions. For instance, it can now grasp an object with its hands and explore it with its fingers over several steps rather than performing only reflexive touches. Figure 5.15 illustrates how the agent accurately predicts the movements and shape of its fingers and investigates objects with its fingers in various ways. We observed that the agent extensively explores the three cubes individually and collectively. Individually, it constantly rubs its fingers over the object and rotates it in various ways during manipulation. When interacting with the cubes as a group, the agent moves them from one side to the other, causing multiple collisions and displacing them across the table. However, the agent has not yet developed sufficient motor coordination to perform more complex actions but has acquired the precision and dexterity to grasp objects effectively.

The agent's behavior during training demonstrated that incorporating multimodal curiosity significantly enhanced its interaction with the cubes. The graph in Figure 5.16 (b) highlights a marked increase in object exploration compared to the baseline results, which relied on curiosity driven by a single sensory modality. Furthermore, the results indicate that the chosen weighting for the reward functions was effective, as the agent continued to learn both visual and dynamic aspects of the scene while developing a heightened curiosity to explore the cubes. Integrating the three sensory modalities was crucial in stimulating and intensifying this interest.

The graph in Figure 5.16 (a) also shows that, at the beginning of the training, the image loss decreases, indicating that the agent is initially focused on learning to imagine the visual aspects of the scene. As the training progresses, these visual aspects are assimilated, and the proprioception and collision losses begin to rise, signaling that the agent is shifting toward exploring new behaviors and interacting with objects differently. However, by the end of the training, these two losses had not stabilized, suggesting that the agent struggled to learn aspects related to these sensory modalities. Due to these challenges, the agent could likely not lift the cubes or perform more complex behaviors. Instead, it remained engaged in activities such as experimenting with hand-object collisions and rotating the cubes, attempting to model the dynamics of collisions without achieving complete success. At this point, including the first-person view can benefit the agent's immersion in the environment. The first-person view can help improve training stability by reducing occlusions and improving the visualization of the table, objects, and hands.



(a) Frame 4 (b) Frame 6

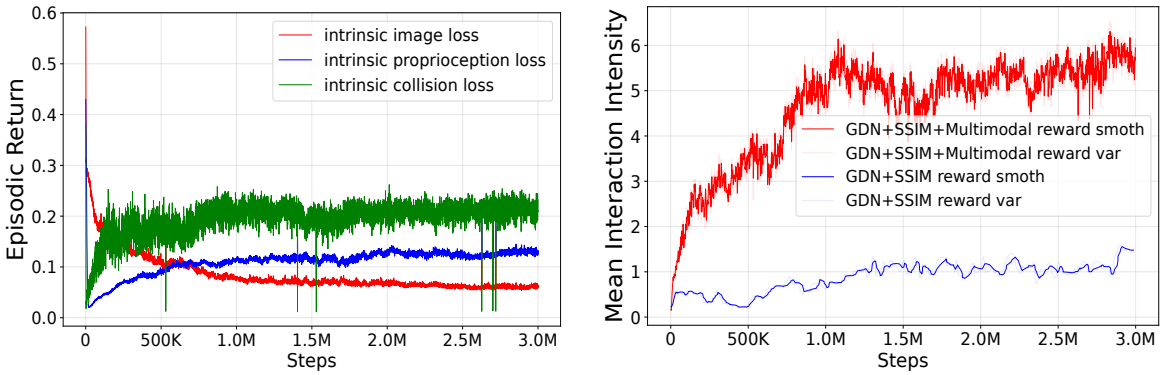
(c) Frame 8 (d) Frame 10

(e) Frame 12 (f) Frame 14

Figure 5.15: Sample pairs from the last episode of the agent's training. In each pair, the left image represents the scene provided by the external environment at time $t+1$, while the right image shows the agent's prediction of this scene at t. Under ideal conditions, the two images would be identical. The sequence of images illustrates the agent's ability to accurately predict the movements and structure of its hands and fingers. Incorporating the multimodal reward enhanced the dexterity of its movements, enabling the agent to select a specific object and investigate it slowly with its fingers, handling it with precision.

5.3.2 First-Person Stereo Vision

We expanded the agent's embodiment by combining multimodal curiosity with first-person stereo vision. Instead of positioning one camera in front of the table and another above it, we placed the cameras near the robot's eyes, each with a 90-degree field of view, and angled them downward to allow the robot to see the entire table without needing to move its head. Figure 5.17 shows that each camera captures the same scene from slightly different perspectives. For simplicity, we chose not to enable the robot's neck movement, as this would require controlling two additional actuators and would complicate isolating the



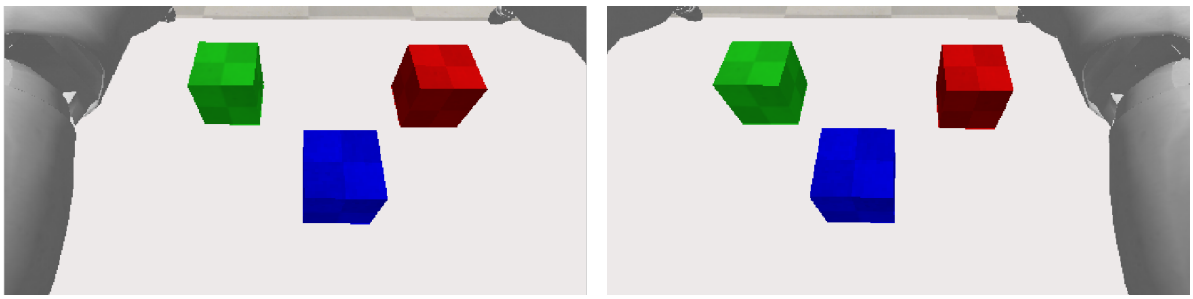
(a) Decomposed intrinsic losses

(b) Mean interaction intensity

Figure 5.16: Training results of our agent using the multimodal curiosity reward function. In (a), the decomposed losses for $\mathcal{L}_{SP}$ are shown. The image loss converged to a small residual error; however, the proprioception and collision losses did not converge as expected. In (b), the agent's interaction intensity with the objects on the table is shown. The agent, guided by the multimodal curiosity reward, shows a higher interest level in exploring objects.

specific impact of first-person vision on autonomous development and the proprioception and collision losses.

To ensure a fair comparison between the agents, we maintained all the hyperparameters used in training the multimodal agent. The results showed that the agent had no difficulty predicting the first-person images. However, it exhibited a higher residual error than the purely multimodal agent, as shown in Figure 5.19 (a). The agent successfully learned the cubes' dynamics, colors, and shapes, as well as its arms and hands (Figure 5.18). This increased error is attributed to the slightly more blurred nature of the generated images and the inherent differences between the images produced by the two agents, which makes direct comparisons challenging. For instance, in the multimodal agent, the third-person images contained many static elements, with only a few pixels changing between frames, corresponding only to the robot's arms, hands, and object movements. In contrast, the first-person images in the current agent included a significantly higher number of pixels



(a) Left camera

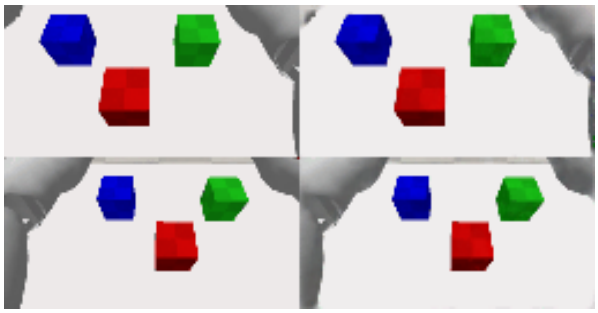
(b) Right camera

Figure 5.17: The robot's stereo vision, with the neck fixed. Each camera captures the same scene from slightly different angles. The cameras were adjusted to give the table and arms a full view without moving the neck.

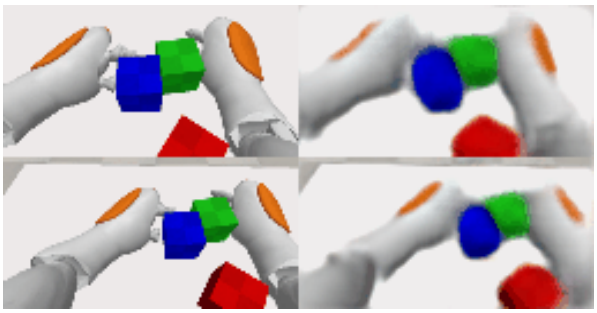
changing between frames due to the movement of both the objects and the hands, making the prediction task much more complex.

Regarding collisions, using first-person cameras proved highly beneficial in reducing occlusion and enhancing the agent's ability to predict them. At the beginning of the exploration, the collision loss started to diverge, reflecting the moment when the agent began interacting with the cubes more frequently and receiving new inputs, as shown in Figure 5.19 (c). After approximately 200,000 training steps, the loss began to decrease, indicating that the agent was learning the dynamics of collisions. This resulted in a consistent reduction in the loss until it stabilized into a plateau for the remainder of the training. During this plateau phase, the agent frequently rubbed its hands on the objects to further minimize the loss. However, it was unsuccessful in maintaining this behavior until the end of the training process.

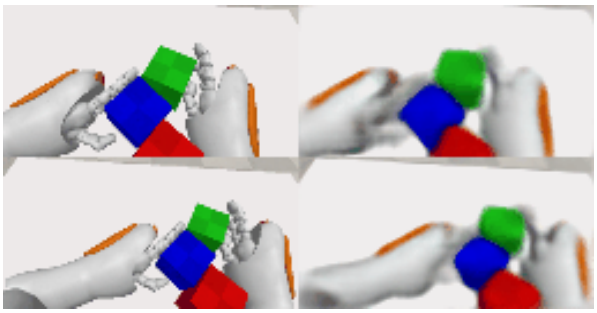
During training, we observed that the agent maintained dexterity and precision in object manipulation, skills previously achieved with the inclusion of multimodality. However, in this experiment, beyond intensively exploring the cubes with its fingers, the agent occasionally attempted to explore the z-axis, achieving greater success than all previous agents. In particular steps, the agent tried to lift one of the cubes but could not hold it securely between its hands and keep it suspended in the air, causing the cube to slip and



(a) Frame 4

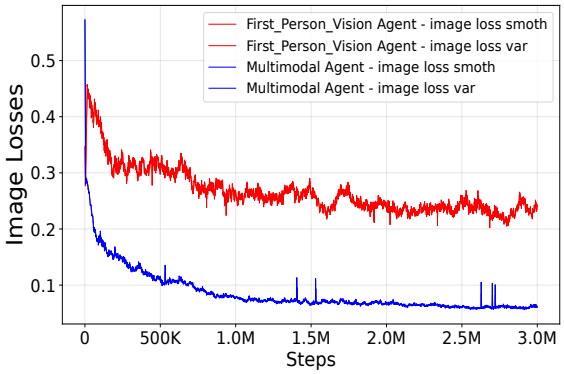


(b) Frame 6

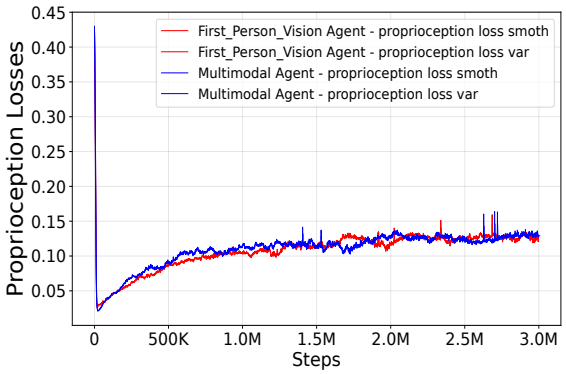


(c) Frame 8

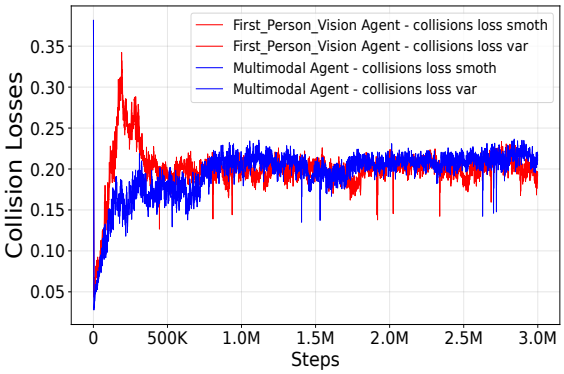
Figure 5.18: Sample pairs from the last episode of the agent's training. In each pair, the left image represents the scene provided by the external environment at time $t+1$, while the right image shows the agent's prediction of this scene at t. Under ideal conditions, the two images would be identical. The sequence of images illustrates the agent's ability to accurately predict the movements and structure of its hands and fingers. The results show that the agent accurately learns various aspects of the external environment using first-person stereo vision.



(a) Intrinsic image loss



(b) Intrinsic proprioception loss



(c) Intrinsic collisions loss

Figure 5.19: The decomposed losses of our intrinsic agent. In (a), the image loss converged but exhibited a residual error more significant than the multimodal agent. In (b), the proprioception loss demonstrates that both agents have similar losses. Finally, in (c), the collision loss reveals that the first-person vision agent with multimodality experiences divergent behavior at the beginning of training and then converges to a plateau. In contrast, the purely multimodal agent shows only a divergent loss.

fall. The agent repeats its attempts without keeping the cube in its hands. We hypothesize that the agent tried to observe the cube more closely to understand the collisions' dynamics better. The motor coordination required for these attempts likely resulted from the agent's partial learning of collision dynamics at the beginning of training.

Regarding proprioception, we observed that the modification to first-person vision did not significantly alter the proprioception loss. The model still diverges, as shown in Figure 5.19 (b). We believe this is due to using 1D convolutions to decode the encoder's information. The filter sizes were significantly reduced, and the proprioception and collision information, compared to the image information, are much smaller in scale and may not be adequately captured during decoding. We hypothesize that adjusting the proprioception and collision decoders could help improve the model's stability and learning performance.

Concerning the agent's autonomous development, we observed significant progress when transitioning from third-person to first-person vision. This experiment marked the first instance where the agent attempted to lift the cubes off the table in a more

structured form rather than relying on reflexive movements. Although the agent has not yet performed this movement entirely due to the lack of integrated coordination between its arms and hands, the fact that it partially learned the dynamics of collisions paves the way for more advanced explorations in this area. Unlocking the neck actuators could significantly enhance this development, allowing the robot to independently adjust its vision and enabling a more precise focus on areas of interest in the environment.

Furthermore, increased movement freedom in the neck would directly impact the agent's intrinsic curiosity. With the ability to explore different angles and perspectives, the diversity of visual information received would increase, allowing the agent to choose what to observe in the environment while eliminating elements that do not capture its interest. This capability could stimulate intrinsic curiosity, encouraging the agent to develop more varied exploratory behaviors. As a result, this expansion in interaction capacity could lead to more creative and strategic behaviors, enhancing the development of motor and cognitive skills throughout the training process. However, unlocking the neck actuators also adds complexity to the prediction problem and reinforcement learning, making the control and training of the agent more challenging.

5.3.3 Active Stereo Vision in First-Person

Implementing active stereo vision required the robot's neck release. This modification increases the complexity of the reinforcement learning training as the state transitions from being fully observable to partially observable. With the freedom to choose where to look, the robot can no longer perceive the entire environment simultaneously. It only observes a limited portion of the scene at any given moment. Also, it cannot fully see its body, relying primarily on its environmental interactions to understand it. Besides, the unrestricted neck introduces new challenges for the curiosity-driven policy, such as determining where the agent should direct its gaze at each time step. Now, the agent must decide where to orient its head and predict what it will see and feel based solely on the current state and chosen action. This task is particularly challenging, as events and actions can occur in the environment without the agent visually perceiving them. These events can influence the agent's imagined perception of the scene, even if they were not directly observed. For instance, the agent might look at one part of the scene while manipulating the cubes with its hands without seeing where the cubes end up. In the next moment, upon shifting its focus, the cubes may be rolling into the newly observed region. To accurately predict the scene, the agent must learn to integrate missing sensory information across modalities and develop sensory synergy, allowing it to coordinate and utilize its different sensors effectively within the environment.

Another important aspect is that the agent can engage more deeply in curiosity-driven exploration of the environment. With limited visual perception, the agent perceives the world from a perspective more like to that of humans, making world prediction more challenging. Introducing first-person vision with a unrestricted neck creates a scenario where the agent must handle constant perspective shifts. Even if it accurately predicts the correct scene elements, any misalignment in rotation or perspective will result in a significant residual error from the visual modality. This error translates into a high

curiosity reward, encouraging the agent to explore the environment further to understand the relationship between its neck movements and their visual impact on the world. Mainly, at the beginning of training, when the agent has no prior knowledge of the environment, they will be even more surprised by frequent expectation violations. Each attempt to predict its future observations based on partial sensory input will generate inconsistencies, intensifying its curiosity-driven exploration. Motivated by high curiosity rewards, the agent may eventually develop more sophisticated manipulation skills than those observed thus far.

Also, observing the same scene from different perspectives significantly increases the variety of samples the agent can obtain from the same environmental configuration. Compared to third-person vision and first-person vision with a fixed neck, first-person vision with a unrestricted neck provides a greater diversity of data, as the agent can observe the same object or region of the environment from different distances and angles. This increased diversity of sensory input can enhance the quality of the agent's learned internal representations, helping it acquire fundamental concepts about world functioning more effectively.

To limit the agent's field of view, we modified the cameras' opening angle from 90 to 45 degrees, adjusting their configuration to restrict the visible area further. Although the cameras' positions were not changed, we tilted their orientations upwards, aiming to replicate the ideal position that allows the agent to view the table in its entirety only if it makes the intentional movement of lowering his head. Neck actuators allow two types of independent movements. The first movement controls the lateral rotation of the head, allowing it to rotate from right to left within a range that varies from angle -119.5 to angle 119.5 degrees. The second movement regulates the vertical tilt of the head, allowing it to move upwards and downwards, also in an interval delimited by the angles from -38.5 to 29.5 degrees, encouraging it to make efficient use of its motor actions to explore and map the surrounding space.

During the experiments, we retained the same hyperparameters as in previous agents, as detailed in Section 5.2, ensuring a fair and direct comparison. This consistency allowed observed differences in behavior and learning to be solely attributed to the new visual constraints and the addition of the movable neck. Additionally, we preserved the baseline architecture and decoders of the multimodal agent with first-person vision and a fixed neck (Section 5.3.2). This decision isolated the effects of the movable neck and restricted field of view on agent performance, enabling an analysis of their impact on environmental engagement, visual exploration, and sensory integration.

