POLITECNICO DI TORINO

Master's Degree in Artificial Intelligence and Data Analytics

Master's Thesis

Inferring Complex Dynamics through Core Knowledge

Experiments in a Game Environment

Supervisors
Giovanni Squillero
Stefano Quer
Alberto Tonda

Candidate
Giorgio Mongardi
Student ID 292490

Academic Year 2024-2025

Abstract

The ability to generalize knowledge across similar environments is a key aspect of human intelligence. Unlike humans, current artificial intelligence (AI) models struggle to extract structured, transferable knowledge from dynamic environments. DeepMind's breakthrough results on Atari games have demonstrated that pure sub-symbolic approaches, such as deep reinforcement learning, lack true understanding: they optimize policies based on raw pixel inputs but fail to generalize when minimal changes are introduced to the environment. This brittleness—where even trivial modifications, such as recoloring a paddle or shifting an object's position, require retraining from scratch—suggests that these models do not learn in the human sense but instead rely on fragile heuristics. As argued by Melanie Mitchell, such systems lack an internal "world model", a structured representation of cause-effect relationships that allows for reasoning and adaptation.

This thesis presents an alternative framework for capturing interpretable knowledge about game dynamics without relying on opaque neural networks. Instead of passively learning from vast amounts of data, our system actively constructs structured models of object behavior using heuristic-based reasoning grounded in core knowledge. The method does not assume pre-defined objects but begins with anonymous patches extracted from game frames, identified solely by their fundamental properties (e.g., position, shape). A set of heuristics then tracks these patches over time, recognizing persistent entities and their interactions. Through rule inference, the system identifies regularities in object behavior, reconstructing the underlying mechanics governing the environment.

These structured representations are abstract and reusable, meaning they could, in principle, transfer across related environments without requiring retraining. Unlike deep learning models, which discard prior knowledge when faced with novel conditions, our system retains and adapts its understanding dynamically. This approach lays the groundwork for AI systems capable of interpretable, human-like reasoning about interactive environments.

The model can potentially be integrated into reinforcement learning (RL) frameworks, serving as an "internal world model" for agents in Dyna-like architectures. Preliminary experiments indicate that leveraging structured knowledge in planning improves sample efficiency and robustness to environmental changes, suggesting promising directions for future work.

Contents

Chapter 1

Introduction

It has been repeated extensively how much progress artificial intelligence has made in recent years; thanks to a combination of devoted research, increased computational power, and vast amount of data, deep learning models have indeed achieved outstanding results in many fields, from text to video generation to 3D protein folding, even surpassing humans in some narrow tasks. It is only fair to be proud of these successes, but to progress further we also need to understand the limitations of these solutions.

Many researchers have highlighted that, despite being exceptionally capable in learning patterns that solve a given task, they still lack some crucial aspects of what we usually intend as intelligence. One of the most prominent critiques is the lack of understanding [77][48], which means that deep learning models learn how to solve tasks, without learning what the task is and how they solve it; they do not develop an internal world model consisting of concepts, hierarchies, and causal reasoning. Another discussed and possibly related limitation is the lack of generalization [14]; most models are narrow and brittle; to face novel challenges or changes in their environment, they usually need fine-tuning or even retraining. Then, there is the problem of their opacity [25]; while some techniques to try and understand which pattern these models follow when making decisions are being studied, in most cases they remain black boxes from the outside. This lack of transparency limits AI in many fields that require reliability and accountability, such as medicine, economy, and, in general, decision making.

The starting point of this thesis was to create a framework able to develop

a structured and symbolic representation of its environment, providing explainability and interpretability, while investigating how to build an internal world model. The initial case of study was DeepMind's Atari-playing AI [49]; this model achieved superhuman performance in these classic video games but perfectly showed the limitations mentioned above, specifically brittleness [55], lack of generalization [46], and black-box nature; when even small changes are made to the game environment, such as recoloring or resizing an element, the model has trouble playing, showing how much it depends on statistical patterns at the raw pixel level. Clearly, the objective of this research was not to reach DeepMind's performances but to lay the foundations for a more structured and generalized learning framework.

Overall, this work investigates the open problem of learning, drawing from Philosophy, Psychology, and Cognitive Science to highlight how the theme influences artificial intelligence, and proposes a novel framework for capturing interpretable knowledge about game dynamics using core knowledge and symbolic learning. By constructing formal representations of elements and their behaviors, the goal is to develop a robust understanding of in-game mechanics that remains valid even when the environment is altered.

The proposed method defines a core knowledge base on fundamental concepts such as shape, motion, and contact. The system process anonymous visual elements extracted from game frames, identified solely by their intrinsic properties, and recognize persistent entities and their behaviors in the environment, generalizing them in abstract and adaptive classes to recognize analogous entities in new games and environments. This approach contributes to laying the foundations for AI systems capable of interpretable human-like reasoning about interactive environments.

Atari games, particularly Arkanoid, serve as the primary case study because they are conceptually simple while still capturing key aspects of AI behavior in interactive environments [17]; They provide an environment with clear and discrete rules that can be learned and many similar others across which knowledge could potentially be transferred. Furthermore, using these environments leaves open the possibility to integrate reinforcement learning in the frameworks, employing the inferred knowledge as an "internal world model" for agents in Dyna-like architectures.

Research Questions and Objectives

This thesis explores the following themes, some identified at the beginning of the project and others that emerged during the development:

1. Understanding the ontological foundations of learning and how they relate to artificial systems.
2. Investigating the current limitations of deep learning approaches and the proposed alternatives.
3. Exploring the potential role of core knowledge principles in symbolic and neuro-symbolic AI models.
4. Design and implementation of a framework capable of constructing an interpretable reconstruction of its environment from low-level perceptual data.
5. Expanding the framework by drawing on insights from recent developments in cognitive science and AI research.

Chapter 2

Background

2.1 Defining Learning

Before exploring what it means for a machine to learn, it is useful to analyze how this theme is approached in humans. Learning is one of the most fundamental aspects of human cognition, yet its nature and mechanisms have been widely debated in philosophy, psychology, and cognitive science. At its core, it is defined as "the process of acquiring new and relatively enduring information or behaviors" [51], but the means by which this happens remain a topic of rich intellectual exploration. From ancient philosophical inquiries to contemporary research in cognitive science, scholars have sought to define learning in ways that capture its complexity.

Historically, the question of how knowledge is acquired has been framed by the rationalism-empiricism debate, which considers whether learning is primarily driven by innate structures of the mind or by experience and observation. This debate has deeply influenced modern learning theories, each offering distinct perspectives on how knowledge is formed and internalized. Moreover, the rise of artificial intelligence has intensified the need for new approaches to learning, bridging traditional theories with advanced computational models that emulate human cognition.

This section briefly explores the philosophical and psychological foundations of learning, beginning with the epistemological divide between rationalism and empiricism before examining how these perspectives shape contemporary learning theories and AI-driven models of knowledge acquisition. By tracing the evolution of these ideas, we aim to understand what it truly

means to learn in both human and artificial contexts.

2.1.1 Rationalism and Empiricism: The Enduring Debate

The debate between rationalism and empiricism has shaped the course of epistemology for centuries, raising fundamental questions about the nature and sources of human knowledge. Rationalists argue that certain knowledge is innate or acquired through pure reason, whereas empiricists contend that all knowledge originates in sensory experience. This philosophical divide, which dates back to the early modern period, continues to influence contemporary discussions in epistemology, cognitive science, and even physics. Although the historical opposition between these positions remains stark, more recent analyses suggest that the rigid distinction between rationalism and empiricism may be overstated, as elements of both are often interwoven in theories of knowledge.

The Foundations of Rationalism

Rationalism, as articulated by philosophers such as René Descartes, Gottfried Wilhelm Leibniz, and Baruch Spinoza, holds that reason is the primary source of knowledge. Descartes, in his *Meditations on First Philosophy* (1641), sought to establish knowledge on an indubitable foundation. Beginning with his famous declaration "Cogito, ergo sum", he argued that through reason alone, one could arrive at certain truths independently of sensory experience. This is reflected in what has been called the Intuition/Deduction Thesis, which asserts that some knowledge, such as mathematical truths, can be grasped through rational insight and logical deduction rather than empirical observation.

Leibniz, in *New Essays on Human Understanding* (1704), further supported rationalism by emphasizing the existence of innate knowledge, propositions that the human mind possesses independently of experience. He famously critiqued John Locke's position by arguing that the mind is not a tabula rasa (i.e., a blank slate), as Locke claimed, but rather contains principles that structure experience. For Leibniz, mathematical truths, logical principles, and moral laws are examples of knowledge that are not derived from experience but rather exist a priori, as part of human cognition.

Rationalists also point to the necessity of a priori knowledge in mathematics and logic. As Immanuel Kant argued in his *Critique of Pure Reason* (1781), mathematical statements such as "7 + 5 = 12" are synthetic a priori judgments, meaning that they extend knowledge while being necessarily true, yet they do not arise from experience. This suggests that human cognition operates according to principles that cannot be reduced to empirical data alone.

One of the strongest defenses of rationalism comes from the history of mathematics itself. The development of non-Euclidean geometry by János Bolyai, Carl Friedrich Gauss, and Nikolai Lobachevsky in the 19th century revealed that mathematical reasoning can extend beyond empirical verification. While Euclidean geometry had long been considered an unshakable truth, independent mathematical reasoning demonstrated the logical consistency of alternative geometries, some of which later found application in Einstein's theory of General Relativity.

The Empirical Challenge

In contrast, empiricists argue that all knowledge arises from experience. John Locke, George Berkeley, and David Hume advanced this perspective, emphasizing that the mind at birth is a blank slate, acquiring knowledge only through sensory perception. Locke, in particular, firmly denied the existence of innate knowledge, insisting that all ideas come from either sensation (i.e., direct perception of external objects) or reflection (i.e., internal observation of mental processes). He famously challenged rationalists to identify a single universally accepted innate idea, arguing that no such knowledge exists independent of experience.

Hume took empiricism further by questioning the very notion of causality. In *An Enquiry Concerning Human Understanding* (1748), he argued that the human mind does not perceive causation directly but instead infers it from repeated associations of events. This radical skepticism undermined the rationalist claim that we can arrive at necessary truths through pure reason. According to Hume, what we consider necessary truths, such as "the sun will rise tomorrow", are merely habitual expectations based on past experience, not logical certainties.

Empirical science has long supported this viewpoint, particularly through its reliance on observation and experimentation. Galileo Galilei and Isaac

Newton demonstrated that empirical evidence, rather than pure reason, is the key to understanding the natural world. Newton's *Principia Mathematica* (1687) revolutionized physics by grounding it in empirical laws rather than rationalist speculation. Modern science continues this legacy, with experimental methods forming the backbone of disciplines ranging from quantum mechanics to cognitive psychology.

A significant modern challenge to rationalism comes from research in cognitive science. Studies have shown that even what appear to be innate concepts may actually be shaped by experience. For example, Stanislas Dehaene, in *The Number Sense* [13], argues that human mathematical abilities arise from a biologically evolved "number sense" rather than purely rational intuition. Similarly, experiments with infants suggest that while they may have certain expectations about the physical world, these are best understood as early learned responses rather than evidence of innate knowledge.

The Modern Synthesis: Reconciling Rationalism and Empiricism

While the historical debate between rationalism and empiricism presents them as opposing schools of thought, contemporary philosophy and cognitive science suggest a more nuanced picture. The traditional empiricist rejection of innate knowledge has been softened by findings in developmental psychology and neuroscience, which indicate that humans do have certain built-in cognitive structures. However, these are increasingly seen as shaped by evolutionary experience rather than as Kantian a priori truths.

At the same time, the rationalist emphasis on deductive reasoning remains crucial, especially in fields like mathematics and logic, where empirical observation alone cannot account for all knowledge. The ability to grasp necessary truths and construct abstract models that go beyond immediate experience remains a central part of human cognition.

Ultimately, the persistence of the rationalism-empiricism debate, as noted by Benjamin Murphy [50], suggests that neither position can fully account for the complexity of human knowledge on its own. While rationalists correctly highlight the necessity of logical reasoning and innate structures, empiricists provide an essential corrective by emphasizing the role of experience and scientific verification. As the study of knowledge continues to evolve, a more integrative approach, one that acknowledges the contributions of both reason and experience, seems to be the most promising path forward.

2.1.2 From Epistemology to Learning Theories

From this historical debate many theories about how knowledge is acquired have been derived, notably Behaviorism, Cognitivism, and Constructivism, each reflecting a different assumption. These theories are of great interest in the context of artificial intelligence, because they can be used to explain how current models are learning and what they are still lacking.

Behaviorism: Learning as a Conditioned Response

Behaviorism, developed by Watson, Pavlov, and Skinner, is deeply rooted in empiricist philosophy. It posits that all knowledge and learning result from experience and interaction with the environment, rejecting the idea of innate knowledge or rational intuition. According to behaviorists, the mind is a blank slate at birth, and knowledge is acquired through conditioning and reinforcement.

Classical conditioning, demonstrated by Pavlov's famous experiments with dogs, illustrates how learning occurs through repeated associations. When a neutral stimulus, such as a bell, is paired with an unconditioned stimulus, such as food, multiple times, it eventually elicits a conditioned response, such as salivation. This principle applies to human learning as well, where repeated exposure to stimuli leads to habitual responses.