The results obtained during training were quite impressive. Initially, the agent's behavior was chaotic, with the neck actuators rapidly moving the head in random directions, exploring all angles, while the arms scattered objects in various directions, indicating a high lack of motor coordination, as shown in Figure 5.20 (a) and (b). Besides, as training progressed, we observed that the agent began to predict the horizon line, establishing a clear distinction between the black background and the gray floor (Figure 5.20 (g)). The agent's behavior gradually became more coordinated and calm with further training. It began to focus its vision on specific points in the environment, shifting from randomly exploring all regions to concentrating its visual resources on particular areas (Figure 5.20

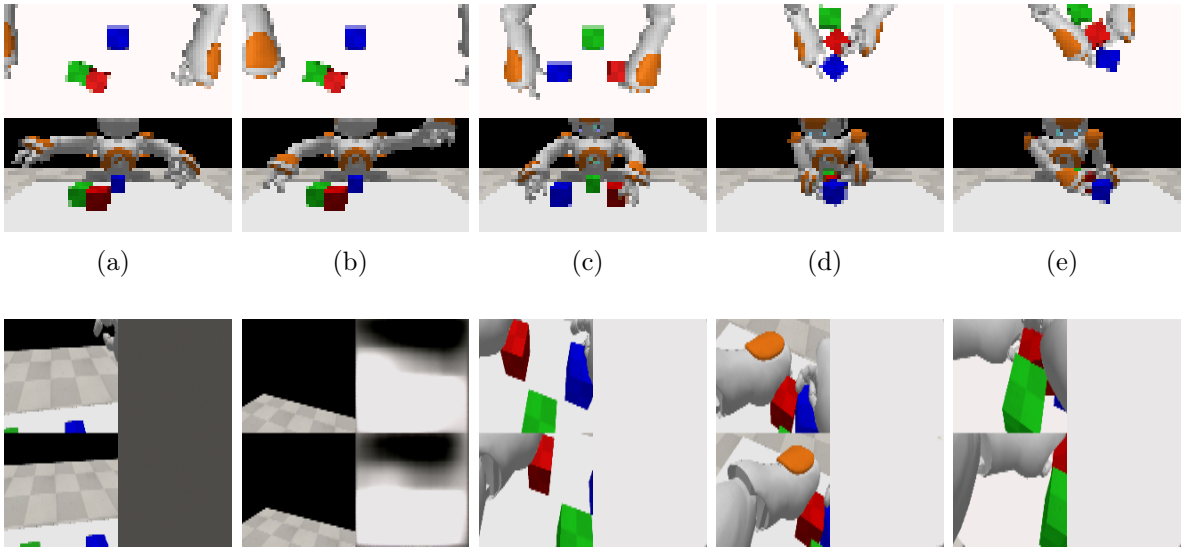


Figure 5.20: Results of training the agent with first-person vision and a unrestricted neck. In each item, the left image represents the scene provided by the external environment at time $t + 1$, while the right image shows the agent's prediction of this scene at time t. Under ideal conditions, these two images would be identical. Images (a) to (e) show the exploratory behaviors developed by the agent from step 0 to step 80,000, as observed through external cameras. Images (f) to (j) show the agent's visual perception and prediction. At the beginning of training, in (a) and (b), the agent exhibits random and uncoordinated behaviors, rapidly shifting its gaze across different parts of the environment in a disorganized manner. In this point, figures (f) and (g) show that the agent begins to differentiate the floor from the background in its predictions. As training progresses from (c) to (e), the agent gradually reduces the speed of its head movements and develops more coordinated motor patterns. It shifts its focus toward objects on the table while simultaneously exploring them with its hands. As sustained attention emerges and the agent's environmental predictions diverge, as demonstrated in (h) to (j), the agent predicts a white frame to represent your visual understanding of the world. Some images have a lower resolution due to processing time during training.

(c) to (e)). Moreover, the movements of the head and upper limbs became more cooperative, with the arms and head operating in the same space in a coordinated manner. This behavior indicated the **emergence of sustained attention** on the objects on the table, accompanied by an increasing interest in touching and investigating these objects, demonstrating improved motor coordination and sensory synergy not observed in previous experiments.

The emergence of sustained attention on objects, coupled with the collaboration between sensors and actuators, represents a significant milestone in this training, as sustained attention to objects is one of the earliest behaviors to emerge in infants, enabling them to focus on objects or events in the external world for detailed and meaningful exploration. Sustained attention is a fundamental skill for lifelong human learning, and in childhood, it provides the foundation for developing sophisticated motor skills, language, and causal understanding. It allows children to comprehend causal relationships and un-

derstand the world, as it enables them to focus on specific stimuli over time, which is necessary for processing complex information. For instance, understanding that pressing a button turns on a light requires focusing on the sequence of events over time. If a child fails to pay attention to an object, they may lose the connection between their action and the consequence, hindering their ability to create causal connections. Children who do not develop sustained attention face challenges in motor, cognitive, and social development, poor academic performance, and difficulty completing complex tasks [31]. The emergence of this capability in our agent, entirely autonomously, demonstrates the potential of our framework to replicate essential aspects of human intelligence development computationally.

The agent's development of sustained attention and motor coordination to simultaneously investigate objects represents a significant milestone, as it indicates that the agent is learning to identify and prioritize areas of the environment with more significant uncertainty or relevance for interaction. This ability enhances the agent's adaptability, allowing it to operate in dynamic and complex contexts. The absence of an extrinsic reward system to model attention suggests that curiosity and the drive to reduce uncertainty can be sufficient for the autonomous development of a complex robotic agent. This behavior parallels what is observed in infant development, where curiosity and the desire for exploration are key drivers in constructing knowledge and skills. Thus, our framework facilitates the natural development of complex competencies, drawing closer to human-like adaptive and creative capacities. It enables the agent to autonomously explore and interpret its environment, generating behaviors that reflect fundamental aspects of our cognition.

However, throughout the training, we observed that the agent failed to imagine the next frame correctly. Instead, it generated white frames representing all the scenes it had seen, as shown in Figure 5.20 (h) to (j). This behavior suggests a significant difficulty in modeling the objects in the environment, prompting us to formulate hypotheses to understand the causes of this behavior and its relationship with attention development. First, we considered the possibility that the neck movement caused the difficulty in learning. We questioned whether the development of attention was a strategy employed by the agent to cope with the complexity of predicting the next frame, which may have led to a reduction in the speed of neck movements and a focus on the objects on the table as an attempt to improve its predictions. If this hypothesis is correct, the question arises whether this attentional behavior would be maintained once the agent learns to predict the next frame correctly.

Another hypothesis is that the difficulty in prediction may be related to the non-stationary target distribution. The dynamic environment, now more complex due to the neck movement, may have resulted in rapid changes in the distributions the model needs to predict, making it more challenging for the agent to keep up with these variations. We also considered the possibility that the issue lies in the normalization. Since it was based on an initial random sampling, it may not have adequately captured the training statistics with the moving neck. Finally, we hypothesized that the agent's embodiment might not be fully optimized. Configurations such as the camera's field of view and observation skipping may have interfered with the quality of perception, affecting the agent's ability

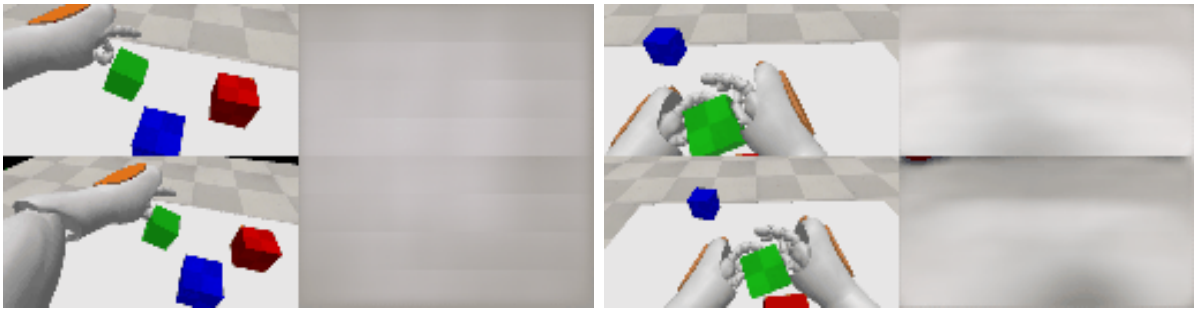
to accurately model the world's behavior.

We first tested whether the normalization was functioning correctly to validate our hypotheses. To do this, we replaced the image generated by the agent with a real image of the environment, normalized it, and passed it through our denormalizer to reconstruct the original scene received from the environment. The result perfectly reproduces the original image, indicating that the statistics used in the normalization and denormalization processes were appropriate. Thus, we ruled out the hypothesis that the model or the normalization was not handling the distribution changes correctly.

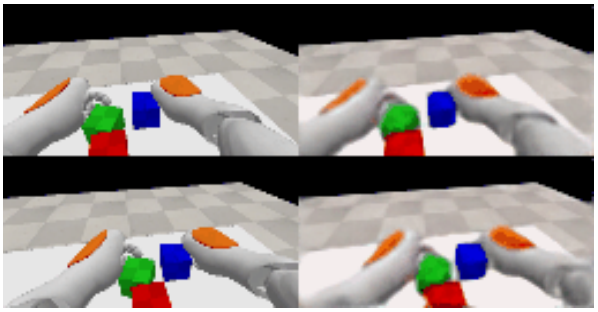
Next, we investigated whether the camera aperture settings and the use of frame skipping were affecting the agent's perception. We examined the camera aperture and found the robot had a very limited field of view. Even when it lowered its head to look at the table, it could not see the sides of its shoulders or torso, which could have hindered its understanding of the events occurring in the environment. Furthermore, when reviewing the impact of frame skipping, we observed that this technique was no longer necessary, with the neck movement unrestrained. With the frame skipping active, the agent lost track of arm movements. For example, when moving a cube, even if it looked around, it could no longer see the tips of its hands, creating the impression that the cubes were moving randomly without the agent's intervention. This issue was more pronounced at the beginning of the training, when the neck speed was very high, causing frame skipping to lead to a loss of causality between the agent's movements and the changes in the environment. By removing frame skipping, this information became available, allowing the agent to perceive its influence on the world when looking around. We also adjusted the camera aperture, increasing the robot's viewing area. However, it still had to lower its head to see the entire table, which caused it to lose sight of the horizon line. Although its vision remained limited, moving its neck sideways or lowering its head could now see part of its torso, arms, and shoulders, making its visual perception more like to a human's.

After implementing these modifications, we retrained the agent, which was then able to model world events accurately, predicting the shape, color, and dynamics of both objects and their arms and hands, as well as correctly reproducing the static elements of the environment, as depicted in Figure 5.21. This result demonstrated that the prediction issue was related to embodiment constraints rather than other previously considered factors. At the beginning of training, the agent still faced difficulties, generating gray frames while rapidly exploring all possible angles with its head and arms. As training progressed, it gradually reduced the speed of its movements and began to distinguish the separation between the floor and the black background. From this point onward, it started rendering the objects on the table and its hands, refining this process. After 100,000 steps, its predictions became more precise. By 1 million steps, it could reasonably model events involving the cubes, their interactions with the agent, and the static elements of the environment (Figure 5.21 (c)).

One of the most striking results of this training was the re-emergence of sustained attention around 100,000 steps, which persisted throughout the entire training process, even after the agent had learned to predict the images accurately. This result invalidates the hypothesis that attention emerged solely as a compensatory strategy to overcome difficulties in modeling scene elements. Instead, it demonstrates that attention arose



(a) Step 1,000 (b) Step 100,000

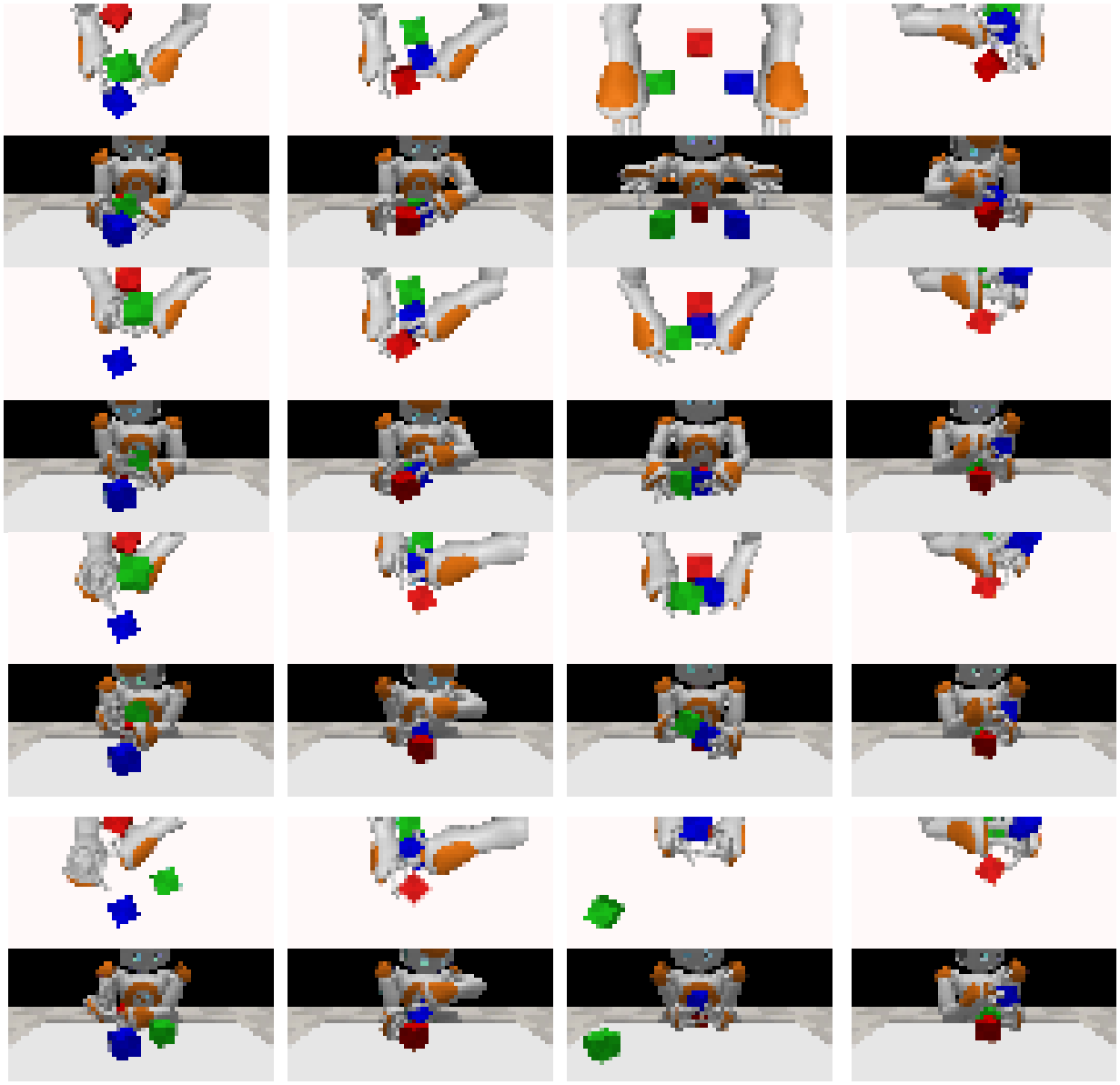


(c) Step 1M

Figure 5.21: Results of our second training with the agent in first-person vision and a unrestricted neck. In each item, the left image represents the scene provided by the external environment at time $t + 1$, while the right image shows the agent's prediction of this scene at time t. Under ideal conditions, these two images would be identical. In (a) is a sample of the agent's prediction at step 1,000 of the training. In (b) is a sample of the agent's prediction at step 100,000 of training. In (c) is a sample of the agent's prediction after one million training steps. The three figures show the evolution of the agent's learning about the environment to perfectly model all the events that occur at the end of the training. For more details, watch the video.

spontaneously from training, highlighting a fundamental aspect of our approach. The fact that the agent maintained this attentional behavior throughout the learning process suggests that our framework naturally promotes the emergence of essential cognitive elements for autonomous development. More importantly, it indicates that we successfully replicated, in computational terms, one of the pillars of infant development, which is sustained attention as the foundation for exploration and active learning of the environment.

In human development, sustained attention shapes perceptual and motor skills. Infants do not merely look at objects; they track them with their eyes, manipulate them in various ways, and integrate them into their motor interactions, learning about their physical properties and spatial relationships. Our agent exhibited an analogous behavior by developing sustained attention toward objects, accompanied by synergistic coordination of its different body parts to explore objects' properties, as illustrated in Figure 5.22. This integration between perception and action is essential for any agent operating autonomously. The successful computational replication of this phenomenon reinforces the relevance of our framework. It opens new possibilities for a deeper investigation of the fundamental mechanisms underlying cognition and autonomous learning in artificial agents.



(a) Trajectory A (b) Trajectory B (c) Trajectory C (d) Trajectory D

Figure 5.22: The trajectory of coordinated exploratory behaviors developed by our agent during training. In (a), the agent investigates objects with its hands, consistently focusing on them before manipulating and tracking their movement when a green cube escapes from the hands and falls on the table. In (b), the agent explores its position on the chair by slowly moving its head from side to side, examining the relationship between its body and the environment while recognizing its arms and shoulders as active interaction components. In (c), the agent refines its object manipulation skills, using its elbows as pivot points on the table to precisely stabilize and inspect a cube. In (d), the agent lifts cubes by supporting them against its own body while visually tracking them, a behavior reminiscent of how infants explore objects by bringing them closer to their faces for detailed examination. For more details, watch the video.

Another important aspect is that, as this agent successfully learned to predict the world alongside the development of sustained attention, it exhibited greater motor coordination than all previously tested agents. Through the use of attention, the agent was able to de-

velop richer and more spontaneously coordinated exploratory actions. The most notable behaviors observed included visually tracking moving objects, investigating the relationship between the body and the environment, executing well-structured and coordinated object manipulation actions, and engaging in creative play with objects. After developing attention, the agent began investigating objects with its hands and consistently looking at them, then proceeded to manipulate and track them as they moved (Figure 5.22 (a)). Subsequently, it started slowly moving its head from side to side to explore its position on the chair, examining the relationship between its body and the surrounding environment while also directing its gaze toward its arms and shoulders, recognizing them as active components of the interaction, as shown in Figure 5.22 (b). An even more intriguing discovery occurred when the agent became aware of the hole in the chair; from that moment, it began engaging in creative interactions, pushing cubes into the hole and then closely observing their movement. This behavior suggests an exploratory curiosity like how infants repeatedly drop objects to observe their fall.

Also, a significant advancement was observed in the quality of object manipulation. The agent developed several highly coordinated lifting and holding strategies, using its elbows as pivot points on the table to precisely stabilize and inspect a cube (Figure 5.22 (c)). This type of postural adjustment is crucial for accurate manipulation and demonstrates motor refinement that emerged without any explicit reward. Additionally, the agent began attempting to lift cubes by supporting them against its own body while tracking them visually to observe them up close, a behavior strikingly similar to how infants explore objects by bringing them closer to their faces to examine them from different angles (Figure 5.22 (d)). This level of coordination between visual perception and manipulation suggests that the model successfully learned essential elements of autonomous learning.

From the point of view of embodied cognition theories, the fact that our agent has developed autonomous head control and can direct its gaze to regions that facilitate the execution of its tactile actions is a fundamental result. The embodied cognition literature has always studied the coordination between vision and action, as it reinforces the hypothesis that the body influences and actively participates in cognitive processes. In humans, coordination between vision and action is essential in reducing cognitive load, allowing the brain to avoid excessive processing of disordered sensory information. Thus, the body organizes and filters information from the environment, delivering aligned multimodal information to avoid brain overload. Similarly, our agent removes the overhead of internal processing from neural networks by spontaneously using its body as an adaptive filter that aligns sensory inputs to facilitate predictions about the world. In this process, the development of more complex manipulation skills emerges.

However, upon analyzing these agent losses, we observed that the StatePredictor still struggles to predict proprioception information accurately, and the collision loss reaches a plateau during training and does not fully converge (Figure 5.23 (b) and (c), in red). This convergence issue may be related to using 1D convolutions with reduced filter sizes. To test this hypothesis, we replaced the 1D convolutions in the collision and proprioception decoders with two linear layers of 2,560 neurons, one for each decoder. This modification significantly improved the stability of proprioception but did not lead to any noticeable

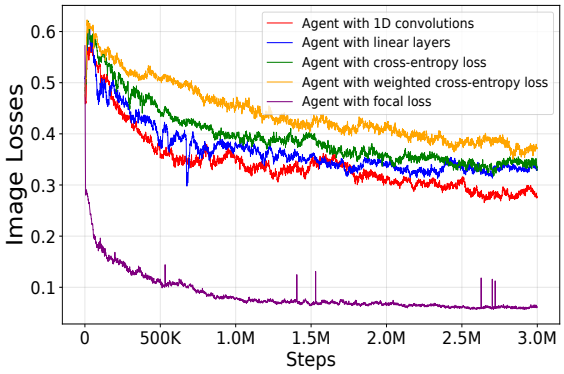
improvement in learning the collision dynamics, as illustrated by Figure 5.23 (b) and (c), in blue.