Expanding on this, Skinner introduced operant conditioning, which describes learning as a process of reinforcement and punishment. Positive reinforcement strengthens desired behaviors, while negative reinforcement or punishment discourages undesired actions. In practice, this principle is applied through rewards, grades, and feedback mechanisms, shaping agent behavior through structured experiences.

Cognitivism and Constrictivism: Learning as Mental Processing

In the 20th century, drawing from rationalist epistemology, critiques to Behaviorism began to emerge; scholars, notably Bartlett and Piaget, started developing theories which argued that learning is not just a result of stimulus-response but an internal process involving memory, problem-solving, and structured thinking.

Bartlett coined the term "schema", representing a structured unit of knowledge that is shaped by experience for a subject or an event. In parallel, also Jean Piaget worked on the same theme and proposed his famous theory of cognitive development. His work introduced the concept for which children pass through distinct stages of learning, where they actively process information and build internal mental models of reality. In this context, a child could create an internal representation (i.e., schema) for a dog (e.g., four legs, furry), mistake a cat for a dog because it fits the schema (i.e., assimilation), then notice some differences (e.g., barking in contrast to meowing) and change its understanding (i.e., accomodation), either by creating a new schema or by modifying a pre-existing one.

From this foundation, two related yet distinct learning theories developed: cognitivism and constructivism. Both theories emphasize the active role of learners in processing and interpreting information, highlighting the importance of internal cognitive processes over passive absorption. However, they diverge significantly in their views regarding knowledge creation. Cognitivism conceptualizes knowledge as structured and objective, emphasizing the role of cognitive functions such as memory, attention, and information processing mechanisms. In contrast, constructivism considers knowledge to be subjective, constructed by individuals as they engage dynamically with their environment and social interactions.

Noam Chomsky further supported rationalist ideas with his theory of universal grammar. In his extensive work on language, he contrasted Skinner's *Verbal Behavior* (1957), starting the decline of linguistic behaviorism.

However, in these theories, empiricism is not discarded; experience is recognized to have a critical role in refining and shaping cognitive structures. For example, Jerome Bruner's discovery learning emphasized that learning is a process guided by exploration, interactions and active problem-solving.

In summary, unlike behaviorism, which focuses on external reinforcement, cognitivism and constructivism prioritizes internal understanding of experiences, making them a bridge between rationalist and empiricist views of learning. The difference stand in the fact that cognitivists assume that knowledge can be structured and transmitted, while constructivists argue that knowledge is always subjective and context-dependent, meaning that a concept is shaped by individual experiences, social interactions, and cultural backgrounds rather than being universally defined.

Cognitive Science: An Interdisciplinary Understanding of Learning

Constructivist and cognitivist theories were part of a growing intellectual movement called the "cognitive revolution", which emerged in the mid-20th century, culminating in the birth of cognitive science. The term cognitive science was coined by Christopher Longuet-Higgins in a report on the state of artificial intelligence, subsequently leading to the founding of the journal *Cognitive Science* and the Cognitive Science Society. Since then, the aim of this new field has been the interdisciplinary scientific study of the mind and its processes, drawing from psychology, economics, artificial intelligence, neuroscience, linguistics, and anthropology.

This subject investigates how beings acquire, represent, manipulate, and use knowledge. Its scope bridges ancient philosophical questions about the nature of thought and knowledge with modern computational models and neuro-biological mechanisms.

Cognitive scientists often adopt a functionalist view, focusing on the roles mental processes serve rather than on the exact physical forms they take.

A fundamental principle of cognitive science is that mental processes should be examined across multiple levels of analysis. David Marr [43] famously proposed three levels of understanding in information processing systems: computational level (i.e., what is the task), algorithmic level (i.e. representation of inputs, outputs and the process that manipulates them), and implementational level (physical realization).

Classically, cognitive science focused primarily on internal cognitive processes, notably memory, attention mechanisms, language processing, perception, learning and consciousness, but, with time, social, cultural, and emotional factors have been taken more in consideration.

In AI, cognitive science provides both inspiration and structure. Consider the way language models function: they are trained to predict the next word based on patterns in large data sets. But the underlying architecture, transformers, attention mechanisms, and embeddings, are attempts to replicate cognitive functions like working memory, attention allocation, and semantic similarity. These mechanisms are not just engineering tricks; they are shaped by decades of cognitive research on how humans process and generate language.

In summary, cognitive science is both a theoretical inquiry into the nature of knowledge and a practical toolkit for building intelligent systems.

It studies perception, memory, reasoning, language, learning, emotion, and consciousness, both in humans and machines. Its promise lies not only in helping us understand ourselves but also in shaping the future of intelligent technology

2.2 The Evolution of Learning in Artificial Intelligence

The study of learning, once confined to philosophy and cognitive science, has expanded into the realm of artificial intelligence, where machines are now capable of processing information, recognizing patterns, and adapting to new data. Just as human learning theories have been shaped by the debate between rationalism and empiricism, AI development has been influenced by similar tensions between explicit knowledge representation and data-driven pattern recognition.

At its core, AI seeks to automate learning, a process that has traditionally been considered a hallmark of human intelligence. Early AI systems attempted to encode knowledge using strict logical rules, akin to rationalist reasoning, but struggled with flexibility and real world uncertainty. The emergence of machine learning shifted the focus to data-driven learning, much like empiricist approaches that prioritize experience and adaptation. More recently, hybrid models, such as Neuro-Symbolic AI, attempt to combine reasoning with learning, echoing the Kantian synthesis that reconciles rational structure with empirical input.

This section explores the evolution of AI through the lens of learning, tracing its development from early symbolic reasoning systems to modern deep learning frameworks. By examining how AI systems learn, whether through explicit programming, data-driven adaptation, or reinforcement mechanisms, we gain deeper insight into both the strengths and limitations of artificial intelligence as a model of cognition. Ultimately, understanding AI's learning mechanisms helps us contextualize its capabilities and constraints, as well as the broader question of whether machines can truly replicate human-like intelligence

2.2.1 A Brief History of AI and Learning Models

The evolution of artificial intelligence has been marked by shifting paradigms in how machines learn and process information. From its early foundations in symbolic reasoning to the emergence of connectionist neural networks, AI's development reflects the ongoing tension between explicitly structured knowledge and experience-driven adaptation. The epistemological divide between rationalism and empiricism, previously explored in human learning theories, is mirrored in AI's trajectory. This historical progression highlights how different approaches to machine learning have shaped the field and informs current research on AI.

The Symbolic AI Era

The earliest AI systems were built on the principles of symbolic reasoning, heavily influenced by rationalist epistemology. In this paradigm, intelligence was understood as the ability to manipulate symbols according to logical rules, mirroring the way rationalists believed human reasoning operates independently of sensory experience. AI researchers attempted to encode knowledge explicitly, designing systems that could reason through logic-based inference rather than learning from raw data.

A foundational intellectual precursor to this approach was Alan Turing's formulation of the Turing machine [73], an abstract computational model that demonstrated how symbolic manipulation could, in theory, simulate any computation. Turing's work provided a formal basis for the idea that reasoning and problem-solving could be reduced to rule-based symbol processing, a notion that strongly influenced early AI research. Additionally, his 1950 paper, *Computing Machinery and Intelligence* [72], posed the question "Can machines think?" and introduced what is now known as the Turing Test, a behavioral criterion for assessing machine intelligence through symbolic communication. Turing's ideas legitimized the pursuit of machine reasoning and set the philosophical and technical groundwork for symbolic approaches to AI.

A landmark development in this era was the General Problem Solver [52], created by Allen Newell and Herbert Simon, which aimed to model human problem-solving using formal logic. This approach laid the foundation for expert systems, such as MYCIN [67] in medical diagnosis, which functioned

by following if-then rule-based logic to provide decisions. These systems represented knowledge in a way that was deterministic and explainable, making them reliable for domains with well-defined rules; however, symbolic AI had significant limitations, struggling with uncertainty, ambiguity, and brittleness. Unlike humans, who can adapt to novel situations and generalize beyond prior knowledge, symbolic AI lacked flexibility and learning capabilities.

The Connectionist Revolution and Neural Networks

The limitations of symbolic AI led to a interest in empirical learning models, giving rise to connectionism, which drew inspiration from neuroscience rather than formal logic. Unlike symbolic systems, which relied on explicit knowledge encoding, connectionist models sought to simulate human learning through networks of artificial neurons, aligning more closely with empiricist theories of knowledge acquisition.

A major breakthrough came with the rediscovery of neural networks and backpropagation. Although early neuron models had been proposed by Warren McCulloch and Walter Pitts in 1943 [44] and formalized into perceptrons by Frank Rosenblatt in 1958 [63][62], they were initially dismissed after Marvin Minsky and Seymour Papert (1969) [45] proved that they had severe limitations in learning complex patterns. They were specifically referring to single-layer perceptrons and their inability in learning not linearly separable functions, but this resulted in a decline in interest and in fundings. However, in the 1980s, multi-layer perceptrons were reintroduced thanks to the development of backpropagation, which allowed to optimize the weights of hidden layers. More than one scholar [37][76][64] derived backpropagation in that period, because it represent an efficient application of Leibniz's chain rule.

Unlike symbolic AI, which depended on handcrafted rules, neural networks could extract patterns from examples, making them particularly effective for pattern recognition tasks. Early applications included handwritten digit recognition and speech processing, where neural networks outperformed traditional rule-based approaches. However, these models required large amounts of labeled data and computational resources, which at the time limited their scalability.

The Machine Learning Boom

At end of the century, as computers became popular and more powerful, the era of big data began. One of the defining characteristics of this period was the shift from knowledge-driven approaches to data-driven approaches. New techniques emerged, such as support vector machines and random forests, and began to be used for tasks like spam filtering, recommendation systems, and fraud detection.

In 1997, the world chess champion was beaten by deepblue [8], a model which combined traditional expert systems with machine learning approaches. This was a great success, because chess had long been regarded as a good measurement of intelligence.

The Deep Learning Era and AI Autonomy

Deep learning can be broadly described as the branch of machine learning models that make use of multi-layered neural architectures, and while the most prominent researches dates back to the previous century, it was only after the diffusion of GPUs that they became competitive.

A major turning point came with AlexNet in 2012 [33], which used CNNs [19] to achieve state-of-the-art performance in image classification. This demonstrated that deep networks, when trained on large datasets and with enough computational power, could outperform traditional machine learning approaches. Following this, architectures such as ResNet, VGG, and EfficientNet further refined image recognition tasks, for example performing face recognition.

In natural language processing, the evolution of recurrent neural networks and, later, the introduction of transformers [74] led to breakthroughs in language understanding. Models like GPT (i.e., Generative Pre-trained Transformer) [56] and BERT (i.e., Bidirectional Encoder Representations from Transformers) [15] enabled AI to perform text generation, translation, and question answering with human-like proficiency.

One of the latest achievement was the prediction of 3D protein folding by DeepMind's AlphaFold in 2018, later improved in 2021 [29].

2.2.2 Major AI Frameworks and Learning Paradigms

As artificial intelligence has evolved, different learning paradigms have emerged, each reflecting distinct approaches to how machines acquire knowledge. These paradigms align, to varying degrees, with the previously mentioned epistemological perspectives. Some models, like supervised learning, rely on structured datasets and explicit guidance, akin to cognitivist approaches, while others, like reinforcement learning, mirror behaviorist principles of learning through interaction and feedback. Additionally, unsupervised and generative learning introduce elements of constructivism, where AI independently discovers patterns and relationships within data.

This section explores the major learning frameworks that define contemporary AI, examining their mechanisms, strengths, and limitations. As AI moves toward more autonomous and adaptable intelligence, these paradigms will continue to shape the field, influencing the extent to which machines can replicate human-like learning and reasoning.

Supervised Learning: Knowledge from Labeled Data

Supervised learning is one of the most widely used AI frameworks, where models learn from explicitly labeled datasets, mapping inputs to corresponding outputs. This paradigm is closely related to cognitivist theories of learning, who emphasize the structured organization of knowledge and systematic reasoning based on prior information. Just as humans acquire knowledge through instruction, feedback, and correction, supervised learning trains models through annotated examples, refining their ability to recognize patterns and make predictions.

In supervised learning, an algorithm is fed pairs of input and output data, learning a distribution that maps the inputs to the correct outputs. The model iteratively adjusts its parameters by minimizing the difference between its predictions and the actual labels using loss functions (e.g., mean squared error for regression, cross-entropy for classification). This process, often optimized through gradient descent and backpropagation, enables AI to improve its predictions over successive training cycles.

Supervised learning reflects how humans internalize knowledge through structured instruction. Similarly to how a child learns to recognize objects by being shown labeled pictures and corrected when making mistakes, drawing a

parallel to cognitivist theories. However, unlike humans, who can generalize from limited examples, traditional supervised learning requires large datasets to achieve robust performance and are often limited to i.i.d (i.e., independent and identically distributed) assumptions.

Unsupervised Learning: Discovering Patterns Without Labels

Unsupervised learning represents a fundamental shift in how AI acquires knowledge, moving away from reliance on explicit labels and towards autonomous pattern discovery. Unlike supervised learning, where models are provided with predefined answers, unsupervised learning seeks to identify intrinsic structures in data without external guidance. This aligns closely with constructivist theories of learning, which emphasize that knowledge is not simply transmitted but actively constructed through experience.