We hypothesized that replacing the regression loss with a classification loss could lead to more stable collision predictions, as the regressor typically produces intermediate values, such as 0.7 and 0.8 to indicate collisions, and lower values, between 0.1 and 0.3, to represent the absence of collisions, rarely generating exact binary values, which results in higher residual errors. Initially, we transformed the collision prediction into 16 binary classifiers using the cross-entropy loss, but the model's convergence was even worse than with a regressor (Figure 5.23 (b) and (c), in green). Investigating the cause of this outcome, we found that collision prediction is an imbalanced classification problem, as there are more negative (no collision) than positive (collision) examples. Even when the robot grasps the cubes, not all phalanges make contact; some may remain slightly distant from the cubes without triggering a collision, which explains why the maximum values observed in our contact curves always range between 6 and 8 points. To mitigate the imbalance issue, we tested weighted cross-entropy loss, assigning a weight of 0.7 to collisions, and focal loss, a variant of cross-entropy designed to handle highly imbalanced classes. However, the results were very similar across approaches. In all cases, the loss proved even more unstable than when using a regressor to model collisions, as shown in Figure 5.23 (b) and (c), in yellow and purple.

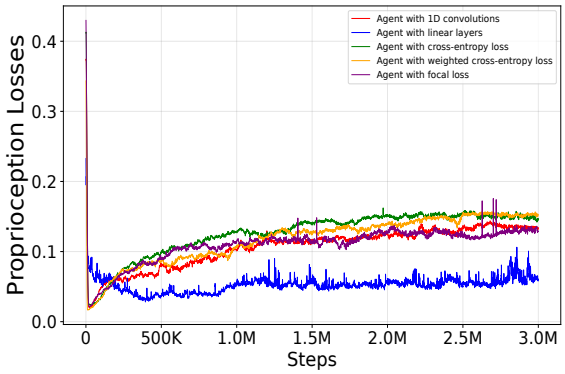
A viable alternative would be the introduction of more advanced tactile sensing in the fingers, such as pressure sensors, reducing the residual error associated with binary prediction. Furthermore, upon deeper investigation, we identified that occlusion remains a significant issue, even with the first-person camera and the agent bringing the cubes closer to its eyes. In many situations, accurately determining which phalanx collided with the cube is challenging, as the fingers can be positioned beneath the cubes, making them difficult to observe from the cameras. We observed that the agent could correctly estimate which hand experienced the collision and, in some cases, even identify the specific finger involved. However, it struggles to determine the exact phalanx, often predicting that the collision occurred across the entire hand or finger.

In response to this issue, we reduced the granularity of the collision information by grouping collisions at the finger level. Specifically, we modified the collision vector from 16 positions to 6, where each position corresponds to a collision occurring in any phalanx of a given finger. Additionally, we reverted to treating the problem as a regression task. Despite this modification, the loss performance remained divergent as the agent continued to confuse collisions between different fingers. This led us to consider modeling collisions at the hand level to facilitate convergence. However, such a simplification could compromise the agent's dexterity, potentially reducing its ability to explore objects in detail and execute precise movements.

Despite the encountered challenges, we observed that, regardless of the problem formulation (regression or classification), the agent's autonomous development remained consistent, demonstrating sustained attention and well-coordinated manipulative actions over the objects in all experiments. Since our main objective is to foster the agent's autonomous development, and this objective has been achieved, we decided to postpone the implementation of more advanced tactile sensors for future experiments. Our hypothesis H2:



(a) Intrinsic image losses



(b) Intrinsic proprioception losses

(c) Intrinsic collisions losses

Figure 5.23: The training results of our agent using different decoders and collision loss functions. In (a), the $\mathcal{L}_{SP}$ image losses are displayed. All losses converged but exhibited a residual error. In (b), the $\mathcal{L}_{SP}$ proprioception losses are shown, where only one alternative demonstrates convergence. Finally, in (c), the $\mathcal{L}_{SP}$ collision losses reveal that our agent with linear decoders is a better alternative.

"*A complex robotic agent autonomously develops structured object manipulation behaviors driven solely by the motivation to predict the world.*", was partially confirmed, as we observed that the agent developed structured manipulation behaviors exclusively through intrinsic motivation generated by curiosity, including actions such as touching, holding, lifting and holding, throwing and dragging, with intense visual attention and highly coordinated behaviors. We consider the hypothesis only partially confirmed because the agent could not develop even more complex behaviors, such as putting and stacking objects, among other more structured behaviors. However, it was able to develop valuable skills for any task, such as head control, sensory alignment between different modalities, and dexterity and precision in the movements performed.

Additionally, we confirmed the hypothesis H3: *The embodiment enables the agent's complete immersion in the environment, promoting the autonomous development of more complex object manipulation skills.* The results show that the agent's full immersion in the environment was essential for the autonomous development of more sophisticated object manipulation skills, such as holding and dragging with two hands and lifting and holding with two hands, interesting games of throwing cubes through the hole in the table to

observe their behavior, and interesting games with its body. The next step is to investigate whether, after this immersion, the agent developed generalizable representations of the environment, allowing it to transfer this knowledge to scenarios distinct from the training environment.

5.4 Evaluating the World Model Generalization

In this section, we present the results of the generalization tests applied to the world model learned by the agent. For this test, we used the agent described in Section 5.3.3, as it demonstrated the most advanced autonomous development in object manipulation. This agent reached a higher level of varied and enriching interactions with the objects during training, making it the most suitable for evaluating the generalization compared to the other agents developed in this work.

Our generalization tests aim to assess whether, during curiosity-driven exploration, the agent developed a latent representation in its world model based on general patterns and rules governing the environment or if it merely memorized specific experiences from the training environment. To this end, we employed the intrinsically trained agent with collision regression loss and decoder with linear layers described in Section 5.3.3 and subjected it to novel test scenarios. In these scenarios, we introduced modifications to the scenes observed by the agent and evaluated its predictions for each sensory modality, including visual frames, collision vectors with objects, and proprioceptive vectors imagined by the agent for subsequent time steps. This approach allowed us to investigate how much the agent can leverage the representation learned in its world model to accurately imagine new situations, such as interactions with objects featuring colors, shapes, and positions different from those encountered during training.

Our objective is not for the agent to perfectly predict what will occur in each sensory modality. While accurate predictions may indicate a deep understanding of environmental patterns and a high level of generalization, our primary focus is on assessing whether the agent has abstracted fundamental principles about the environment's functioning. Specifically, we aim to observe whether, during testing, the agent can make plausible predictions about central aspects such as the general direction of movement of a new object, the effect of gravity, the reaction to touching an unfamiliar object, and whether its predicted body movements and collisions are reasonable. We mainly evaluate whether the agent produces hallucinations in its predictions, such as imagining three objects when there is only one in the scene, predicting nonexistent collisions, placing its arms in completely incorrect positions or angles, or inserting elements absent from the scene. Such hallucinations suggest that the agent might be merely memorizing the conditions of the training environment rather than learning generalizable concepts about its operation.

We subjected the agent to five distinct test scenarios (Figure 5.24), organized into two experimental configurations: one with real-environment feedback, referred to as *real-feedback experiments*, and another without this feedback, called *autofeedback experiments*. In the real-feedback configuration, the agent receives, at each step, real-environment inputs to predict the future environment observation. This configuration aims to assess the

agent's short-term generalization capability. In contrast, in the autofeedback configuration, the agent receives real-environment data only during the first five test steps, after which, from the sixth step, it uses its own predictions to generate future environment state predictions. This experimental configuration is more challenging than the real-feedback setup, as the agent relies solely on its internal state to imagine future states of the environment over medium to long-term intervals. Additionally, it may need to manage potential distortions from imperfect predictions, where small errors can accumulate over time.

For each configuration, we designed five test cases to evaluate whether the agent is not merely memorizing the training scene but has learned the fundamental principles underlying the interactions experienced during intrinsic training, enabling it to generalize to new scenarios. For instance, if the agent understands that a cube moves when pushed, it should be able to apply this knowledge to cubes of different colors or other objects with similar properties. In this context, we developed the following test cases:

- **Test 1 (Single Blue Cube)**: In this test, we created a scene with only one blue cube on the table, a color familiar to the agent from training. This test aims to evaluate whether the agent has grasped the concepts related to the dynamics and outcomes of its interactions with each object independently or whether its ability to make accurate predictions depends on the presence of other cubes that were part of the training environment. This assessment will help determine whether the agent has developed an individualized understanding of each object or if its predictions will hallucinate the presence of absent objects.

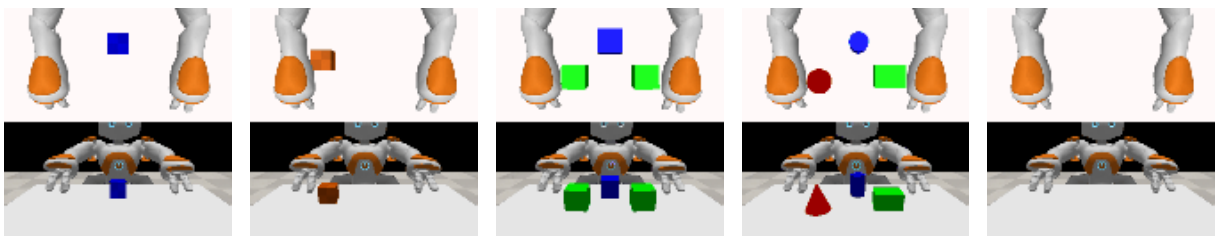
- **Test 2 (Single Orange Cube)**: In this test, the scene contains only one orange cube, a color the agent previously observed during training, but only on parts of its arms, never on objects. This test aims to evaluate whether the agent can decouple the information about color from the behavior of objects. We aim to determine whether the agent will treat the orange cube in its predictions as part of its own body or if it will correctly predict the cube's dynamics as an external object. Another possibility we examine is whether, upon recognizing the orange cube, the agent attempts to assign colors previously associated with cubes during training, as the red cube encountered during training has a color somewhat closer to the orange cube seen during testing. It is also possible that the agent hallucinates the presence of all three cubes initially observed in the training scene.

- **Test 3 (Multiple Cubes)**: In this test, the scenes contain more than three cubes, ranging from four to ten, arranged in a row on the table. All colors used correspond to those previously encountered by the agent during training and are randomly assigned for each test configuration. This experiment evaluates the agent's ability to extrapolate its environment representation to scenarios with more significant visual and interactive complexity. With an increased number of cubes, the agent faces the challenge of predicting interactions among multiple objects, adapting its world model to a more dynamic scenario with potential additional collisions and new movement dynamics arising from the presence of more objects.

- **Test 4 (Different Object Shapes)**: This test evaluates the agent's adaptation to familiar objects with slightly different shapes, challenging its understanding of geometry and interactions. The objective is to assess the flexibility of the agent's world model in handling minor variations in object geometry. The presence of slightly altered shapes tests whether the agent understands physical properties, such as the effect of gravity or the impact of touch, even when objects do not conform precisely to the geometry seen during training. Suppose the agent can correctly predict interactions with objects of new shapes. In that case, it demonstrates a strong capacity for abstraction and generalization regarding object structure, indicating that it has internalized general rules about the environment's functioning.

- **Test 5 (No Objects)**: In this test, the agent is exposed to a scene identical to the training environment but without any objects on the table. This test is particularly insightful as it aims to determine whether the agent anticipates the presence of objects in the scene or can adapt its predictions to an empty scenario. If the agent hallucinates the presence of objects in an environment where none exist, it suggests a direct dependence on the context of the training environment. In an object-free scenario, any movement the agent imagines should be based exclusively on its dynamics and proprioception, allowing us to evaluate whether it can predict its movements without external influences. Moreover, this situation is especially relevant because, in an empty environment, a human would unlikely imagine the presence of nonexistent objects, as this would contradict the rules of reality and common sense.

The quantitative results of all the tests are presented in Table 5.3. Tests 1 and 2 were executed 100 times each, with the cube's position randomly altered on the table for each iteration. For Test 3, involving multiple cubes, we also conducted 100 iterations, incrementally adding one cube to the table every 10 executions. Within each group of 10 iterations with the same number of cubes, the colors were randomly assigned while maintaining the same spatial positions. In Test 4, we performed 100 iterations as well, randomly placing three different objects on the table for each run. The object types were selected randomly, including cylinders, cones, spheres, and cuboids of fixed dimensions. For Test 5, we replicated the same training scene without any objects on the table, executing it 100 times. We perform 100 iterations for all tests to ensure a robust and statistically significant evaluation of the agent's performance. Many iterations provide



(a) Test 1 (b) Test 2 (c) Test 3 (d) Test 4 (e) Test 5

Figure 5.24: Test scenarios for the intrinsic agent.

a more reliable estimate of the agent's generalization ability. To quantitatively evaluate generalization, we employed L1 distance, MSE, and SSIM to measure hallucinations in the collision vector, proprioception, and vision, respectively. To understand the performance drop in generalization, we conducted the same five tests in the training environment, collecting the same metrics and thus enabling a direct comparison between the training and testing scenarios.

The results presented in Table 5.3 show that, in all experiments with the real-feedback configuration, there was little discrepancy between the errors observed in the training and test environments. This fact highlights the model's ability to generalize well in the short term, even in scenarios that deviate significantly from the training distribution, as seen in Tests 3 and 4, which involve multiple objects and objects with different shapes. Another clear result of generalization is observed in the collision errors of Tests 1 and 2, where the agent exhibited lower values than those recorded in the training environment, demonstrating that predicting collisions resulting from a single object was simpler than dealing with three objects, simultaneously, meaning the model was able to decouple collision predictions, clearly understanding the dynamics of a single object when the others were not present. Furthermore, in Tests 3 and 4, the agent had to handle many objects and objects with different shapes. However, the collision errors presented were not higher than those obtained in the training environment, revealing an implicit understanding by the agent of the physical concept of collision, in which touching an external object changes the state of its finger joints.

We observe that vision was the most impacted sensory modality in all test cases under the real-feedback configuration, achieving the lowest similarity in Test 1 (0.51 ± 0.03). Indeed, vision is the most complex sensory modality, as it accurately predicts many visual elements in the scene, such as the position of all objects, their shape, color, and dynamics. In our analysis, we found that in the training environment, the agent's

Table 5.3: The generalization results for each test type in real-feedback and autofeedback configurations. The reported errors are the mean ± standard deviation calculated over 100 test executions. Test 0 is conducted in the same training environment without modifications to the initial positions of the cubes across all 100 test executions. Test 1 involves scenes with a single blue cube on the table, with the cube's position randomly initialized for each test execution. Test 2 follows the same procedure as Test 1 but uses an orange cube instead of a blue one. Test 3 involves multiple cubes on the table, up to 10 cubes. Test 4 incorporates objects of different shapes, such as rectangular objects similar to cubes, cones, cylinders, and spheres. Finally, the results from Test 5 correspond to scenes without any objects on the table.

	Real-feedback (mean ± std)			**Autofeedback (mean ± std)**		
Test	**Image error**	**Col. error**	**Prop. error**	**Image error**	**Col. error**	**Prop. error**
0	0.69 ± 0.03	0.22 ± 0.12	0.15 ± 0.03	0.51 ± 0.03	0.12 ± 0.10	0.79 ± 0.05
1	0.51 ± 0.03	0.03 ± 0.07	0.16 ± 0.04	0.48 ± 0.02	0.24 ± 0.13	0.81 ± 0.09
2	0.63 ± 0.02	0.03 ± 0.07	0.16 ± 0.04	0.49 ± 0.02	0.26 ± 0.12	0.78 ± 0.09
3	0.53 ± 0.05	0.22 ± 0.11	0.16 ± 0.04	0.38 ± 0.05	0.29 ± 0.15	0.90 ± 0.11
4	0.65 ± 0.02	0.23 ± 0.13	0.15 ± 0.03	0.47 ± 0.02	0.36 ± 0.18	0.82 ± 0.09
5	0.60 ± 0.00	0.00 ± 0.00	0.12 ± 0.00	0.46 ± 0.00	0.25 ± 0.01	0.79 ± 0.02

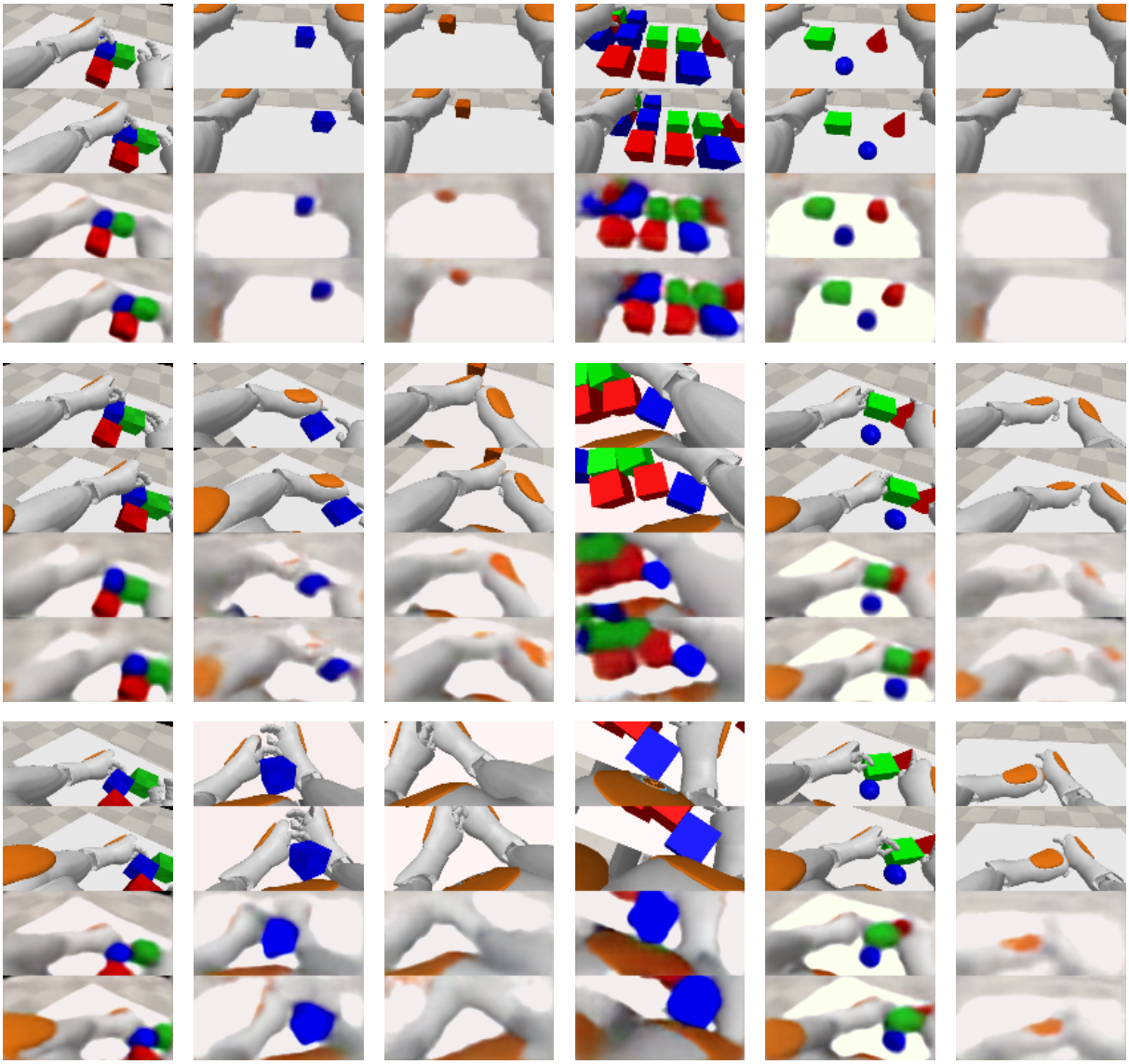
prediction is nearly perfect and free from hallucinations, as shown in Figure 5.25 (a), yet the similarity attributed is 0.69, primarily because the imagined image quality is slightly lower. The imagined image is blurrier, with fewer high-frequency elements, and some shapes are blurred, but there are no hallucinations in the images, which is the most important aspect. In the other tests, no hallucinations occurred either, and the drop in similarity between the images is mainly due to the agent's difficulty in accurately rendering the shape of the arms in certain situations, such as in frame 0 of all tests, where the arms are drawn in a blurry manner.