At its core, unsupervised learning encompasses a range of techniques that allow AI to extract meaningful representations from unstructured data. These methods vary in their level of autonomy and complexity, ranging from purely unsupervised approaches, such as clustering and dimensionality reduction, to self-supervised learning, where the model generates its own supervisory signals, and generative learning, where AI creates new data samples based on learned distributions. Each of these paradigms plays a distinct role in the evolution of AI, contributing to its increasing abilities.

The most fundamental form of unsupervised learning involves finding hidden patterns or structures in raw data. This process, often referred to as knowledge extraction, is crucial in tasks where there are no predefined categories or labels, requiring AI to make sense of information without explicit supervision. One of the most widely used techniques in this domain is clustering, which groups data points based on their similarities. K-means, DBSCAN, and hierarchical clustering are popular methods that allow AI to segment data into meaningful clusters, a capability extensively used in customer segmentation, anomaly detection, and biological data analysis. Similarly, dimensionality reduction techniques like Principal Component Analysis and t-SNE help AI uncover lower-dimensional structures in high-dimensional datasets, making it possible to visualize complex data distributions and extract essential features.

Moving forward, self-supervised learning (SSL) represents a hybrid approach that combines the autonomy of unsupervised learning with the structured guidance of supervised learning. Like unsupervised learning, it works with raw, unlabeled data, but unlike unsupervised methods that only find patterns, SSL creates its own predictive tasks to train a model, similar to supervised learning, just without human-provided labels. To achieve this, SSL generates pseudo-labels from the data itself (e.g., defining a task where the model must predict missing or transformed parts of its input data). Originally, it was born as a way to initialize weights before fine-tuning the model using supervised or unsupervised approaches, but the field rapidly evolved and it is now been used as a standalone paradigm.

This approach has been particularly transformative in natural language processing (NLP) and computer vision, where large amounts of data exist, but labeled examples are expensive to obtain. For instance, in NLP, models like BERT and GPT use self-supervised learning techniques such as masked language modeling, where certain tokens in an input sequence are hidden, and the model learns to predict them. Similarly, in computer vision, contrastive learning methods like SimCLR [9] and MoCo [26] teach models to recognize images by learning whether different views of an object are similar or different.

Unlike traditional unsupervised learning, which focuses on discovering natural patterns, self-supervised learning (SSL) actively trains models through structured learning tasks. This approach aligns with constructivist theories of learning, where knowledge is acquired by engaging in tasks that require the learner to actively construct understanding. This approach has made AI more data-efficient, enabling models to learn from raw data without requiring human-labeled examples. However, challenges remain, such as designing the right self-supervised tasks, which can be difficult, and the need for large-scale computing power to train these models effectively.

As AI advances, self-supervised learning is becoming a crucial step toward more autonomous intelligence, where models can learn not just to recognize patterns but also to generate meaningful insights. This capability is further extended by generative learning, which allows AI to move beyond recognition and create new data, such as text, images, and even videos; it enables AI to model the underlying structure of data and generate new records, which reflect in AI ability to generate synthetic images, text, music, and even human voices that follow the same patterns as real-world data.

This paradigm is best demonstrated by models like Generative Adversarial Networks (GANs) [21], Variational Autoencoders (VAEs) [31], and Diffusion Models [69], which have shown remarkable success in creating realistic content across various fields.

Generative learning works by estimating the probability distribution of real-world data and then sampling from it to generate new examples. In GANs, a generator creates realistic outputs, while a discriminator evaluates their authenticity, pushing both networks to improve through an adversarial unsupervised process. VAEs, on the other hand, compress data into a simplified mathematical representation (i.e., latent space) and then reconstruct it, enabling AI to generate diverse yet structured outputs. Similarly, diffusion models are trained with a forward diffusion process, which gradually adds noise, followed by a reverse denoising process, making them able to reconstruct results coherent with the learned distribution from random gaussian noise.

This technology has revolutionized AI applications in art, design, and game development. Models like DALL · E [58] and Stable Diffusion [61] can create (semi-)original artwork from text descriptions, while language models like GPT-4 [53] generate coherent human-like text responses.

However, generative learning introduces ethical and practical challenges. The ability of AI to create realistic synthetic media has raised concerns about deepfakes, misinformation, and intellectual property rights. Additionally, these models require large amounts of data, computational power and energy, making them expensive and environmentally costly to train and use at scale.

Unsupervised learning, in its various forms, highlights
AI's growing independence in acquiring and structuring knowledge. Specifically, self-supervised and generative models are driving AI beyond rigidly programmed behavior, allowing it to discover, represent, and even generate knowledge autonomously. As research progresses, these frameworks are expected to play a key role in future AI advancements.

Reinforcement Learning: Learning Through Experience

Reinforcement learning (RL) is an adaptive learning framework where AI agents interact with an environment, making decisions to maximize cumulative rewards. Inspired by behaviorist psychology, RL closely parallels Skinner's operant conditioning, where learning occurs through trial and error,

rewards, and punishments; this parallel highlights how the "reasoning" of these models follow patterns emerged by reward-based adaptation rather than an actual understanding of their environments.

In RL, an AI agent operates in a dynamic environment and follows a policy, which is a strategy that dictates which actions to take based on the current state. The learning process unfolds as follows:

- `State Observation`: The agent perceives its current environment or receives the current state.
- `Action Selection`: The agent chooses an action based on its policy.
- `Reward Feedback`: The agent receives a reward or penalty based on its action.
- `Policy Update`: The agent adjusts its strategy to maximize long-term rewards.

Mathematically, RL is based on Markov decision processes [6] and employs optimization techniques such as Q-learning [75], which search for optimal policies that maximize the expected value of the total reward, and policy optimization [78], which directly learn policies without explicitly learning value estimates.

In recent years, reinforcement learning (RL) has been significantly enhanced by deep learning, giving rise to deep reinforcement learning. One of the landmark models in this space is the Deep Q-Network (DQN) [49], which combines Q-learning with deep neural networks. In DQN, the q-function is replace by a neural network, which approximate the expected cumulative reward for taking a certain action in a given state.

A notable example is AlphaGo [68], which in 2016 defeated human champions in the game of Go, an achievement demonstrating AI's ability. Similarly, RL has been widely tested in robotics, autonomous vehicles, and financial trading systems.

In summary, RL operates within structured environments (i.e., representable with a state) with well-defined rewards and focuses on learning policies for decision-making through interaction. As RL continues to integrate with deep learning and scale to more complex settings, it holds promise for solving a wide range of real-world problems requiring autonomous behavior.

Chapter 3

Challenges in Modern AI

As artificial intelligence continues to advance, its achievements grow ever more impressive, yet its fundamental limitations remain deeply apparent. Despite surpassing human capabilities in specific tasks, AI still lacks true understanding, flexible reasoning, and reliability. Many researchers argue that today's AI models are not truly intelligent but rather powerful statistical systems that excel in controlled environments yet break down when faced with novel, ambiguous, or dynamic real-world situations.