In the real-feedback configuration, the agent had little difficulty drawing shapes accurately in the test, even cones, spheres, and cylinders, which are slightly different shapes from those seen during training. When these objects were stationary in the scene, the agent could render them almost perfectly, and when the objects were in motion, some subtle deformations occurred, making them appear more cube-like. However, the deformation did not occur with every movement, only with very abrupt movements, as evidenced in Figure 5.25 (e). Moreover, the most noticeable deformations primarily occurred with objects very different from those in the training set, such as spheres and cones. In these cases, although the agent drew the objects slightly inaccurately, it retained its learning about collisions, and it was able to reasonably predict the dynamic trajectories of these objects, showing significant correspondence with their actual trajectories, even without perfectly preserving the shape.

Another significant result shown in Figure 5.25 is the preservation of the learned stereoscopy, meaning that the generated images adhered to the correct camera angles. This behavior indicates that the agent not only generalized to different situations but also adapted its understanding of perspectives and depth when exposed to new scenarios, which is crucial for creating a consistent visual representation of the environment. Furthermore, the generalization of stereoscopy suggests a robust learning of the spatial relationships between objects and the agent, an essential element for tasks involving manipulation, navigation, and refined spatial perception.

The results presented in Table 5.3 also revealed significant differences in the agent's performance between the real-feedback and autofeedback configurations, highlighting a greater difficulty for the agent to generalize in the autofeedback configuration. Collisions were not significantly impacted, suggesting that this sensory modality exhibits greater robustness and generalization capacity under challenging conditions. However, vision and proprioception showed greater sensitivity. Proprioception was the most affected, with the error increasing from (0.15 ± 0.03) to (0.79 ± 0.05), even in Test 0. Meanwhile, the similarity in the images decreased from (0.69 ± 0.03) to (0.51 ± 0.03), indicating that the autofeedback configuration also negatively impacted visual generalization, though to a lesser extent.

We observed notable differences when visually comparing the images generated under the autofeedback configuration (Figure 5.27) with those generated under the real-feedback configuration (Figure 5.25). The images produced with autofeedback exhibited a higher degree of distortion in the arms and hands, particularly in the final frames of the sequence, where accumulated errors were more pronounced. Furthermore, the agent struggled to accurately imagine the arms and head angles (Figure 5.27 (a), (b), and (c)), especially



(a) Test 0 (b) Test 1 (c) Test 2 (d) Test 3 (e) Test 4 (f) Test 5

Figure 5.25: Qualitative results of generalization for the visual sensory modality using the real-feedback configuration. The tests carried out are presented in columns. For each test, we present three rows: the first row corresponds to frame 0, the second to frame 6, and the third to frame 14. In each row, the first frame corresponds to the environment's real frame, and below it is the frame generated by the agent (under ideal conditions, both frames would be identical). The results are quite promising, with good predictions from the agent, even in the last frame of the sequence. However, it is observed that the agent struggles to imagine finer image details, such as the finger region, and tends to generate blurrier images with fewer high-frequency details, resembling images that have undergone Gaussian filtering. This factor contributes to the similarity between the real and imagined environment not exceeding 0.69.

in the later frames of the sequence. This issue appears to be directly linked to the significantly larger errors observed in the proprioception modality. Since this modality does not accurately reflect reality, its errors conflict with the more reliable information provided by other sensory modalities, thereby compromising the generation of precise

(a) Sample 1 (b) Sample 2 (c) Sample 3

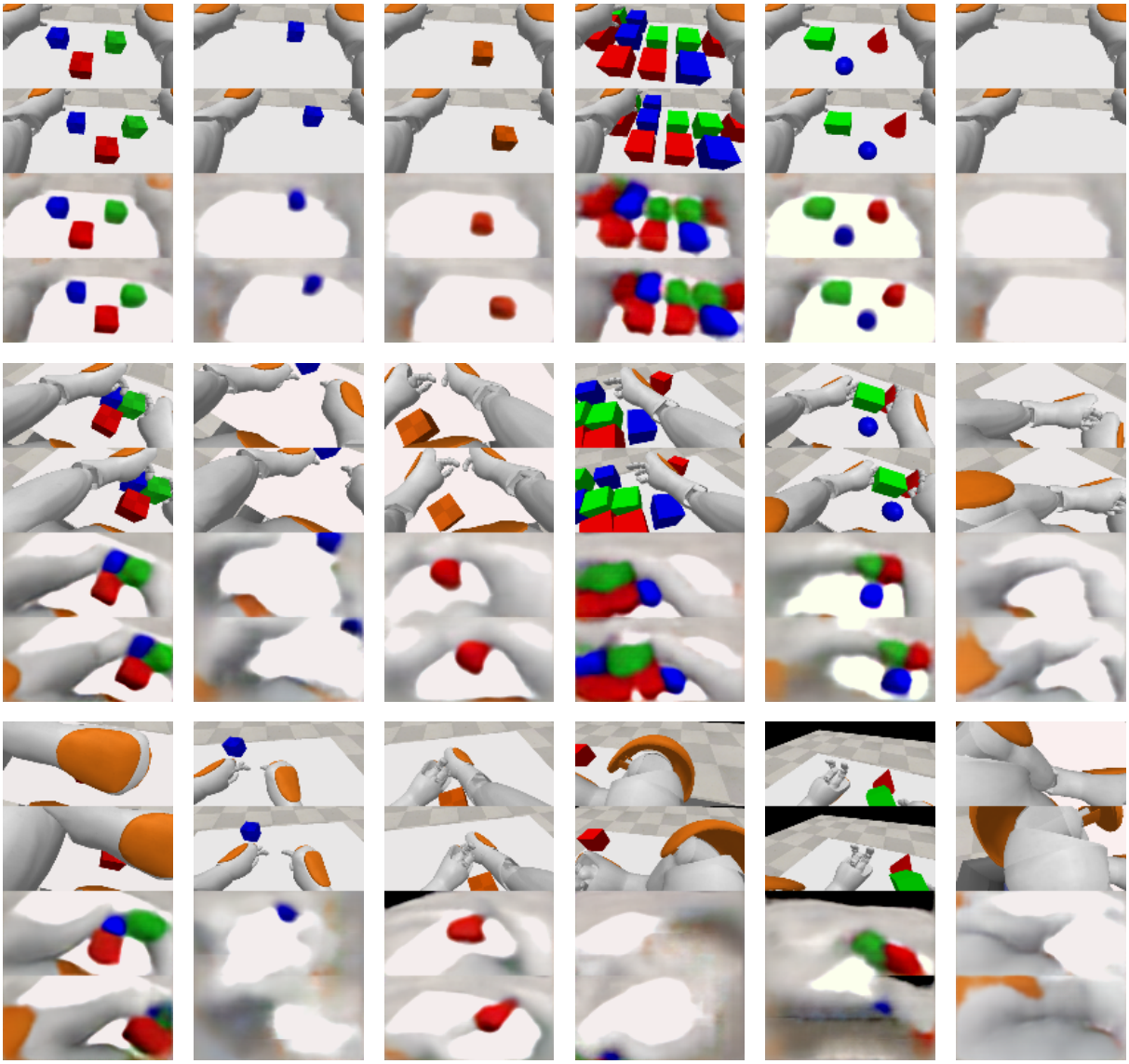
Figure 5.26: Agent hallucinations' samples obtained in Test 1 under the autofeedback configuration. All hallucinations were collected from the last frames of the tests.

images, particularly in aspects related to the agent's body positioning.

Regarding other types of hallucinations, such as including non-existent objects in the scene or misinterpreting objects as part of the agent's body, we analyzed all images generated with autofeedback in Test 1. We observed that in approximately 40% of cases, the agent tended to subtly include green or red patches in the scene, varying in size as if attempting to incorporate elements seen during training (Figure 5.26). This behavior occurred exclusively under the autofeedback configuration and was more frequent when the blue cube raised in positions significantly distant from those in which cubes were placed during training. Since such errors were not observed in the real-feedback configuration, we hypothesize that these hallucinations may be triggered by feedback loops involving inaccurate predictions, highlighting a vulnerability of the model when handling accumulated errors. This fragility likely stems from the agent being exclusively trained with highly precise environmental information.

In Test 2, we observed fewer hallucinations, and those that occurred generally involved attempts to color the orange cube with shades of red in the prediction's final frames or to paint parts of the robot's torso red instead of orange. In rare cases, the agent attempted to insert cubes of other colors into the scene. In Tests 3 and 4, hallucinations were almost nonexistent, with the main issues being deformations in the shapes of objects, particularly during collisions. Finally, in Test 5, no hallucinations of this type were detected; however, the agent experienced difficulties in accurately rendering the arms and hands, likely due to the high errors in the proprioception modality.

Overall, the agent demonstrated good generalization, even in configurations involving autofeedback. Through its predictions, the agent understood fundamental concepts about the environment's dynamics, such as the concept of collisions, which was captured by both the tactile sensory modality and vision. It also showed an ability to comprehend dynamic events, such as the impact generated when pushing one object against another with its hands. For instance, in Figure 5.27 (d) and (e), the agent interacts with multiple cubes and objects of different shapes. During these interactions, it accurately predicts the final positions of the objects despite the accumulated error inherent to the autofeedback



(a) Test 0 (b) Test 1 (c) Test 2 (d) Test 3 (e) Test 4 (f) Test 5

Figure 5.27: Qualitative results of generalization for the vision sensory modality, using the autofeedback configuration. The tests carried out are presented in columns. For each test, we present three rows: the first row corresponds to frame 0, the second to frame 6, and the third to frame 14. In each row, the first frame corresponds to the environment's real frame, and below it is the frame generated by the agent (under ideal conditions, both frames would be identical). The results indicate more difficulty for the agent in predicting long-term frames, such as frame 14 in the sequences. In particular, the agent faces challenges in correctly imagining the arms and head position and accurately representing the arms and hands. This performance is related to the significantly higher errors in proprioception modality within the autofeedback configuration. These errors affect the latent space, making more confused and imprecise visual predictions.

configuration. This behavior suggests that the agent internalized the essence of some fundamental object properties, such as the general trajectory shapes and the effects of applied forces. However, particularly in scenarios with numerous objects, the agent tends to overlook predicting the behavior of some of them. For example, in Figure 5.27 (d), it

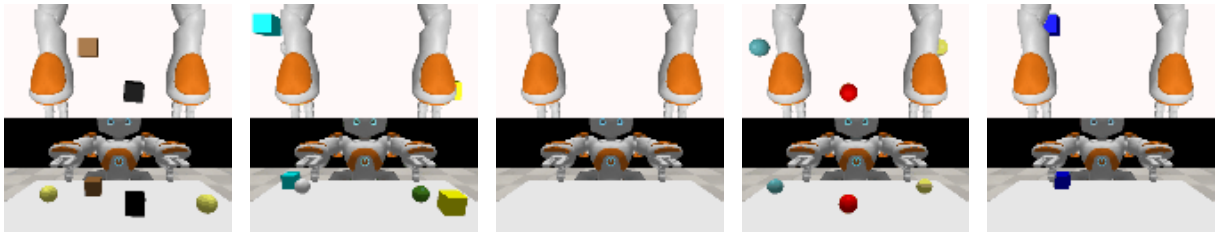
failed to position a red and blue cube, making it appear that the scene contained fewer objects than it did.

Based on the results obtained, the generalization capability of our agent to unseen situations is promising. However, we identified aspects that can be improved to enhance its performance in the test cases, particularly in the autofeedback configuration, where errors were more pronounced and consistent across all five tests. Notably, hallucinations occurred exclusively in this configuration, suggesting a weakness in the model when dealing with accumulated predictions. To mitigate this issue, we propose training the agent with a subset of frames from its own world model. This approach could better prepare the agent to handle noisy or inconsistent inputs during testing, thereby reducing the occurrence of hallucinations. Another critical aspect is the proprioception generalization, particularly in the autofeedback configuration. To address this limitation, we propose diversifying the cubes' initial positions within the environment and incorporating more varied scenarios to encourage the agent to explore its workspace more effectively. Additionally, enriching the training environment with a wider variety of shapes, colors, quantities, and object dynamics could enhance the agent's ability to generalize to more complex situations, benefiting all sensory modalities.

5.4.1 Enhancing Generalization Through Imagined Feedback

In this section, we refine the training protocol to introduce variability in samples to which the agent is exposed. We randomly assign initial positions to all objects on the table, including scenarios where one object may be stacked on top of another. Additionally, we introduce new object colors, such as yellow, magenta, cyan, black, dark olive green, white, light red, grayish blue, light green, light yellow, grayish purple, medium gray, light pink, aqua green, and light brown, to promote the decoupling of color information. We also incorporate training episodes where no objects are present on the table, allowing the agent to focus on learning its body dynamics. The number of objects varies from none to up to four, randomly selected from spheres and cubes, as shown in Figure 5.28. Each scenario is generated before the start of each rollout, determining the number of objects, their positions, and whether stacking occurs, all through randomization. We discretize the table space to prevent overlaps so that each object occupies a unique position. With this strategy, we generate a total of 15,120 distinct scene configurations. With the modifications, Tests 1 and 5 are no longer entirely out of distribution but represent classes seen during training, albeit infrequently. These tests still exhibit a slight distribution shift but are better aligned with the training data. In contrast, Tests 2, 3, and 4 remain significantly out of distribution, as they contain samples never encountered during training and deviate further from the original training distribution.

Due to the long training time, we did not introduce scene variability from the beginning. Instead, we performed fine-tuning on an agent previously trained in the fixed environment, whose generalization results were already presented in Section 5.4. Additionally, we considered restarting training from scratch unnecessary, as the agent had already demonstrated significant generalization capabilities, exhibiting more difficulty in the autofeedback configuration. Therefore, we used the training checkpoint at 2.5 million



(a) Sample 1 (b) Sample 2 (c) Sample 3 (d) Sample 4 (e) Sample 5

Figure 5.28: Samples of randomly generated scenes during training. In each rollout, objects of either cube or sphere type are randomly selected, with up to 18 possible colors, and are placed in random positions on the table.

steps and continued training until reaching 3 million steps, ensuring that the new conditions represented only a tiny fraction of the total training. We observed that the agent had not fully assimilated the new information during these steps. For this reason, we extended the training to 3.7 million steps. At that point, we observed that the losses had stabilized again and that the agent had successfully incorporated the newly introduced information.

In addition to modifications in the training protocol, we also incorporated imagined information generated by the agent to represent the state $\mathbf{s_t}$. During trajectory collection, we randomly selected steps in which the state $\mathbf{s_t}$ was complemented with data imagined by the agent itself. As a result, the sensory information was not entirely accurate. Instead, it is influenced by the agent's world model, similar to how the interplay between sensory input and internal predictive models shapes human perception [84]. This strategy enables the agent to learn how to handle imperfect inputs while refining key errors in its world model through new information. Specifically, in each rollout, we randomly selected a step in which the sensory input was not entirely derived from the environment but complemented by the agent's world model. When a step was chosen, we removed the most recent information from the observation stack and replaced it with data generated by the world model. This procedure was applied across all sensory modalities. The process continued for two additional consecutive steps, ensuring that, in at least one step, the agent had to select an action and predict the next observation based exclusively on self-generated data.

We obtained promising results with the modifications, demonstrating a significant improvement in predictions across most test cases, particularly in the sensory modalities of vision and collision, as shown in Table 5.4. All values highlighted in bold in this table correspond to performance metrics that surpass those of the previous agent, whose results were discussed in Section 5.4. We observed the most notable improvements in the autofeedback configuration, where all test cases exhibited significantly better performance in both vision and collision tasks. We attribute this enhancement primarily to complementing the state representation with information generated by the world model. This approach allowed the agent to handle imperfect sensory inputs better, reducing reliance on purely noise-free data.

Furthermore, the implemented modifications significantly reduced overfitting across

Table 5.4: The generalization results for our agent with imagined feedback and new training protocol. The reported errors are the mean ± standard deviation calculated over 100 test executions. Test 0 is conducted in the same training environment without modifications to the cubes' initial positions across all 100 test executions. Test 1 involves scenes with a single blue cube on the table, with the cube's position randomly initialized for each test execution. Test 2 follows the same procedure as Test 1 but uses an orange cube instead of a blue one. Test 3 adds multiple cubes on the table, up to 10 cubes. Test 4 incorporates objects of different shapes, such as rectangular objects more similar to cubes and cones, cylinders, and spheres. Finally, the results from Test 5 correspond to scenes without any objects on the table. All values highlighted in bold in this table correspond to performance metrics that surpass those of the previous agent, whose results were discussed in Section 5.4.

	Real-feedback (mean ± std)			**Autofeedback (mean ± std)**		
Test	**Image error**	**Col. error**	**Prop. error**	**Image error**	**Col. error**	**Prop. error**
0	0.55 ± 0.00	**0.18 ± 0.08**	**0.13 ± 0.02**	**0.55 ± 0.03**	**0.02 ± 0.03**	0.94 ± 0.10
1	**0.69 ± 0.04**	0.04 ± 0.08	**0.13 ± 0.03**	**0.56 ± 0.02**	**0.01 ± 0.05**	0.84 ± 0.06
2	**0.65 ± 0.02**	0.08 ± 0.01	**0.13 ± 0.04**	**0.59 ± 0.02**	**0.04 ± 0.09**	0.85 ± 0.10
3	0.53 ± 0.04	**0.13 ± 0.11**	0.16 ± 0.06	**0.51 ± 0.04**	**0.07 ± 0.11**	0.90 ± 0.11
4	0.59 ± 0.02	**0.12 ± 0.10**	**0.14 ± 0.04**	**0.59 ± 0.04**	**0.02 ± 0.06**	0.86 ± 0.09
5	**0.73 ± 0.00**	0.00 ± 0.00	**0.08 ± 0.00**	**0.70 ± 0.00**	**0.00 ± 0.01**	0.83 ± 0.03

different tests and the high discrepancies between the real-feedback and autofeedback configurations. The increased sample variability also played a crucial role in this improvement, as it exposed the agent to a broader range of situations, with objects appearing in different positions, thereby fostering a higher level of abstraction and positively impacting collision learning, a sensory modality in which the agent exhibited a substantial reduction in errors. The Test 5 results further support our observations, demonstrating that the agent maintained consistent performance across both configurations, with a similarity of 70% in autofeedback and 73% in real-feedback in the visual modality, suggesting that the agent became less sensitive to noisy samples generated by itself, enhancing its ability to adapt to different input conditions. One area for improvement identified in this agent is that the implemented modifications did not enhance proprioception prediction, which prevented us from achieving even better results.

5.5 Adapting Intrinsic Skills to Extrinsic Tasks

To evaluate whether the motor skills acquired by the intrinsic agent during curiosity-driven exploration are transferable to other tasks, we propose a training environment featuring an extrinsic task. In this environment, a ball appears in random positions and moves across the table, requiring the agent to capture it with both hands. The *capturing-ball task* was deliberately chosen because it demands that the agent utilize and adapt skills already developed during intrinsic training to complete it successfully. Capturing the moving ball requires the agent to demonstrate manual dexterity, track its trajectory with gaze, anticipate its future position, and adjust hand placement to intercept it at the right moment. This process requires sustained attention to the ball, coordinated arm

movements to grasp it, and a firm grip to maintain possession once captured. The agent has already developed some of these skills, either fully or partially, through its intrinsic environment exploration. It has learned to maintain visual attention on the table and moving objects, align perception and action with directing gaze toward hand contact areas, and precisely coordinate arm and head movements to lift objects and keep them in hand for multiple steps. The challenge is effectively repurposing these skills by adapting them to the current task's demands.