3.1 Where AI Falls Short: Current Limitations

This growing realization has sparked intense debate among AI theorists, cognitive scientists, and machine learning pioneers. Figures such as Gary Marcus, François Chollet, Melanie Mitchell, Yann LeCun, and others have weighed in on why current AI falls short and what is required to bridge the gap between mere pattern recognition and genuine intelligence.

3.1.1 The Illusion of Understanding

Recent advancements in artificial intelligence have been remarkable, especially in the domain of natural language processing. Models such as GPT-4 and BERT have achieved unprecedented performance on a variety of linguistic tasks, from translation to question answering, showcasing near-human

accuracy on standard benchmarks.

However, this impressive performance has led to considerable hype, with some observers suggesting these large language models have begun to exhibit genuine intelligence or understanding. Amid this optimism, it is essential to critically examine such claims and remain cautious about the underlying capabilities these models truly possess.

One influential critique labels such models as "stochastic parrots", highlighting that their outputs, though seemingly intelligent, are merely sophisticated statistical pattern-matching without genuine comprehension [7]. This metaphor underscores the idea that LLMs generate text by predicting statistically probable word sequences, rather than through any deeper cognitive understanding or intentionality.

Concrete evidence supporting this critique includes the common phenomenon of model "hallucinations", where LLMs confidently produce false or nonsensical statements simply because they fit statistically plausible patterns [28]. These errors are being extensively studied in order to find a way to mitigate them. In contrast, some argue that hallucinations are intrinsic to these models [4], stating that architectural improvements, dataset enhancements, or fact-checking mechanisms are not enough to overcome this issue.

Another clear example can be found in AI-assisted content moderation, where deep learning models are often used to enforce platform policies. Social media platforms such as Facebook, YouTube, and X employ these automated moderation systems to scan vast amounts of text, images, and videos for violations, including hate speech, incitement to violence, and misinformation. However, these AI moderation tools frequently struggle with context, nuance, and cultural differences, leading to both over-censorship and under-enforcement.

Melanie Mitchell notably argues on the shortcomings of contemporary AI systems; she explored some recent solutions, providing clear explanations of their mechanisms and of their current limitations in the book *A Guide For Thinking Human* [46].

She expressed on Natural-Language Processing Systems, such as Conversational, Image Generation and, Image Captioning models. She reported how both text-based and image-based architectures failed when faced with handcrafted adversarial inputs; these sentences or images are designed to be comprehended easily by humans while confusing AI.

For sentences this imply leveraging commonsense knowledge and context-aware reasoning, such as the Winograd Schema Challenge [35], which is a linguistic reasoning task designed to evaluate an AI's ability to understand context and ambiguity. This specific challenge involves sentences where the meaning of a pronoun depends on commonsense knowledge. For example:

"The trophy did not fit in the suitcase because it was too big. What was too big?"

A human immediately understands that "it" refers to "the trophy", because we intuitively grasp the concept of size and spatial relationships. However, AI models often fail at these tasks because they do not have a conceptual understanding of physical properties like size, weight, or space, they only detect patterns in text sequences. The results in these kind of tests do not have surpassed 70%; a human should tipically score 100%.

Another case of study treated by Mitchell were DeepMind's game-playing models, for which she highlighted an important consideration on the fact that, while they use similar architecture for the different games, each one of these models have to be extensively trained on a particular game and none of this learned knowledge can be transferred, this make them extremely narrow on their training environment; the idea behind training a model to play a game is to test its ability in navigating and manipulating an environment, but, since the real world is complex and simulations cannot be perfect, she emphasizes that the ability to generalize knowledge to novel environment should be taken more in consideration when evaluating these results.

In this context, Studies on DQN robustness [55] shows how when minor visual changes are made, such as altering the background color or modifying object positions, the AI completely fail to play the game. Unlike humans, who understand the core mechanics of the game and can adjust to superficial changes, the AI had memorized pixel patterns and reward functions without developing an abstract understanding of the game's rules.

Mitchell has more recently argued that the limitations faced by these models comes from their lack of an internal world model, meaning a structured representation of concepts, objects, relationships, and causality essential for genuine reasoning and understanding [47]. Without such a model, Mitchell asserts, AI systems cannot reliably generalize knowledge to new contexts or apply common-sense reasoning effectively.

A similar Discussion is made by Gary Marcus [42], which emphatizes the need for robust models that move away from narrow task solving to more

generalized and structured knowledge. He showed similar examples for these limitations, suggesting that Hybrid-AI and neuro-symbolic approaches are the next step forward.

Another point raised by Bender et all. [7] is the latest trend of increasing model and dataset sizes to improve "intelligence". They argue that expanding model parameters and data volume merely improves benchmark scores by better capturing superficial patterns rather than genuinely enhancing deeper cognitive capabilities. Moreover, larger models and datasets incur substantial environmental and financial costs.

Nonetheless, some experts argue a contrasting view: larger models could eventually develop internal world models or conceptual understanding if supplemented with novel architectural changes, explicit grounding, or improved training techniques. Yann Lecun's position [20], for example, is that we should focus on using self-supervised learning to guide the emergence of an internal world model in deep learning solutions. In this regard, recently some studies seems to point out the emergence of preliminary forms of conceptual coherence in latest architectures, but there is still no proof of actual understanding in AI.

In conclusion, despite impressive progress, current AI models remain limited in genuine understanding and reasoning capabilities. Future developments in AI should critically empathize the generalization capabilities of AI systems and the development of an internal world model to understand context, meaning and causality.

3.1.2 The Black Box Problem

AI's limitations extend beyond its lack of understanding. One is the intrinsic black-box nature of deep learning. Architectures become more complex and powerful, but their decision-making processes remain difficult to interpret, raising concerns about transparency, accountability, and fairness.

The lack of explainability is a well-known problem that hinder the application of AI in different contexts. If an AI system denies a loan, misdiagnoses a patient, or recommends a prison sentence, stakeholders need to understand why.

Gary Marcus has been particularly vocal about this issue [41], arguing

that AI's reliance on statistical pattern recognition without structured reasoning makes it inherently unreliable. He contends that AI cannot be truly trustworthy without interpretable decision-making mechanisms, advocating for hybrid models that incorporate both neural learning and explicit symbolic reasoning to ensure more transparent and verifiable outputs.

Bias is a related problem; it can be defined as a systematic error that reflect or amplify inequalities found in training data. One real-world example comes from hiring algorithms trained on corporate datasets, these have been found to favor male candidates over women, reinforcing gender biases present in historical hiring practices.