In intrinsic training, the agent developed generalizable motor skills as a foundation for solving complex tasks. Similarly, to catch a moving ball or place an object inside a bowl, children must first develop basic motor and perceptual skills that serve as a foundation for more sophisticated actions. Over time, these skills are reused and refined, allowing the child to adapt previously acquired coordination, such as manipulating tools, or opening doors without relearning basic coordination from scratch. In the same way, we argue that during curiosity-driven exploration, the agent constructed an internal representation that encodes fundamental motor strategies, making its skills more flexible and transferable. This enables the agent to adjust only specific components of the learned policy to adapt to new tasks, significantly accelerating its adaptation to different contexts and challenges. In contrast, an extrinsically trained agent, learning from scratch, must acquire all these skills solely through the task-specific reward function. This process can make learning more difficult, as the agent relies entirely on extrinsic reinforcement to develop attention, coordination, precision, and motor control.

We fine-tuned the intrinsic agent from Section 5.4.1 to validate our hypothesis. Then, we compared its performance on the task with that of an extrinsic agent trained from scratch using only simple extrinsic rewards. Both agents were trained with first-person stereo vision and had unrestricted neck actuators, allowing them to choose where to look. The extrinsic agent was given more training steps than the intrinsic agent, compensating for the time spent on purely intrinsic training and ensuring that both had the same number of environment interactions. Additionally, we maintained similar model capacity for both agents, ensuring that the number of parameters remained comparable.

Also, we removed the StatePredictor in the intrinsic agent and the curiosity reward from training, retaining only the actor, critic, and StateNet. We made only the last layer of the StateNet, half of the critic's layers, and the sole layer of the actor available for policy adaptation. We aimed to remove the curiosity reward and the StatePredictor and freeze part of the weights to ensure a more robust evaluation of the skills learned during intrinsic training. We aimed to prevent the agent from developing a new form of curiosity-driven exploration associated with the task, which could make it more challenging to determine whether the skills acquired purely through intrinsic training were genuinely being transferred and refined or if the agent was merely developing new exploratory strategies by leveraging the curiosity reward in combination with the task reward.

Therefore, for the *capturing-ball task*, we designed a scenario where the robot is positioned in front of a confined table containing a red ball, as illustrated in Figure 5.29. The walls surrounding the table were added to reduce the frequency with which the ball exits the environment. At the beginning of each rollout, the ball spawns at a random position on the table and moves in a randomly selected direction. Its trajectory may change upon

colliding with the walls, the robot's hands, or, with a 10% probability, upon being assigned a new randomly sampled direction. The task's primary objective is to encourage the agent to capture and retain the moving ball in its hands. However, this is not a trivial task, as the ball may spawn in locations that are difficult to reach, requiring the agent to extend or retract its arms of different forms to retrieve it from the table's corners. The target distribution constantly shifts, requiring the robot to observe the table attentively to capture the ball.

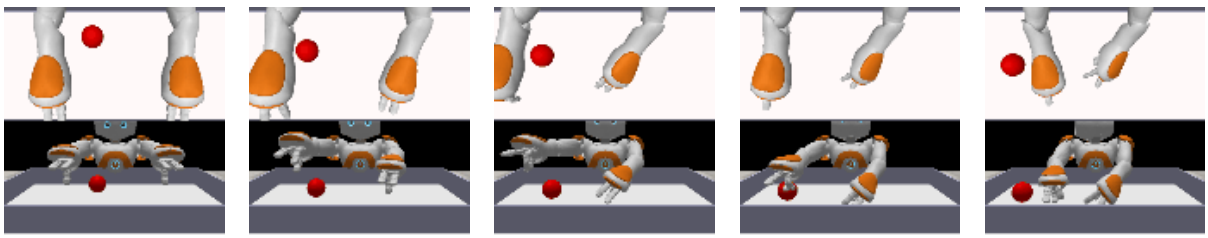
The successful execution of this task requires the agent to develop a range of motor and perceptual skills. It must learn to closely observe the ball and predict its trajectory based on its velocity and direction, as it is constantly in motion. Additionally, the agent must be capable of executing rapid and reflexive manipulative actions to capture the ball when it is within reach, as it does not move slowly, exhibiting a positional update rate of 0.05 centimeters per frame. After capture, another essential skill is grip stability, as the ball easily slips through the agent's fingers, making retention an additional challenge. Furthermore, since the ball can spawn in different locations and its movement is random, the policy must generalize effectively across various configurations, ensuring that the robot can capture it regardless of its point of origin.

The extrinsic reward function r_{ext} considers one point for each touch made by the phalanges on the ball, thereby encouraging the agent to keep the ball in its hands, as following

$$r_{\text{ext}} = \sum_{i=1}^{M} c_i, \tag{5.18}$$

where c_i is a binary value, with 1 indicating a collision and 0 indicating no collision, and M is the number of phalanges.

We trained both agents with the same parameters to ensure a fair comparison; in terms of model complexity, both have 330,631 trainable parameters. However, in the intrinsic agent, a large portion of the weights is frozen, leaving only 69,533 parameters for fine-tuning. The intrinsic agent was adapted to the task for 2 million steps, whereas the extrinsic agent was trained for 5.7 million steps. This number was chosen to compensate for the 3.7 million steps of prior intrinsic training, ensuring that both agents explore



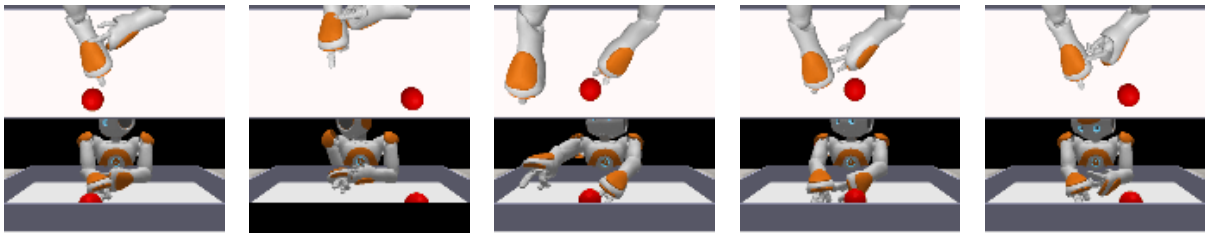
(a) Sample 0 (b) Sample 1 (c) Sample 2 (d) Sample 3 (e) Sample 4

Figure 5.29: Sequence of samples from the capturing-ball environment. In (a), we have the first frame of the sequence, in which the ball is born randomly on the table. In (b) to (e), we have frames demonstrating moments in which the ball moves across the table in random directions while the agent moves its arms.

the environment for the same number of steps. We believe that, with only the task-specific reward, the purely extrinsic agent will struggle to develop all the necessary skills to complete the task successfully. In contrast, the intrinsic agent is expected to adapt more efficiently by leveraging the skills acquired during intrinsic training.

At the beginning of the adaptation phase, the intrinsic agent starts training with behaviors inherited from the curiosity-driven policy and, within 300,000 steps in the environment, already exhibits a significant adjustment of its policy to improve the rewards obtained in the current task. The agent resumes from the point where it left off in intrinsic training, primarily exploring the right side of the environment, as it had already learned the dynamics and objects present in the scene, which no longer elicited as much interest, as shown in Figure 5.30 (a) and (b). However, upon starting extrinsic training, the agent eventually made contact with the ball using its fingertips and received a reward, which led to a policy adaptation. This feedback prompted the agent to adjust its body movements, redirecting its attention to the table and tracking the ball with its head, since without this adjustment, maximizing the obtained rewards would have been significantly more challenging (Figure 5.30 (c), (d), and (e)). This behavior is quickly adopted, as the agent has mastered his neck movement and understands its impact on visual perception. Upon receiving the reward, it promptly associates that it should reuse this movement to optimize its task performance.

From a million adaptation steps, the agent began tracking the ball's movement with its eyes, observing its dynamics on the table, as shown in Figure 5.31 (a). It developed sophisticated exploratory strategies to maximize rewards, such as orienting its entire body to follow the ball and delicately touching it with its fingertips. These behaviors indicate that the agent could transfer and adapt the exploratory strategies acquired during intrinsic training, efficiently coordinating its limbs to optimize reward acquisition. This process suggests that the initial policy was not discarded but refined to align with the new reward function. In Figure 5.31 (b), the agent refined its motor control, slowing down its hands and fingers to gently touch the moving ball. It adjusted movement intensity to maximize



(a) Sample 0 (b) Sample 1 (c) Sample 2 (d) Sample 3 (e) Sample 4

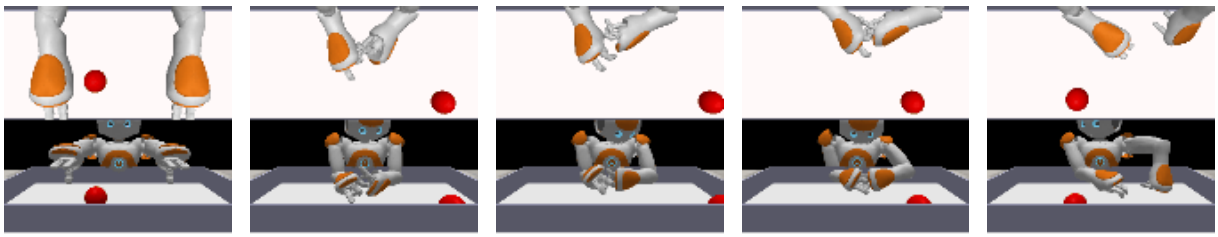
Figure 5.30: Sequence of samples from the early stages of the intrinsic agent's adaptation. In (a) and (b), two frames taken within the first 384 adaptation steps show that the agent starts from where its curious exploration ended, looking toward the right side of the environment. In (c), (d), and (e), frames illustrate the agent's behavior from 300,000 adaptation steps. At this stage, the agent associates that redirecting its attention to the table enhances its perception, allowing it to achieve higher rewards by reusing previously learned behaviors.

the frequency of successful interactions. This adaptation indicates that the agent learned to track the ball visually and improved motor control to optimize reward acquisition. Later, as shown in Figure 5.31 (c), the agent attempted to grasp the ball with its fingers, even when it appeared in more distant positions, using small taps to move it into more accessible regions. These exploratory strategies reflect the development of two distinct types of behaviors: *deliberative*, where the agent observes the ball and waits for the right moment to touch it, and *reactive*, where the agent quickly acts to capture the ball before it escapes.

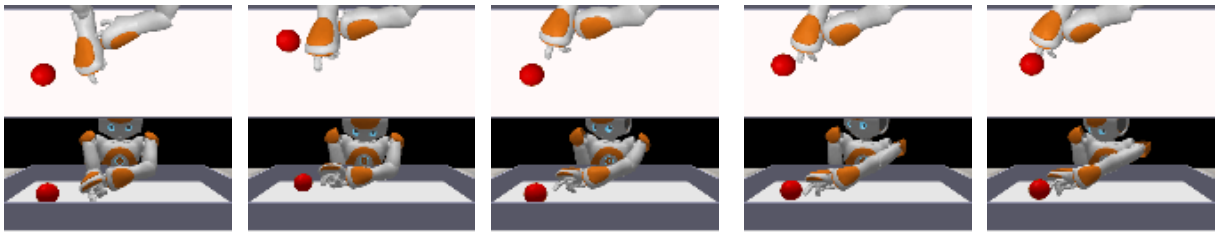
In contrast, the extrinsic agent, even after 1 million steps, still failed to develop this association, consistently keeping its head turned away from the table and its arms extended for most steps, as shown in Figure 5.31 (d). This behavior suggests that the agent could not establish a connection between visual perception and task execution, limiting its ability to explore the environment effectively. We acknowledge that the extrinsic agent starts from scratch and needs to learn how to control its body. However, when initialized under the same conditions during purely intrinsic training, the intrinsic agent had already developed motor control across all limbs and visual attention only into one hundred steps. The extrinsic agent's difficulty acquiring these skills solely through task-based rewards, even though they were not highly sparse, highlights the importance of intrinsic mechanisms in fostering more sophisticated exploratory behaviors and building internal representations that facilitate adaptation to new challenges.

The reward curve for task training of both agents is illustrated in Figure 5.32. It demonstrates that the intrinsic agent adapted satisfactorily to the task, achieving a maximum reward of 2.68 points at the end of training. In contrast, the purely extrinsic agent, trained from scratch, struggled to reach the same level of success, exhibiting a significantly lower reward curve, even after spending more time on the task to compensate for the intrinsic agent's exploration time. The extrinsic agent remained random until about 2 million steps, with disorganized actions in the environment. After this point, it began to learn a task policy, but with reward gains much lower than those of the intrinsic agent, which, despite spending less time on the task, had already developed key skills that facilitated its learning. In contrast, the extrinsic agent was unable to develop the necessary skills to maximize its reward gain using only the reward function we had defined.

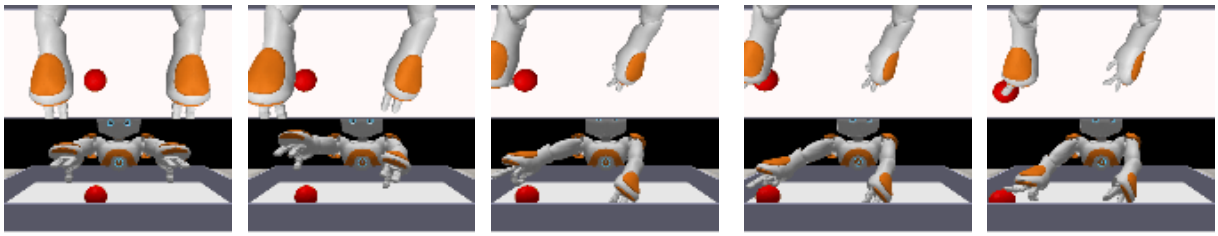
At the end of training, the reward curve for the intrinsic agent exhibited an upward trend, suggesting that continued learning could further enhance its performance. In contrast, the reward curve for the extrinsic agent remained constant, indicating no potential for further improvement, even with additional training steps. We evaluate the policies learned by both agents executing 100 test cases in which the ball appeared in random positions with random movement. We computed the mean and standard deviation of the extrinsic rewards obtained. The intrinsic agent was rewarded 1.35 ± 1.26, with 12% of test cases in which the agent failed to touch the ball with any phalanx. In comparison, the extrinsic agent obtained a reward of 0.84 ± 0.82, with 21% of test cases in which it failed to touch the ball. These results highlight the superior performance of the intrinsic agent in completing the task. However, both agents exhibited high standard deviations due to the inherent difficulty of the task. Their performance varied significantly across test cases, leading to substantial fluctuations in the standard deviation.



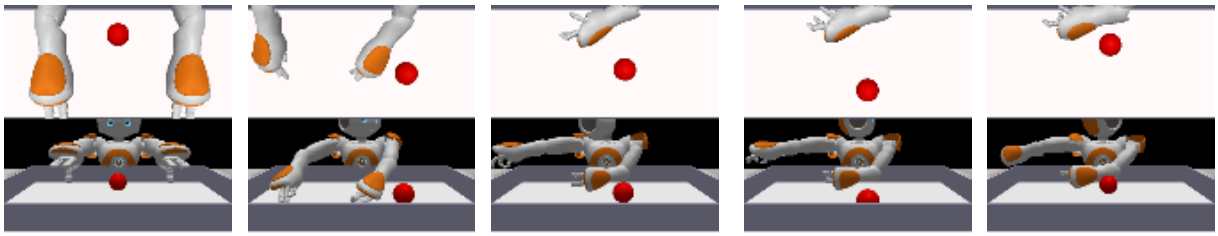
(a) Trajectory A



(b) Trajectory B



(c) Trajectory C



(d) Trajectory D

Figure 5.31: Trajectory samples of the agent's adaptation process from a million adaptation steps. In (a), the intrinsic agent begins tracking the ball's movement with eyes, observing its dynamics on the table. In (b), the intrinsic agent further refines its motor control, adjusting the speed and intensity of hand and finger movements to gently touch the moving ball, aiming to optimize the frequency of successful interactions. In (c), the intrinsic agent attempts to capture the ball even when it appears in distant positions, using rapid movements to direct it toward more accessible regions. These strategies reflect two distinct types of behavior: *deliberative*, in which the agent observes the ball and waits for the optimal moment to touch it, and *reactive*, in which the agent acts quickly to capture the ball before it escapes. In (d), we have samples of an extrinsic agent's rollout from a million adaptation steps, where it fails to fix its vision on the table and coordinates its arms to touch the ball.

Our results emphasize that curiosity-driven multimodal learning and the agent's immersion in the environment are crucial for developing fundamental motor skills to solve complex tasks. Moreover, our approach facilitates the construction of a representation

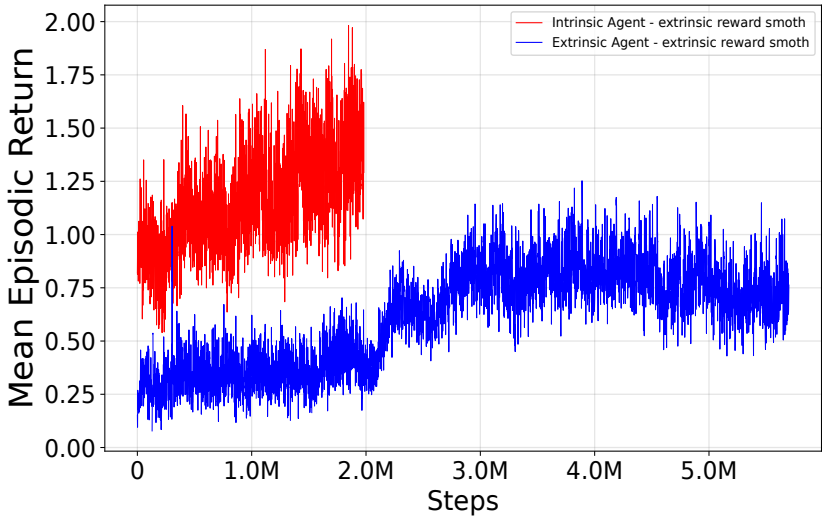
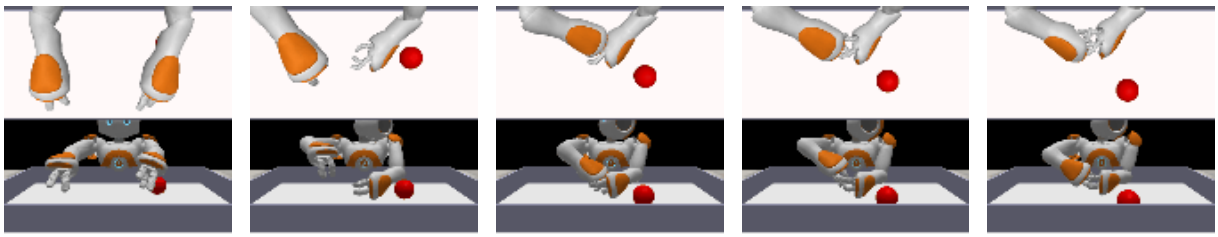


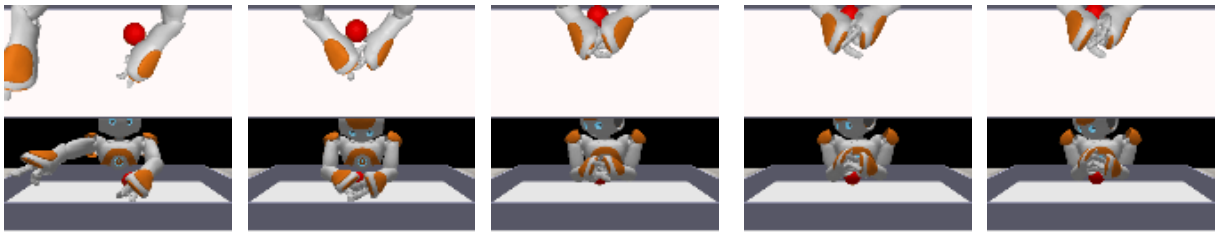
Figure 5.32: Mean episodic return of intrinsic and extrinsic agents. The blue curve represents the mean reward of extrinsic agent, while the red curve corresponds to the mean reward of intrinsic agent. Both agents were trained for 5.7 million steps. However, the intrinsic agent spent the first 3.7 million steps using the curiosity-driven reward to predict the next environmental observation, followed by 2 million steps of task-specific adaptation. The curve represents only the reward obtained during the task adaptation phase. In contrast, the extrinsic agent was trained exclusively with the task reward for the entire 5.7 million steps.

that preserves transferable motor skills, which are essential for embodied agents to acquire new tasks successfully. Figure 5.33 illustrates examples of the intrinsic agent's policy behavior compared to the policy learned by the extrinsic agent in the test environments. In Figure 5.33 (b) and (d), we observe that the intrinsic agent maintains an attentional focus on the table and objects, allowing it to track the ball's movement and adjust its arm positioning accordingly for effective capture. Conversely, in Figure 5.33 (a), the extrinsic agent struggles to maintain visual focus on the table and the ball's trajectory, frequently looking away and exhibiting significant difficulty in bending its arms and positioning them along the ball's path. We hypothesize that these challenges stem from its limited visual coordination; since it frequently looks backward, it fails to infer the ball's motion and adjust its arm movements accordingly. The simple reward function, which only reinforces finger contact with the ball, was insufficient for developing the coordination between vision and arm movements, suggesting that a more complex reward structure is required for the extrinsic agent.