There are many types of biases [18] and researchers try to take them into account when creating training datasets, but this is made difficult by the scale of current datasets. They emerge because AI does not question or correct the patterns it learns, it optimizes for accuracy based on existing data, regardless of whether that data is biased. This connects with the previous section, current AI simply replicates patterns without an understanding of context, fairness, or moral considerations. The consequences of this can be severe: biased AI systems risk amplifying societal inequalities, embedding discrimination into automated decision-making at scale.

While remaining a problem even for human cognition, some biases in artificial intelligence could be mitigated with more explainable models by helping demarcating where and how they emerge [57].

Bias and opacity of deep learning models represent a limitation that involves both ethic and practical challenges in AI today. As these systems are increasingly deployed in governance, healthcare, hiring, and finance, their impact on society cannot be left unchecked. This translate to the need for a more transparent decision making process, ones that can explain what factors influenced the result.

3.1.3 How Do We Measure Intelligence?

Despite not being a limitation of machines themselves but rather a limitation on our part, a still open problem is how to define and measure intelligence. While AI has demonstrated extraordinary capabilities in specific domains, its evaluation is largely based on narrow, task-specific benchmarks rather than a comprehensive understanding of intelligence itself. This raises the question: are we measuring intelligence, or just an AI's ability to optimize for a specific

test?

Traditional AI benchmarks, such as ImageNet for computer vision, GLUE for natural language processing, and various reinforcement learning test suites, have played a crucial role in driving AI advancements. However, these benchmarks do not assess intelligence in a generalizable way. Instead, they encourage task-specific optimization, where AI models become exceptionally good at one predefined challenge but struggle to adapt beyond it. This issue was central to François Chollet's argument in *On the Measure of Intelligence* [10], where he criticized conventional AI evaluation methods for rewarding brute-force computation and memorization rather than adaptive reasoning. Chollet proposed that a more meaningful test of intelligence should focus on an agent's ability to generalize efficiently from minimal experience, a skill that is central to human cognition but mostly absent in AI. The proposed alternative is the Abstraction and Reasoning Corpus (ARC) [11]. This framework emphasizes an agent's ability to solve novel problems using limited prior information, reflecting human-like generalization from sparse data. Rather than rewarding scale and data exposure, Chollet's method prioritizes efficiency, flexibility, and the reuse of abstract knowledge.

The debate over what constitutes intelligence, and on how it should be assessed, remains one of the most fundamental open questions in AI research and beyond. Even in psychology, intelligence remains a multifaceted and debated concept. IQ tests, for example, capture some aspects of human intelligence, such as pattern recognition and logical reasoning, but fail to measure other essential components, including creativity, emotional intelligence, and social intuition. If we struggle to quantify intelligence in humans, it is no surprise that measuring intelligence in machines remains an elusive challenge.

Shane Legg and Marcus Hutter, in their foundational work on universal artificial intelligence [34], attempted to formalize intelligence as an agent's ability to succeed across a broad range of environments. Their definition suggests that intelligence is not about mastering a single domain but about adapting efficiently to many different kinds of problems, particularly those that were not explicitly trained for; this is similar to Chollet's definition and also aligns with Alan Turing's original perspective, which stated that intelligence is not about what a system knows but what it can do when faced with new and unpredictable challenges.

The lack of a universal way to measure intelligence remains one of the

biggest obstacles to understanding whether AI is truly advancing. While AI has become remarkably proficient at specific tasks, it remains unclear how much progress has been made toward intelligence in a broader sense. As AI systems grow more complex, the need for better, more meaningful evaluation metrics will become even more urgent. Without them, we may struggle to distinguish between machines that are truly intelligent and those that are simply very good at playing the test.

3.2 Exploring New Frontiers: Research Directions Toward True AI Learning

As discussed in the previous section, modern AI systems remain fundamentally limited in their ability to learn and reason in a way that mirrors human intelligence. Although deep learning models can recognize patterns and make predictions, they struggle with abstraction, have difficulty adapting to new information without extensive retraining, and lack a deeper understanding of the world. This raises fundamental questions: What enables humans to learn efficiently? Can AI develop innate cognitive structures rather than rely solely on massive datasets? How can AI move beyond correlation-based learning toward genuine reasoning and long-term knowledge retention?

Leading AI researchers have proposed different paths to address these limitations. Yann LeCun [20] argues that AI should move toward self-supervised learning, allowing models to construct hierarchical world representations through prediction and abstraction, much like how humans learn from experience. Instead of relying solely on labeled data, self-supervised models could form structured understandings of the world by continuously refining their own internal representations. In contrast, Gary Marcus [42] critiques the current deep learning paradigm as fundamentally flawed, emphasizing the need for explicit cognitive structures and innate mechanisms, much like those seen in human intelligence; he advocates for a neuro-symbolic approach, in which deep learning is supplemented with rule-based reasoning and structured knowledge representations. Similarly, Melanie Mitchell [46] argues that true AI intelligence requires mechanisms for analogy-making and abstraction, skills that are central to human reasoning but largely absent from current AI systems.

These perspectives highlight the need for AI to go beyond statistical

pattern-matching toward more structured, adaptable, and explainable reasoning processes. This section explores key research directions that attempt to bridge these gaps.

3.2.1 Core Knowledge: Learning from First Principles

As discussed in the previous chapter, the debate between empiricism and rationalism finds a common ground in saying that living beings have some innate form of knowledge (in this context, it isn't important how this innate knowledge came into being) and that they use those first concepts to create new ones that are refined through experience. This perfectly align with the current perspective in epistemology, describing innate structures that have evolved throughout human history and that guide the formation of knowledge.

Elizabeth Spelke, in her studies on the early cognitive development of infants [70], coined the term Core Knowledge to represent innate systems that allow newborns to develop advanced cognitive structures to navigate and understand their environment. She recognized five core knowledge systems in both human and non-human infants, each guiding a different understanding:

- Objects and their interactions (Object Representation)
- Agents and goal-directed actions
- Number and mathematical reasoning (number sense)
- Space and spatial navigation (intuitive geometry)
- Social relationships and interactions

The research in Core Knowledge Theory is ongoing and more systems have been proposed.

Through these foundations, infants are able, for example, to perceive object boundaries, shapes and expected behaviors. In early state of life, they do not possess cognitive systems for representing and reasoning about specific objects such as foods or artifacts, but they will eventually form those using their Core Knowledge and their experience of the environment.

The idea of equipping AI systems with core knowledge priors has gained attention as a promising path toward human-like generalization and robustness, motivating researchers to explore whether the incorporation of core

knowledge could serve as a foundation for more efficient learning. For instance, Battaglia et al. [5] introduced Graph Networks, which use structured relational representations to reflect human-like reasoning about object interactions. These networks are designed to incorporate relational inductive biases—mirroring the way humans understand the physical world through core object representations—and have been successfully used to improve generalization in physical reasoning tasks.