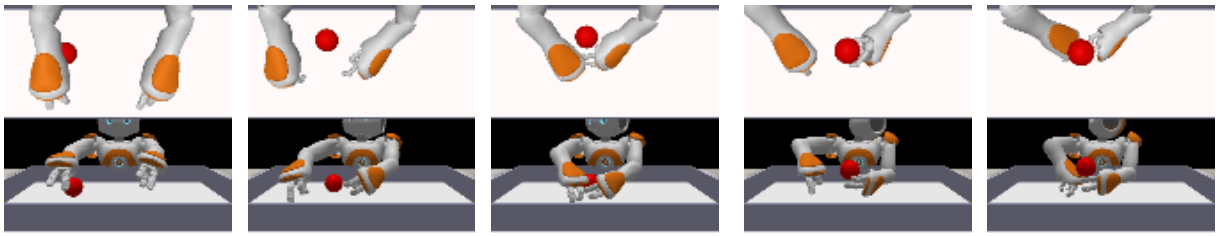
Even though the extrinsic agent achieved an average reward of 0.84, its method of touching the ball remains less human-like than the intrinsic agent, as shown in Figure 5.33 (c). In some test cases where the intrinsic agent does not achieve high rewards, it can still capture the ball and keep it pressed against the table wall to prevent it from escaping, even if it fails to maintain finger contact at every step, as observed in Figure 5.33 (b). Sometimes, the ball slips from its grasp, but the intrinsic agent continues to attempt recovery. In contrast, the extrinsic agent often fails to even attempt capturing the ball, primarily due to its tendency to look away from the scene. These qualitative re-



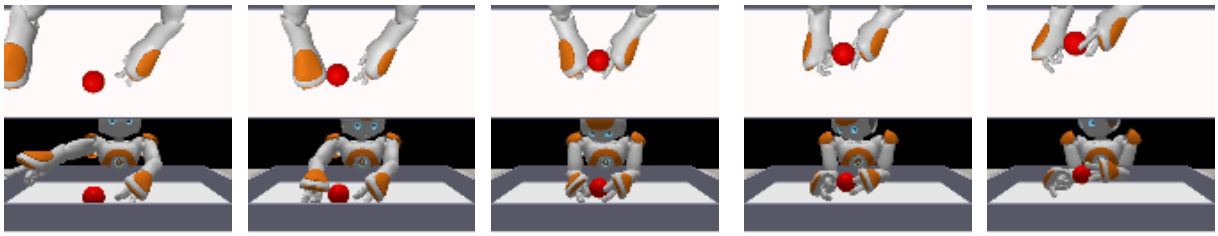
(a) Test 2 - Ext.



(b) Test 2 - Int.



(c) Test 8 - Ext.



(d) Test 8 - Int.

Figure 5.33: Trajectory samples of the policy learned by the extrinsic and intrinsic agents in different test cases. In figure, Int. indicate intrinsic agent, and Ext. indicate extrinsic agent. In (a) and (b), we observe the agents' behavior for Test 2, while in (c) and (d), we analyze their behaviors for Test 8. In both scenarios, the intrinsic agent demonstrates superior performance, even without consistently touching the ball using its fingers. Nevertheless, it successfully captures and maintains control of the ball, as highlighted in (b) and (d). In contrast, the extrinsic agent exhibits more difficulty coordinating its movements, frequently shifting its gaze to the sides, impairs its perception of the events occurring on the table. In each test, the frames captured from the agents are not temporally aligned, as our goal was to highlight moments when the agents performed the most relevant movements for the task, which do not always occur at the same time steps within each trajectory.

sults demonstrate that the extrinsic agent could not learn complex motor skills using only our task reward function, as the necessary elements for solving the task are challenging to acquire through simplistic reward designs.

5.5.1 Testing Cognitive Biases

Until now, we have achieved highly significant results in our *capturing-ball task*. This task has highlighted circumstances in which our approach outperforms purely extrinsically motivated agents and demonstrated how acquiring skills through intrinsic training can be crucial for enhancing the flexibility of complex agents. To evaluate whether the results of our task adaptation experiment can be enriched, such as increasing adaptation speed, improving reward gains, or enhancing policy generalization across different situations encountered in the capturing-ball task, we introduce, in this section, additional cognitive biases into the neural architecture that support the agent's internal environment.

We hypothesize that policy adaptation in monolithic architectures is far more limited than in sparse, modular, and bidirectional hierarchical architectures. There is evidence in other contexts and applications that these biases improve generalization, provide more flexible structures, and better capture the compositional structure of the world [94, 42, 88] providing more flexible and generalizable models. Using these architectures in our framework to build a modular and hierarchical curious policy can enable the localized and efficient adaptation of specific parts of the curious policy during task adaptation, resulting in a more fluid and effective agent adaptation.

To test our hypothesis, we replaced some monolithic and unidirectional hierarchal layers in the StateNet, Actor, and Critic to independent recurrent modules organized hierarchically with bidirectional flow, as illustrated in Figure 5.34. We preserved isolated encoders to process each sensory modality separately and delegated the fusion of these signals to the recurrent modules in the first hierarchical layer. In the Critic, the linear layer (256×128) responsible for fusing multiple sensory inputs was replaced with two layers of independent recurrent modules, each containing four modules of 32 neurons, with an attentional bottleneck restricting simultaneous activation to only two modules at a time. Similarly, the StateNet replaced the 256×128 linear layer with a layer comprising four modules of 32 neurons each, with only two actives at any given time. In the Actor, we removed two linear layers and introduced a modular layer with four modules of 32 neurons each, enforcing the same constraint of activating only two modules simultaneously.

We incorporated an intermediate self-attention mechanism between the StateNet and the Actor to facilitate the hierarchical transition of information from sensory fusion in the StateNet to the motor action region in the Actor, enabling an adaptive flow between perception and control, as illustrated in Figure 5.34. At the end of the Actor, we added a linear decoder with 64 neurons, responsible for transforming modular information into a 28-dimensional vector corresponding to the agent's actions. At each time step, only the data from the two active modules are passed to the decoder, ensuring that only the modules containing essential knowledge for maximizing the reward at that moment are utilized. During learning, the gradient flows to adjust the weights only of the active modules, leaving the weights of inactive modules unchanged.

We trained this agent intrinsically and did not observe significant changes in the agent's autonomous development compared to the previous version. The agent acquired the same previously identified skills, such as lifting objects for closer examination, developing attention around 100,000 steps, and improving finger movement precision and dexterity.

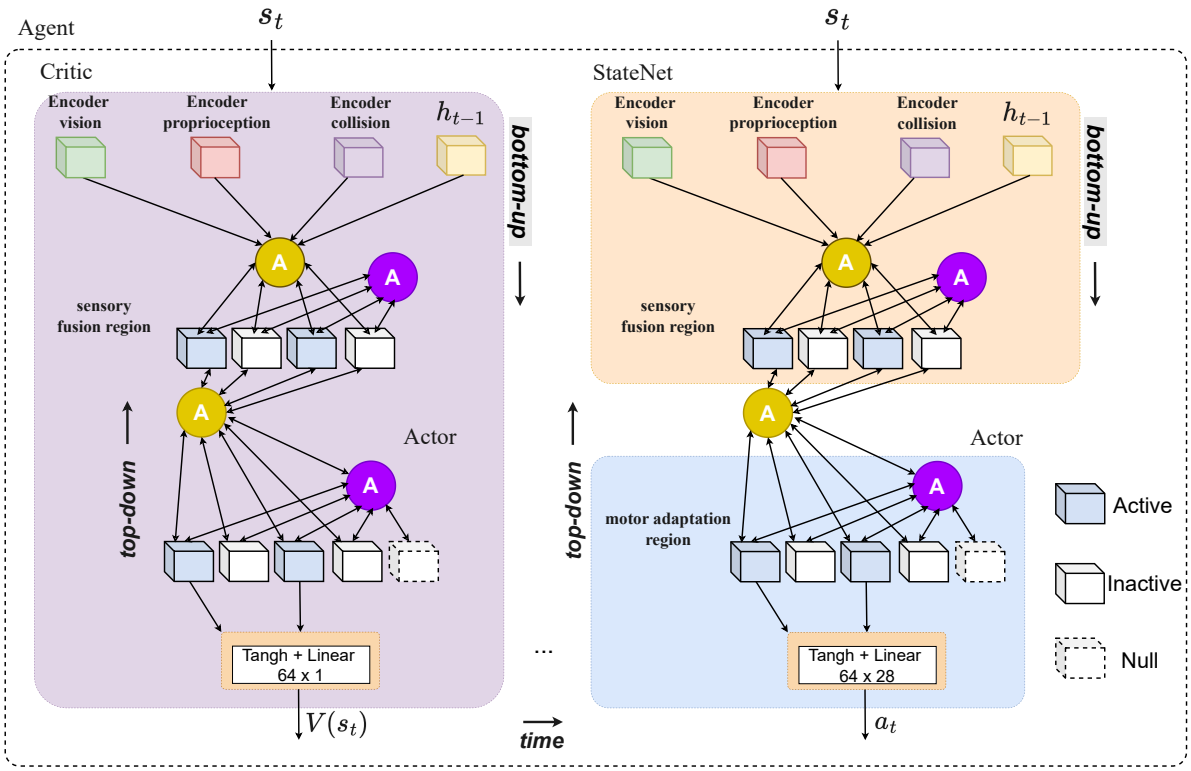
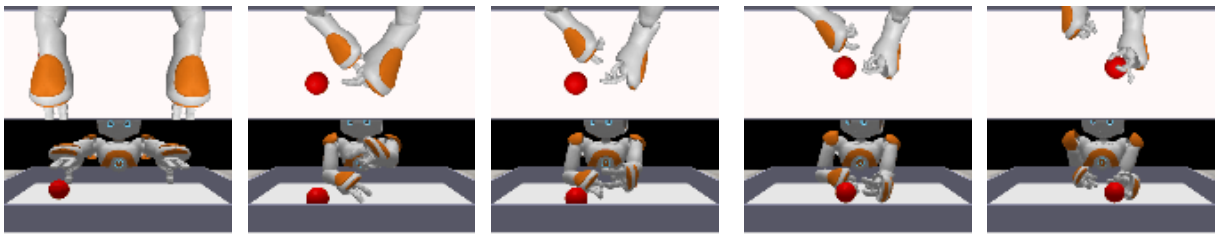
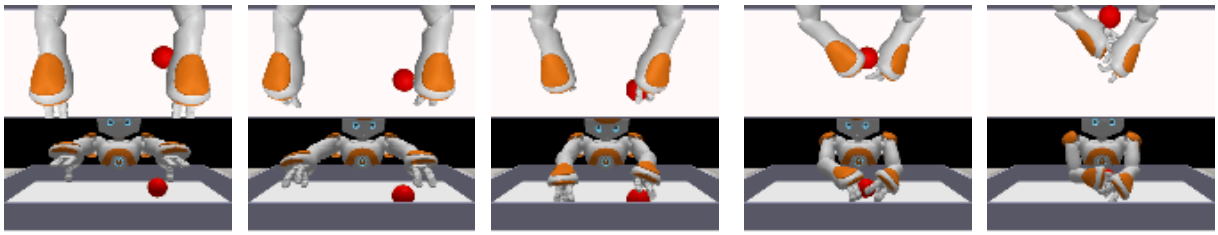


Figure 5.34: Our agent's Independent Recurrent Modules for task adaptation. This figure illustrates the hierarchical organization of independent recurrent modules in the extrinsic agent, used to replace monolithic layers in the StateNet, Actor, and Critic. Each sensory modality is processed separately by isolated encoders, and sensory fusion occurs in the first hierarchical layer of recurrent modules. The Critic and StateNet incorporate four recurrent modules of 32 neurons each, with an attentional bottleneck allowing only two to be active at a time. Similarly, the Actor includes a modular layer with the same structure, ensuring selective activation of relevant modules. An intermediate self-attention mechanism facilitates hierarchical information flow between the StateNet and the Actor, enabling adaptive transitions between perception and action. At the final stage of the Actor, a linear decoder transforms modular information into a 28-dimensional action vector, ensuring that only the most relevant modules influence decision-making. During learning, the gradient only updates the active modules while the remaining inactive ones remain unchanged. Attention mechanisms facilitating communication between a layer's input and its modules are marked with yellow "A", while attention mechanisms enabling interaction between modules within the same layer are indicated by purple "A".

The agent's body coordination during movements also remained highly similar, as did the losses and prediction quality across the three sensory modalities. This result was expected, as we did not modify the StatePredictor, which was directly responsible for the agent's future predictions. To ensure a fair comparison between agents, we trained the current agent purely intrinsically up to 3.7 million steps, initially using a fixed scenario with three cubes for 2.5 million steps. From that point onward, we introduced more diverse scenarios and the agent's imagined feedback until the end of training. We adjusted the model's capacity to approximately 4.8 million parameters to maintain equivalence with the prior



(a) Trajectory A



(b) Trajectory B

Figure 5.35: Samples of different strategies used by the agent with various cognitive biases to capture the ball during task adaptation. In (a), the agent observes the direction the ball is moving and lowers its right arm to a specific point on the table where it anticipates the ball will pass. It then waits with its hand and fingers open, positioned to catch the ball. In (b), the agent employs a different strategy by reaching directly for the ball since it is closer to its arm. It then pushes the ball toward the nearest wall, trapping it between its arms while keeping its fingers lightly touching the ball.

agent. To preserve the hierarchy among the modules and the attentional communication, we removed two convolutional layers and one GDN layer from the StatePredictor, ensuring that both agents had the same capacity. However, we did not observe any impact from this removal on the quality of the agent's visual predictions.

During the agent's adaptation task, we followed the same adaptation protocol, removing the intrinsic reward, deactivating the StatePredictor, and freezing all weights of the StateNet, along with half of the weights of the Critic. Only the modular layer of the Actor, the decoder, and the self-attention mechanism responsible for bidirectional information transmission between the StateNet and the Actor remained trainable for fine-tuning. We deliberately kept the self-attention weights adjustable, as we believe this mechanism plays a crucial role in retrieving task-relevant knowledge, thereby facilitating the agent's adaptation. As a result, the model contained 330,631 trainable parameters, a number minimized by restricting the recurrent modules to only 32 neurons each. This compression reduced the latent space vector, used for predicting both the state value in the Critic and the joint actions in the Actor, to 64 neurons, half the previous size when using linear layers. However, the modular architecture requires self-attention mechanisms for efficient communication between modules within the same layer and across hierarchical layers. Despite optimization efforts to minimize parameter count, these mechanisms inevitably add extra parameters. Since our primary interest lies in evaluating the interaction between these biases, we opted to maintain the model with this parameter count.

During the agent's adaptation to the task, we once again observed the refinement of

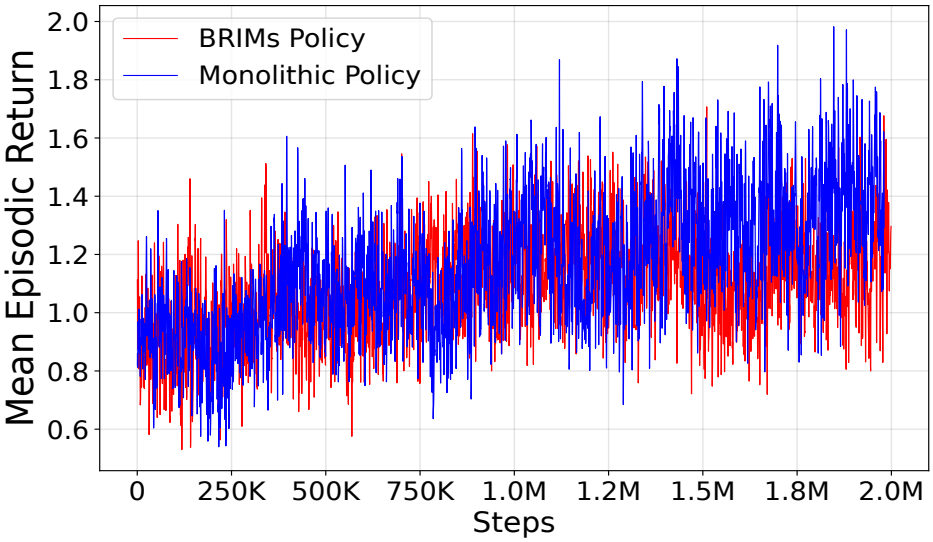
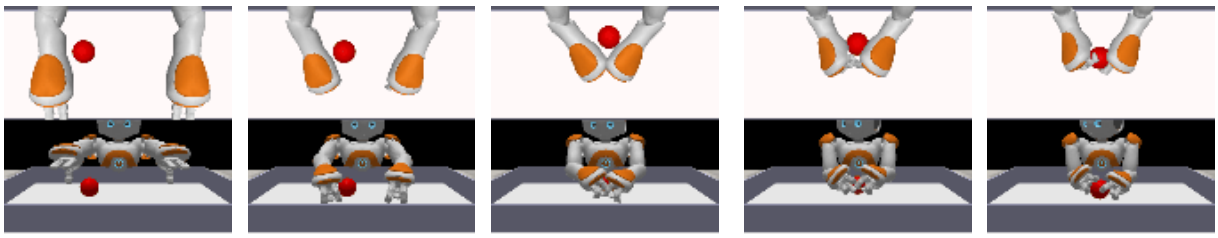


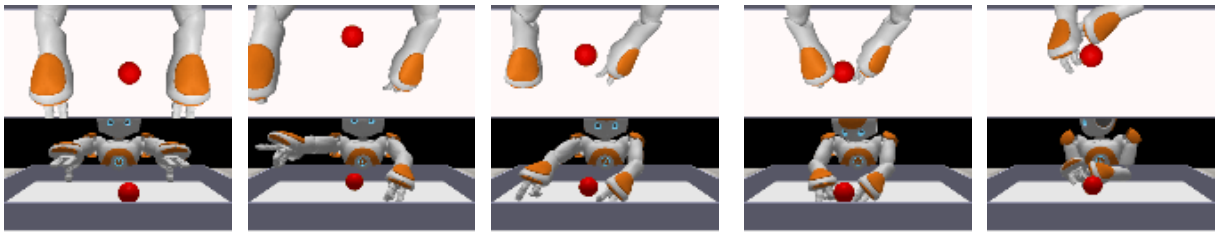
Figure 5.36: Mean episodic return of the intrinsic agent with a modular, sparse, and bidirectional hierarchical policy and the intrinsic agent with a monolithic policy. The curves indicate that both agents exhibited similar performance regarding reward gain and adaptation speed.

previously learned skills and the exploration of novel and effective strategies for capturing the ball, as illustrated in Figure 5.35. In (a), the agent analyzes the ball's trajectory and anticipates the point on the table where it is most likely to pass. Then, it positions its right arm by lowering it to the table surface while keeping its hand open, with fingers prepared to securely grasp the ball as soon as it reaches the expected location. The agent then waits for several steps, adjusting the posture of its fingers to maximize the chances of a successful capture. In (b), the agent adopts a more active behavior. Instead of waiting for the ball to reach a predefined position, it swiftly moves its arm toward the object. Upon reaching the ball, the agent applies light pressure and directs it toward the nearest wall, leveraging the wall as an auxiliary element to restrict movement. To ensure the ball remains under control, the agent positions both arms around it, maintaining slight contact with its fingers to secure the ball and receive the extrinsic reward. These distinct yet highly advanced strategies for a robot of this complexity demonstrate the agent's efficient adaptation, enabling it to exploit environmental opportunities to explore diverse capture techniques.

Our results showed that the intrinsic agent's mean episodic return with more cognitive biases, compared to the intrinsic agent with a monolithic architecture, did not significantly improve adaptation speed or reward gain. The two curves display similar characteristics, as shown in Figure 5.36. However, quantitative tests indicate that the current agent's policy achieved an average extrinsic reward of 1.16 ± 0.97, while the agent with a monolithic policy obtained 1.36 ± 1.26. By adding more cognitive biases, we achieved a 23% improvement in the task's standard deviation. Although the monolithic agent's mean episodic return was higher, the standard deviation was lower in this agent. This reduction in standard deviation suggests that this agent's performance was more consistent across episodes. In contrast, the monolithic agent exhibited more significant variability,



(a) BRIMs



(b) Monolithic

Figure 5.37: Samples of the agents' policy behavior in Test 13. In (a), we have the agent with a BRIMs policy that performs well in the test case, successfully capturing the ball. In (b), we have the behavior of the monolithic policy. In this test, this agent fails to keep the cube in its pose because when the ball slips out of its hands, instead of capturing it, it starts to look back.

indicating that the additional biases promote a more stable adaptation process, reducing extreme learning oscillations and making the policy less susceptible to fluctuations between successful and unsuccessful attempts.

We believe that the 23% improvement in the standard deviation, resulting in a more stable policy for the tested scenarios, is related to the fact that the introduced biases assist the agent in combining different strategies more efficiently, allowing for a more balanced adaptation to the various testing conditions. Upon analyzing the accumulated rewards for each test case and the behaviors exhibited by the agent, we confirm that the agent with the monolithic policy is more sensitive to the different test cases. In some tests, the accumulated rewards achieved are higher than those of the agent using the BRIMs policy; however, in other cases, the rewards are significantly lower, indicating that the agent with the BRIMs policy is less sensitive to the variations observed among each test case, as illustrated in Figure 5.37. Based on these results, the BRIMs policy maintained greater consistency across the different cases, demonstrating superior adaptability potential.

Chapter 6

Conclusion and Future Works

In this chapter, we discuss the key results achieved in our work (Section 6.1). Also, we analyze the hypotheses (Section 6.2) raised in the first chapter. Finally, in Section 6.3, we conclude by outlining the limitations of our work and key directions for future developments in this field.

6.1 Key Results

This work proposes DreamerRL, a framework to enable more adaptable and autonomous humanoid robots through predictive world model construction. Grounded in world model theories, DreamerRL focuses on constructing an internal predictive world model of the environment instead of optimizing policies for specific tasks. Through this predictive learning, the robot gains adaptable policies and acquires different and general motor skills by actively exploring unfamiliar states. Our approach underscores the critical role of embodiment, biologically inspired neural architectures based on neocortical circuity, and intrinsic motivation in forming and refining internal world models. Additionally, this work advances the development of more autonomous and cognitively flexible robotic agents capable of learning from their own experience without reliance on imitation or manually engineered supervision.

We evaluate our proposed framework with experiments in a robotic manipulation environment. In the first experiment, the agent began learning to model the environment using intrinsic motivation driven solely by third-person visual perception. The results demonstrated a limited capacity for exploration and skill emergence. However, in the following experiments, when we introduced multimodal curiosity integrating proprioception, tactile, and first-person vision, the agent exhibited a remarkable leap in behavioral complexity and autonomously developed more complex skills. Notably, when we enabled neck mobility, allowing the agent to direct its gaze freely, we observed the **spontaneous emergence of sustained visual attention** aligned with upper-limb actions. The sustained attention and alignment between vision and upper-limb motor behavior **represent a significant milestone in autonomous robotic learning**. Remarkably, these behaviors emerged without external supervision and mirror key developmental observed in human infants, suggesting that our approach can computationally replicate foundational

aspects of child development within a complex robotic system. These findings validate our framework's embodied, intrinsically motivated approach and open new research avenues for building truly autonomous, self-supervised complex agents capable of developing sophisticated behaviors using only interaction with the world without human supervision.

Finally, we validated the effectiveness of internal representations in a downstream task, *CaptureBall*, which demanded real-time coordination, attention, and goal-directed behavior. **DreamerRL adapted to this novel extrinsic task with minimal retraining** and **outperformed a baseline agent trained purely with extrinsic rewards**. These results prove that our approach is essential for adaptable representations. Our findings underscore the transformative potential of grounding representation learning in the agent's own sensorimotor experience, ultimately enabling the development of robotic agents that are more flexible, adaptive, and capable of complex generalization. Furthermore, the introduction of modularity and hierarchical processing—architectural principles inspired by the neocortical circuit—significantly enhanced the stability and generalization of the learned policies. **Our model exhibited greater policy robustness and improved transfer capabilities** compared to agents trained without such structures.

6.2 Hypotheses Evaluation

The hypotheses raised by this work were essential in guiding the development of the experiments. Therefore, their respective answers following the experiments are presented:

1. **H1**: A complex robotic agent, trained to model the world in an object manipulation environment, can accurately predict both the dynamics of the external environment and its behavior.

   Answer: This hypothesis was **confirmed** through experiments conducted with the baseline agent, as described in Section 5.2. The results indicated that the agent successfully predicted its own body's dynamics, accurately anticipating its general movements. However, some limitations were observed in predicting finger movements, which exhibited lower accuracy than other body parts. Furthermore, the agent demonstrated excellent predictions of the visual characteristics of objects in the environment, precisely identifying both the shape and color of static and dynamic objects. More importantly, it could predict the effects of its actions on these objects, correctly anticipating the outcomes of its interactions. This includes the direct impact of its actions on the manipulated objects and the interactions between different objects within the environment.

2. **H2**: A complex robotic agent autonomously develops structured object manipulation behaviors driven solely by the motivation to predict the world.

   Answer: This hypothesis was **partially confirmed** through the experiments described in Sections 5.2 and 5.3. The results demonstrated that the agent developed structured manipulation behaviors exclusively through intrinsic motivation generated by curiosity, including actions such as touching, holding, lifting and holding,

throwing and dragging, with intense visual attention and highly coordinated behaviors. We consider the hypothesis only partially confirmed because the agent could not develop even more complex behaviors, such as putting and stacking objects, among other more structured behaviors. However, it was able to develop valuable skills for any task, such as head control, sensory alignment between different modalities, and dexterity and precision in the movements performed.

3. **H3**: Increasing embodiment enables the agent's complete immersion in the environment, promoting the autonomous development of more complex object manipulation skills.

   Answer: This hypothesis was **confirmed** by the experiments described in Section 5.3, which demonstrated a direct relationship between the progressive use of sensors and actuators and the agent's engagement with the environment. As its sensorimotor capabilities expanded, the agent exhibited more active interactions with the environment and acquired more refined object manipulation skills. The results also indicated that a more complete immersion of the agent is essential for effective object manipulation. The agent's exploratory curiosity was significantly limited when operating with restrictions, such as a fixed neck and third-person vision, leading to more superficial and reflexive interactions with objects without robust autonomous development. These findings highlight the importance of the synergy between intrinsic curiosity, sensory perception, and motor control. They demonstrate that a higher degree of embodiment enhances a more autonomous, adaptive, and structured learning process in manipulation tasks.

4. **H4**: The world model learned through sensorimotor experiences enables the robotic agent to learn abstract concepts about how the world functions, allowing it to imagine and simulate novel situations not encountered during training.

   Answer: This hypothesis was **partially confirmed** by the experiments described in Section 5.4, which demonstrated that the agent generalizes very well in the *real feedback* configuration, where the agent is required to predict only a single frame, its performance was very significant, showing a strong ability to anticipate future states. However, its performance deteriorates when the agent must recursively predict multiple frames, using its own imagined data as input for subsequent predictions. This decline is primarily associated with residual errors, particularly in the proprioception modality, accumulating over iterations. Additionally, hallucinations were observed in arm movements and object interactions, especially in collision scenarios, indicating that the model still has limitations in this setting. These findings suggest that while the agent demonstrates a strong short-term imagination capability, there are necessary improvements when extending the simulation over multiple steps.

5. **H5**: The behaviors learned during world exploration are task-independent, making the agent more adaptive and capable of quickly applying the acquired exploration skills to accelerate the adaptation to a new extrinsic task.

   Answer: This hypothesis was **confirmed** through the capturing ball experiment described in Section 5.5, where our agent, with frozen weights, was exposed to the

task adaptation of capturing a randomly moving ball on the table. Our agent was able to reuse the skills previously acquired during the curious exploration phase and adapt them into more sophisticated behaviors, enabling it to achieve its goals in just a few steps. In contrast, a purely extrinsic agent, trained from scratch to learn the same task, failed to optimize its reward gain as effectively as the intrinsic agent (Figure 5.32). Even by the end of the training, the extrinsic agent failed to develop essential skills, such as motor coordination, dexterity, hand stability, visual attention on the table, and the ball's movement. These skills are fundamental for the agent to successfully solve the task and maximize its reward gain more efficiently. This result underscores the importance of our approach in developing foundational skills, which proved crucial when the agent is exposed to manipulation tasks requiring complex perceptual-motor synergy.

6. **H6**: Incorporating sparsity, modularity, and hierarchical biases enhances intrinsic policy adaptation to a new extrinsic task.

   Answer: This hypothesis was **confirmed** by the experiment described in Section 5.5, which demonstrated that incorporating sparsity, modularity, and bidirectional hierarchy biases into the intrinsic policy improves the agent's adaptability in scenarios where the target distribution is constantly changing. In the adaptation experiment for the ball-catching task, we observed that when the ball's speed remained constant across all rollouts, the monolithic and BRIMs policy agents exhibited similar performance regarding rewards obtained and adaptation speed. However, the 23% improvement in the standard deviation of the tests indicates that the introduced biases enhanced the generalization of the policy to the various test cases presented. In contrast, the agent with the monolithic policy exhibited more extreme behaviors across different cases, resulting in very low rewards in some tests and significantly high rewards in others, which reflects a more inflexible policy in handling the variability of situations encountered. Meanwhile, the BRIMs policy was able to maintain greater consistency across the different cases, demonstrating superior adaptability potential.

6.3 Limitations and Future Works

Our work presents significant advances in autonomous robotics development, leaving room for further improvements. Although we demonstrated the importance of sensorimotor integration and intrinsic motivation in constructing world models, the agent still operates within simulated environments that impose physical and perceptual constraints distinct from those encountered in the real world. The transition of our framework to real-world robotic scenarios requires further investigation, particularly regarding the robustness of internal representations in the presence of sensory noise, latency, and uncontrolled disturbances. Another limitation concerns the scope of sensory modalities employed. While first-person vision and proprioception have proven effective, other modalities, such as more detailed tactile sensing, audition, or interoception, remain unexplored, reducing the diversity of sensory experiences available to the agent.

Additionally, we faced challenges related to the complexity of the training process. Integrating multiple sensory modalities and novelty-driven intrinsic motivation can lead the agent to suboptimal or unstable behaviors during the early stages of learning, particularly in environments with high unpredictability in sensory inputs, which may result in slower learning, greater performance variance, and a tendency for the agent to become trapped in local minima. Furthermore, the intrinsic motivation system employed, while effective in promoting exploration and the emergence of skills, relies on relatively simple metrics of novelty or prediction error. We believe that more sophisticated motivational systems could further enhance autonomous development and construct more robust world models. Furthermore, there are opportunities to expand the methods employed and explore their applicability in diverse scenarios.

Adaptation Tasks. In this work, we tested the agent's adaptation to two challenging tasks, demonstrating its ability to learn in novel scenarios. However, there remains a vast field for exploration, including adaptation to different domains, new task categories, and more complex environmental variations. Additionally, investigating which agent feature facilitates or hinders this adaptation compared to extrinsically trained agents could foster new and relevant discussions. Further studies could focus on adaptation tests using independent recurrent modules and evaluate the agent in environments with unpredictable dynamics, such as abrupt changes in the scene or object distribution, to assess whether its internal representation provides the flexibility required for efficient adaptation.

Multiple-step Prediction. Our agent was trained solely to predict a single step. While this approach was sufficient to validate our hypotheses, it would be valuable to investigate whether predicting multiple future steps could lead to more complex behaviors. If the agent can anticipate future events, it may avoid redundant actions and develop long-term strategies, resulting in more structured and efficient behavior.

Embodiment is Essential. Our experiments demonstrated that increasing the agent's immersion in the environment through additional sensors and actuators resulted in more autonomous and robust development. This finding reinforces the importance of embodiment and sensory perception in motor and cognitive learning, as the agent interacts with the environment more richly and develops more refined motor skills by exploring its action space. A promising direction for future research would be to examine the influence of embodiment in more challenging scenarios, such as a complex domestic environment with doors, furniture, unstructured object distributions, and dynamic obstacles. The agent could be allowed to explore freely, relying solely on our multimodal curiosity reward based on next-observation prediction, enabling an assessment of whether a higher level of environmental immersion is sufficient for the agent to autonomously learn to balance itself, navigate the environment to investigate objects, and even interact with elements such as opening a door to access a new room. Furthermore, evaluating the agent's memory capacity in this setting would be crucial to understanding whether it can retain information about object locations, recall previously traversed paths, and optimize its exploratory strategies over time [48].

Novel Intrinsic Rewards. In this work, the agent was trained using curiosity as its sole intrinsic reward. However, other types of intrinsic signals, such as pain, hunger, fatigue, emotions (e.g., anger, sadness, joy), and even energy balance, play fundamental

roles in regulating behavior in biological systems. It would be valuable to test these reward signals individually and with curiosity to assess their impact on the agent's exploration, motor development, and adaptability. A particularly relevant experiment would involve modeling artificial physiological states, simulating an agent that must manage its energy levels to avoid exhaustion, and ensuring that a need for self-preservation balances its drive for knowledge. Furthermore, investigating how different reward combinations influence the construction of the agent's internal representation and whether the resulting emergent behaviors resemble evolutionary strategies found in biological organisms would provide valuable insights into intrinsic motivation in artificial agents.

Learning from Imagination. Given that our agent demonstrated the ability to predict the next multimodal observation and exhibited good generalization performance, a relevant investigation would be to explore the extent to which it can learn extrinsic tasks using only its internal world model as a simulation environment. This approach would enable training the agent without direct interaction with the real environment, a critical factor for applications where access to the physical environment is limited or costly. For such training to be viable, the world model must be sufficiently generalizable to handle previously unseen situations. One of the key challenges would be enhancing the fidelity of the internal model, ensuring that it accurately represents environmental dynamics without excessive bias toward previously encountered states. If the agent can successfully conduct training internally, this approach could drastically reduce the need for constant physical interactions with the environment, making reinforcement learning more efficient and practical for real-world applications where data collection can be expensive or time-consuming [47].

Curiosity and Attention. In the experiments presented in Section 2.2, we observed a synergistic relationship between multimodal curiosity-driven reward and the emergence of visual attention toward objects in the environment. A deeper investigation into this interaction would be highly valuable, particularly in contexts involving richer visual scenes. One possible direction would be to compare the behavior of the curious agent with human visual tracking data, assessing the extent to which the agent's emergent attention resembles human patterns of fixation and visual exploration. Furthermore, it would be interesting to test whether manipulating the curiosity reward influences how the agent allocates its attention over time, potentially leading to different exploration patterns.

Language. We believe incorporating language into the agent's world model could enhance generalization and structure internal representation. Expressing abstract concepts through language may help the agent better organize their predictions and structure their exploration. It would be interesting to test different approaches to integrating language, such as associating textual descriptions with environmental states, allowing the agent to develop an internal vocabulary to represent its experiences. Moreover, language could serve as a mechanism for planning and reasoning, enabling the agent to make more informed decisions by anticipating the consequences of its actions based on verbal descriptions. Investigating this relationship could contribute to developing more interpretable agents and advances in embodiment, integrating vision, proprioception, and language to construct a more structured understanding of the world [97] [17].

Reasoning and Planning. As the agent develops a world model, its predictions

could later facilitate tasks requiring reasoning and planning. A potential direction for future research would be to test whether the agent can use its predictions to construct more complex action sequences, anticipating consequences and adjusting its strategies based on future expectations. If the agent has predictions for multiple steps ahead, it could use them to develop planning strategies, avoiding actions that do not lead to the maximization of its reward. This mechanism may be crucial in enabling agents to perform highly complex tasks [125].

Social Domains. Social interaction influences all the elements used in this work, which presents an opportunity to expand the investigation into social domains. An interesting direction would be integrating multiple agents within the same scene, conducting experiments where two agents interact, such as playing with cubes while attempting to model the world. This setup could give rise to new and unexpected behaviors distinct from those observed in single-agent scenarios. In particular, the social interaction between the two agents could reveal collaborative or competitive behavior patterns, providing fertile ground for analyzing social phenomena. Such an approach would allow for an in-depth exploration of emerging social dynamics, with valuable implications for the developmental robotics field, as it investigates how agent interactions influence the development of cognitive and social skills while also opening new perspectives on artificial agent learning in social contexts [68].

Other Domains. Our approach is not confined to complex robotic agents but rather represents a self-supervised learning framework through reinforcement learning. It would be valuable to explore whether this approach, when applied to other tasks such as image classification, gesture recognition, and sentiment analysis, among others, yields interesting outcomes. Furthermore, comparing the results of an encoder pre-trained with this approach and traditional self-supervised learning would be insightful. Such a comparison would allow us to identify the strengths and weaknesses of our approach in terms of adaptation to new tasks, providing a deeper understanding of its potential across different domains [8].

Bibliography

[1] Joshua Achiam and Shankar Sastry. Surprise-based intrinsic motivation for deep reinforcement learning. *arXiv preprint arXiv:1703.01732*, 2017.

[2] James S Albus and Anthony J Barbera. Rcs: A cognitive architecture for intelligent multi-agent systems. *Annual Reviews in Control*, 29(1):87–99, 2005.

[3] Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976*, 2019.

[4] Parsa Bagherzadeh and Sabine Bergler. Multi-input recurrent independent mechanisms for leveraging knowledge sources: case studies on sentiment analysis and health text mining. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 108–118, 2021.

[5] Shi Bai, Fanfei Chen, and Brendan Englot. Toward autonomous mapping and exploration for mobile robots through deep supervised learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2379–2384. IEEE, 2017.

[6] Dana H Ballard. *Brain computation as hierarchical abstraction*. MIT press, 2015.

[7] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. Density modeling of images using a generalized normalization transformation. In *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[8] Adrien Bardes, Jean Ponce, and Yann LeCun. Mc-jepa: A joint-embedding predictive architecture for self-supervised learning of motion and content features. *arXiv preprint arXiv:2307.12698*, 2023.

[9] Lawrence W Barsalou. Grounded cognition. *Annu. Rev. Psychol.*, 59(1):617–645, 2008.

[10] LW Barsalou. Perceptual symbol systems. *The Behavioral and brain sciences/Cambridge University Press*, 1999.

[11] Andrew G Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically motivated learning in natural and artificial systems*, pages 17–47. Springer, 2013.

[12] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279, 2013.

[13] Daniel E Berlyne. Curiosity and exploration: Animals spend much of their time seeking stimuli whose significance raises problems for psychology. *Science*, 153(3731):25–33, 1966.

[14] Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

[15] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*.

[16] Bryan Chen, Alexander Sax, Francis Lewis, Iro Armeni, Silvio Savarese, Amir Zamir, Jitendra Malik, and Lerrel Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. In *Conference on Robot Learning*, pages 2328–2346. PMLR, 2021.

[17] Cédric Colas. *Towards Vygotskian Autotelic Agents: Learning Skills with Goals, Language and Intrinsically Motivated Deep Reinforcement Learning*. PhD thesis, Université de Bordeaux, 2021.

[18] Colin Cook and Yoram Jerry R Wind. *The power of impossible thinking: Transform the business of your life and the life of your business*. Pearson Prentice Hall, 2006.

[19] Alana de Santana Correia and Esther Luna Colombini. Neural attention models in deep learning: Survey and taxonomy. *arXiv preprint arXiv:4070703*, 2021.

[20] Kenneth James Williams Craik. *The nature of explanation*, volume 445. CUP Archive, 1967.

[21] Mihaly Czikszentmihalyi. *Flow: The psychology of optimal experience*. New York: Harper & Row, 1990.

[22] Kahneman Daniel. *Thinking, fast and slow*. 2017.

[23] Peter Dayan and CJCH Watkins. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[24] Richard De Charms. *Personal causation: The internal affective determinants of behavior*. Routledge, 2013.

[25] Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. See, hear, explore: Curiosity via audio-visual association. *Advances in neural information processing systems*, 33:14961–14972, 2020.

[26] Edward L Deci and Richard M Ryan. *Intrinsic motivation and self-determination in human behavior*. Springer Science & Business Media, 2013.

[27] Javier DeFelipe, Lidia Alonso-Nanclares, and Jon I Arellano. Microstructure of the neocortex: comparative aspects. *Journal of neurocytology*, 31(3):299–316, 2002.

[28] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173, 2021.

[29] Shirin Dora, Sander M Bohte, and Cyriel MA Pennartz. Deep gated hebbian predictive coding accounts for emergence of complex neural response properties along the visual cortical hierarchy. *Frontiers in Computational Neuroscience*, 15:666131, 2021.

[30] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[31] Heidi M Feldman and Michael I Reiff. Attention deficit–hyperactivity disorder in children and adolescents. *New England Journal of Medicine*, 370(9):838–846, 2014.

[32] Daniel J Felleman and David C Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

[33] Leon Festinger. A theory of cognitive dissonance. evanston: Row, peterson & company. *Go to original source*, 1957.

[34] Lucia Foglia and Robert A Wilson. Embodied cognition. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(3):319–325, 2013.

[35] Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning. *Journal of Machine Learning Research*, 23(152):1–41, 2022.

[36] Jay Wright Forrester et al. Principles of systems. 1968.

[37] Karl Friston and Stefan Kiebel. Predictive coding under the free-energy principle. *Philosophical transactions of the Royal Society B: Biological sciences*, 364(1521):1211–1221, 2009.

[38] Zhiqi Ge, Hongzhe Huang, Mingze Zhou, Juncheng Li, Guoming Wang, Siliang Tang, and Yueting Zhuang. Worldgpt: Empowering llm as multimodal world model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7346–7355, 2024.

[39] Charles D Gilbert and Torsten N Wiesel. Clustered intrinsic connections in cat visual cortex. *Journal of Neuroscience*, 3(5):1116–1133, 1983.

[40] Colleen J Gillon, Jason E Pina, Jérôme A Lecoq, Ruweida Ahmed, Yazan Billeh, Shiella Caldejon, Peter Groblewski, Tim M Henley, Eric Lee, Jennifer Luviano, et al. Learning from unexpected events in the neocortical microcircuit. *bioRxiv*, 2021.

[41] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[42] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[43] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[44] Daniel J Graham and David J Field. Sparse coding in the neocortex. *Evolution of nervous systems*, 3:181–187, 2006.

[45] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022.

[46] Richard Langton Gregory. The intelligent eye. 1970.

[47] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[48] Nick Haber, Damian Mrowca, Stephanie Wang, Li F Fei-Fei, and Daniel L Yamins. Learning to play with intrinsically-motivated, self-aware agents. *Advances in neural information processing systems*, 31, 2018.

[49] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*.

[50] Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in neural information processing systems*, 34:3680–3693, 2021.

[51] Nicklas Hansen and Xiaolong Wang. Generalization in reinforcement learning by soft data augmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13611–13617. IEEE, 2021.

[52] Jianye Hao, Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[53] Jeff Hawkins. A thousand brains: A new theory of intelligence, 2021.

[54] Jeff Hawkins and Subutai Ahmad. Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex. *Frontiers in Neural Circuits*, 10, March 2016.

[55] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A Theory of How Columns in the Neocortex Enable Learning the Structure of the World. *Frontiers in Neural Circuits*, 11:81, October 2017.

[56] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. A theory of how columns in the neocortex enable learning the structure of the world. *Frontiers in neural circuits*, page 81, 2017.

[57] Jeff Hawkins, Subutai Ahmad, and Yuwei Cui. Why Does the Neocortex Have Columns, A Theory of Learning the Structure of the World. preprint, Neuroscience, July 2017.

[58] Jeff Hawkins, Marcus Lewis, Mirko Klukas, Scott Purdy, and Subutai Ahmad. A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex. *Frontiers in Neural Circuits*, 12:121, January 2019.

[59] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[61] Petr Hluštík, Ana Solodkin, Douglas C Noll, and Steven L Small. Cortical plasticity during three-week motor skill learning. *Journal of clinical neurophysiology*, 21(3):180–191, 2004.

[62] Kjell Jørgen Hole and Subutai Ahmad. A thousand brains: toward biologically constrained ai. *SN Applied Sciences*, 3(8):1–14, 2021.

[63] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

[64] Clark Leonard Hull. Principles of behavior: an introduction to behavior theory. 1943.

[65] Mingxiao Huo, Mingyu Ding, Chenfeng Xu, Thomas Tian, Xinghao Zhu, Yao Mu, Lingfeng Sun, Masayoshi Tomizuka, and Wei Zhan. Human-oriented representation learning for robotic manipulation. *arXiv preprint arXiv:2310.03023*, 2023.

[66] Khawar Islam, L Minh Dang, Sujin Lee, and Hyeonjoon Moon. Image compression with recurrent neural network and generalized divisive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1875–1879, 2021.

[67] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[68] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019.

[69] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: Robot manipulation with multimodal prompts. 2023.

[70] Mark Johnson. Embodied understanding. *Frontiers in psychology*, 6:875, 2015.

[71] Mark Johnson. *Embodied mind, meaning, and reason: How our bodies give rise to understanding*. University of Chicago Press, 2017.

[72] Philip N Johnson-Laird. Mental models and probabilistic thinking. *Cognition*, 50(1-3):189–209, 1994.

[73] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[74] Jerome Kagan. Motives and development. *Journal of personality and social psychology*, 22(1):51, 1972.

[75] Namasivayam Kalithasan, Sachit Sachdeva, Himanshu Gaurav Singh, Divyanshu Aggarwal, Gurarmaan Singh Panjeta, Vishal Bindal, Arnav Tuli, Rohan Paul, and Parag Singla. Sketch-plan-generalize: Continual few-shot learning of inductively generalizable spatial concepts for language-guided robot manipulation. *arXiv preprint arXiv:2404.07774*, 2024.

[76] Immanuel Kant. Critique of pure reason. 1781. *Modern Classical Philosophers, Cambridge, MA: Houghton Mifflin*, pages 370–456, 1908.

[77] Kuno Kim, Megumi Sano, Julian De Freitas, Nick Haber, and Daniel Yamins. Active world model learning with progress curiosity. In *International conference on machine learning*, pages 5306–5315. PMLR, 2020.

[78] Natasha Z Kirkham, Jonathan A Slemmer, and Scott P Johnson. Visual statistical learning in infancy: Evidence for a domain general learning mechanism. *Cognition*, 83(2):B35–B42, 2002.

[79] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pages 128–135. IEEE, 2005.

[80] Tomas Kulvicius, KeJun Ning, Minija Tamosiunaite, and Florentin Worgötter. Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Transactions on Robotics*, 28(1):145–157, 2011.

[81] George Lakoff and Mark Johnson. *Metaphors we live by*. University of Chicago press, 2008.

[82] George Lakoff, Mark Johnson, and John F Sowa. Review of philosophy in the flesh: The embodied mind and its challenge to western thought. *Computational Linguistics*, 25(4):631–634, 1999.

[83] Niels Leadholm, Marcus Lewis, and Subutai Ahmad. Grid cell path integration for movement-based visual object recognition. *arXiv preprint arXiv:2102.09076*, 2021.

[84] Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

[85] Kwangjun Lee, Shirin Dora, Jorge F Mejias, Sander M Bohte, and Cyriel MA Pennartz. Predictive coding with spiking neurons and feedforward gist signaling. *Frontiers in Computational Neuroscience*, 18:1338280, 2024.

[86] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. In *Deep Reinforcement Learning Workshop NeurIPS 2022*.

[87] Donald M MacKay. The epistemological problem for automata. *Automata studies*, pages 235–51, 1956.

[88] Kanika Madan, Rosemary Nan Ke, Anirudh Goyal, Bernhard Schölkopf, and Yoshua Bengio. Fast and slow learning of recurrent independent mechanisms. In *The 9th International Conference on Learning Representations (ICLR 2021)*. OpenReview. net, 2021.

[89] Yutaka Matsuo, Yann LeCun, Maneesh Sahani, Doina Precup, David Silver, Masashi Sugiyama, Eiji Uchibe, and Jun Morimoto. Deep learning, reinforcement learning, and world models. *Neural Networks*, 152:267–275, 2022.

[90] JH Maunsell and DAVID C van Essen. The connections of the middle temporal visual area (mt) and their relationship to a cortical hierarchy in the macaque monkey. *Journal of Neuroscience*, 3(12):2563–2586, 1983.

[91] Ryan McCall and Stan Franklin. Cortical learning algorithms with predictive coding for a systems-level cognitive architecture. 2013.

[92] HUNT J McV. Intrinsic motivation and its role in psychological development. In *Nebraska symposium on motivation*, volume 13, pages 189–282. University of Nebraska Press, 1965.

[93] Beren Millidge, Anil Seth, and Christopher L Buckley. Predictive coding: a theoretical and experimental review. *arXiv preprint arXiv:2107.12979*, 2021.

[94] Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *International Conference on Machine Learning*, pages 6972–6986. PMLR, 2020.

[95] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

[96] Vernon B Mountcastle. The columnar organization of the neocortex. *Brain: a journal of neurology*, 120(4):701–722, 1997.

[97] Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktäschel, and Edward Grefenstette. Improving intrinsic exploration with language abstractions. *Advances in Neural Information Processing Systems*, 35:33947–33960, 2022.

[98] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023.

[99] Ulric Neisser. *Cognitive psychology: Classic edition*. Psychology press, 1967.

[100] Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487, 2004.

[101] Randall C O'Reilly, Jacob L Russin, Maryam Zolfaghar, and John Rohrlich. Deep predictive learning in neocortex and pulvinar. *Journal of Cognitive Neuroscience*, 33(6):1158–1196, 2021.

[102] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:108, 2007.

[103] Jyothish Pari, Nur Muhammad Shafiullah, Sridhar Pandian Arunachalam, and Lerrel Pinto. The surprising effectiveness of representation learning for visual imitation. *arXiv preprint arXiv:2112.01511*, 2021.

[104] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *international conference on machine learning*, pages 17359–17371. PMLR, 2022.

[105] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[106] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International conference on machine learning*, pages 5062–5071. PMLR, 2019.

[107] Charles Sanders Peirce. *Collected papers of charles sanders peirce*, volume 5. Harvard University Press, 1974.

[108] Cyriel MA Pennartz. *The brain's representational power: on consciousness and the integration of modalities*. MIT Press, 2015.

[109] Jean Piaget. *Piaget's theory of intelligence*. Englewood Cliffs, NJ: Prentice Hall, 1978.

[110] John Piaget. The origins of intelligence in children. *International University*, 1952.

[111] Zygmunt Pizlo. Perception viewed as an inverse problem. *Vision research*, 41(24):3145–3161, 2001.

[112] Clare Press, Peter Kok, and Daniel Yon. The perceptual prediction paradox. *Trends in Cognitive Sciences*, 24(1):13–24, 2020.

[113] Scott Purdy. Encoding data for htm systems. *arXiv preprint arXiv:1602.05925*, 2016.

[114] 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*, pages 8748–8763. PMLR, 2021.

[115] Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pages 7953–7963. PMLR, 2020.

[116] Stephen Walter Ranson and Sam Lillard Clark. The anatomy of the nervous system. its development and function. *Academic Medicine*, 34(5):553, 1959.

[117] Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

[118] E. Rohmer, S. P. N. Singh, and M. 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.

[119] Edmund T Rolls and Martin J Tovee. Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *Journal of neurophysiology*, 73(2):713–726, 1995.

[120] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[121] Meredith L Rowe and Susan Goldin-Meadow. Differences in early gesture explain ses disparities in child vocabulary size at school entry. *Science*, 323(5916):951–953, 2009.

[122] Meredith L Rowe, Şeyda Özçalışkan, and Susan Goldin-Meadow. Learning words by hand: Gesture's role in predicting vocabulary development. *First language*, 28(2):182–199, 2008.

[123] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[124] Richard M Ryan and Edward L Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary educational psychology*, 25(1):54–67, 2000.

[125] Cansu Sancaktar, Sebastian Blaes, and Georg Martius. Curious exploration via structured world models yields zero-shot object manipulation. *Advances in Neural Information Processing Systems*, 35:24170–24183, 2022.

[126] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[127] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[128] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018.

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

[130] Pedro Sequeira, Francisco S Melo, and Ana Paiva. Emotion-based intrinsic motivation for reinforcement learning agents. In *International conference on affective computing and intelligent interaction*, pages 326–336. Springer, 2011.

[131] Rutav M Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. In *International Conference on Machine Learning*, pages 9465–9476. PMLR, 2021.

[132] Syamimi Shamsuddin, Luthffi Idzhar Ismail, Hanafiah Yussof, Nur Ismarrubie Zahari, Saiful Bahari, Hafizan Hashim, and Ahmed Jaffar. Humanoid robot nao: Review of control and motion exploration. In *2011 IEEE International Conference on Control System, Computing and Engineering*, pages 511–516, 2011.

[133] Lin Shao, Toki Migimatsu, Qiang Zhang, Karen Yang, and Jeannette Bohg. Concept2robot: Learning manipulation concepts from instructions and human demonstrations. *The International Journal of Robotics Research*, 40(12-14):1419–1434, 2021.

[134] Lawrence Shapiro. *Embodied cognition*. Routledge, 2019.

[135] Michael Shermer. *The believing brain: From ghosts and gods to politics and conspiracies—How we construct beliefs and reinforce them as truths*. Macmillan, 2011.

[136] Stewart Shipp. Neural elements for predictive coding. *Frontiers in psychology*, 7:1792, 2016.

[137] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on robot learning*, pages 894–906. PMLR, 2022.

[138] Andrew Silva, Nina Moorman, William Silva, Zulfiqar Zaidi, Nakul Gopalan, and Matthew Gombolay. Lancon-learn: Learning with language to enable generalization in multi-task manipulation. *IEEE Robotics and Automation Letters*, 7(2):1635–1642, 2021.

[139] Avi Singh, Larry Yang, and Sergey Levine. Gplac: Generalizing vision-based robotic skills using weakly labeled images. In *Proceedings of the IEEE international conference on computer vision*, pages 5851–5860, 2017.

[140] Satinder Singh, Richard L Lewis, Andrew G Barto, and Jonathan Sorg. Intrinsically motivated reinforcement learning: An evolutionary perspective. *IEEE Transactions on Autonomous Mental Development*, 2(2):70–82, 2010.

[141] Shivang Singh and Jie Hie Liao. Concept2robot 2.0: Improving learning of manipulation concepts using enhanced representations.

[142] Michael W Spratling. A review of predictive coding algorithms. *Brain and cognition*, 112:92–97, 2017.

[143] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[144] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[145] Julius Taylor and Jochen Triesch. Using models in intrinsically motivated reinforcement learning. In *Master Tesis*. Frankfurt Institute for Advanced Studies, 2021.

[146] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, et al. Octo: An open-source generalist robot policy, 2023.

[147] Joshua B Tenenbaum, Thomas L Griffiths, and Charles Kemp. Theory-based bayesian models of inductive learning and reasoning. *Trends in cognitive sciences*, 10(7):309–318, 2006.

[148] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[149] Lev S Vygotsky. *Mind in society: The development of higher psychological processes*, volume 86. Harvard university press, 1978.

[150] Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European Conference on Computer Vision*, pages 55–72. Springer, 2024.

[151] Zhou Wang, Alan C Bovik, and Hamid R Sheikh. Structural similarity based image quality assessment. In *Digital Video image quality and perceptual coding*, pages 225–242. CRC Press, 2017.

[152] Robert W White. Motivation reconsidered: the concept of competence. *Psychological review*, 66(5):297, 1959.

[153] Margaret Wilson. Six views of embodied cognition. *Psychonomic bulletin & review*, 9:625–636, 2002.

[154] Philipp Wu, Alejandro Escontrela, Danijar Hafner, Pieter Abbeel, and Ken Goldberg. Daydreamer: World models for physical robot learning. In *Conference on robot learning*, pages 2226–2240. PMLR, 2023.

[155] Robert H Wurtz. Recounting the impact of hubel and wiesel. *The Journal of physiology*, 587(12):2817–2823, 2009.

[156] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[157] Annie Xie, Lisa Lee, Ted Xiao, and Chelsea Finn. Decomposing the generalization gap in imitation learning for visual robotic manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3153–3160. IEEE, 2024.

[158] Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *NeurIPS 2023 Workshop on Generalization in Planning*.

[159] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7286–7293. IEEE, 2020.

[160] Youssef Zaky, Gaurav Paruthi, Bryan Tripp, and James Bergstra. Active perception and representation for robotic manipulation. *arXiv preprint arXiv:2003.06734*, 2020.

[161] Semir Zeki and Stewart Shipp. The functional logic of cortical connections. *Nature*, 335(6188):311–317, 1988.

[162] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[163] Albert Zhan, Ruihan Zhao, Lerrel Pinto, Pieter Abbeel, and Michael Laskin. A framework for efficient robotic manipulation. In *Deep RL Workshop NeurIPS 2021*, 2021.

[164] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Zhengping Che, Chaomin Shen, Yaxin Peng, Dong Liu, Feifei Feng, et al. Language-conditioned robotic manipulation with fast and slow thinking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4333–4339. IEEE, 2024.