The goal is not to encode all possible knowledge into machines, but to start from proven inductive biases, just as humans do. By grounding AI in these conceptual primitives, we may bridge the gap between low-level perception and high-level reasoning, fostering systems that learn more like humans: efficiently, flexibly, and with minimal supervision.

3.2.2 Embracing Uncertainty: Causality in AI Learning

An essential question in AI is whether current systems truly comprehend why events happen. Without understanding causality, models cannot capture the mechanisms that drive change. Differentiating correlation from causation is critical, as it allows AI to reason about interventions, anticipate outcomes, and generalize beyond learned experiences. Without this capability, AI systems risk making unreliable or even harmful decisions when facing unfamiliar scenarios.

Traditional deep learning focuses on statistical patterns, which can lead to misleading conclusions. Without the ability to reason causally, AI systems are limited to passive observation. Consider an AI trained to recommend treatments based on medical records: it may observe that patients receiving a certain drug tend to recover and infer the drug is effective. However, it might overlook confounding factors, such as healthier patients being more likely to receive that drug. True understanding comes from recognizing whether the treatment itself caused recovery or if another variable explains both. This distinction is crucial for safe and effective decision-making.

Furthermore, consider an AI analyzing loan approvals: it might find that applicants from certain neighborhoods are more often denied loans and infer risk based on location. However, without counterfactual reasoning, it cannot ask, “Would this person have been approved if they lived elsewhere?” This kind of reasoning is essential to avoid encoding systemic biases as causal

facts. Without it, decisions may perpetuate unfair patterns under the guise of data-driven logic.

Causal understanding also supports generalization. Imagine an AI trained to detect manufacturing defects in one factory using visual cues correlated with faults. When deployed in a different factory with different lighting or materials, the same cues may no longer apply. However, if the AI had learned the causal factors behind the defects—such as stress points or process irregularities—it could transfer that knowledge and adapt effectively to the new setting.

The concept of causality has evolved through centuries of philosophical and scientific thought. Aristotle distinguished four types of causes—material, formal, efficient, and final, laying an early framework for understanding causation in nature. In the Islamic Golden Age, philosophers like Avicenna debated the nature of cause and effect, asserting necessary links between causes and outcomes. In early modern Europe, Francis Bacon promoted empirical investigation and inductive reasoning, emphasizing observation over speculation, though without a formal causal theory. David Hume later challenged the very notion of necessary causation, arguing that causality is inferred from habit, not logical necessity, a view that sparked enduring debate. John Stuart Mill responded with systematic approaches, such as the Method of Agreement and the Method of Difference, to empirically investigate causal relationships. These foundational ideas set the stage for later formalizations, including Judea Pearl's Structural Causal Models, a formal framework for representing and reasoning about causality using directed graphs and do-calculus [54].

Various methodologies have been proposed in attempts to introduce causal reasoning into AI, aiming to move beyond surface-level pattern recognition and toward systems that can potentially act on and explain cause-effect relationships. Some of these efforts include:

- `Causal Discovery Algorithms` - These algorithms attempt to infer causal relationships from raw observational data. Methods include:
  - Granger Causality [22] – Determines if one time series predicts another, commonly used in economics.
  - PC Algorithm [71] – Uses conditional independence tests to construct causal graphs.

- `Causal Reinforcement Learning` [24] - Traditional reinforcement learning optimizes actions based on trial and error, but lacks an understanding of why actions lead to rewards. Causal RL integrates structural causal models into decision-making, allowing agents to reason about the effects of their actions before executing them.

- `Causal Generative Models` [32] - Standard deep learning models are trained to generate data based on statistical distributions. Causal generative models, such as CausalGAN, explicitly model causal dependencies, allowing for better controllability and interpretability. Other, such as Causal-enhanced neural networks, integrate symbolic causal reasoning with deep learning, enabling AI to perform structured interventions rather than simply generating data based on past patterns.

- `Causal Representation Learning` [66] - Unlike standard machine learning, which learns feature correlations, causal representation learning seeks to discover meaningful causal variables from raw data. This is essential in domains like healthcare, where AI must identify the true causes of diseases rather than just recognizing statistical associations.

Challenges remain: causal discovery often requires interventions, many algorithms do not scale well, and integrating causality with deep learning remains complex. Yet, the benefits, more robust, adaptive, and trustworthy AI, make this a vital direction for future research.

To ground future progress in AI, it's useful to distill some guiding ideas. Core principles of causal reasoning [54] include:

- Correlation does not imply causation – Just because two variables co-occur frequently does not mean that one causes the other.

- Interventions change the world - AI should be able to reason about the consequences of its actions (e.g., what happens if I press this button?).

- Counterfactuals matter – AI should be able to imagine alternative scenarios (e.g., what if this patient had taken a different medication?).

- Generalization depends on causality – Learning causal relationships allows AI to transfer knowledge to unseen situations, rather than memorizing patterns that only apply in specific contexts.

3.2.3 Learning That Lasts: Memory-Augmented Intelligence

Traditional artificial intelligence models inherently lack explicit and persistent memory mechanisms. Instead, they rely mainly on internal parameters (i.e., weights) updated during training, implicitly encoding past data experiences. Some models employ limited short-term memory through recurrent architectures like Recurrent Neural Networks or Long Short-Term Memory networks. However, these implicit memories do not function as explicit storage systems, meaning that they cannot selectively recall specific information and they also suffer from limited capacity and difficulties in retaining long-term context.

The limitations of conventional architectures make a compelling case for integrating explicit and persistent memory integrations, structures that more closely resemble the human cognitive model of memory, capable of flexible storage, retrieval, and manipulation of knowledge across time.

Several fields within artificial intelligence would have clear benefits from robust, explicit memory augmentation:

- Natural Language Processing (NLP): Effective conversation agents and text comprehension tasks depend heavily on context management and on factual information. Retrieval Augmented Generation improves large language models' factual accuracy by incorporating information retrieval before generating responses, but the knowledge base is external and can't be updated by the models. On the other hand, Context windows, used in NLP models to maintain temporal consistency and reason on sequential inputs, can function like a sort of short working memory, but these models still lack a a long-term memory making their underlying transformer architectures struggle with long input sequences and sequential tasks.

- Reinforcement Learning: Two major ways in which an explicit long-term memory augmentations would benefit reinforced learning are context based decision making, using information such as states and past actions, and episodic memory reasoning, using stored past experiences to structure and understand new information or to infer the context and adapt to it.

- Continual Learning: Many systems are trained over specific tasks and

usually, if the task is changed, they need to go through retraining, which mean to completely forget previously derived knowledge and train from scratch, or fine-tuning, which doesn't assure that previous knowledge is maintained (i.e., catastrophic forgetting) when new information are encoded. Memory augmentation that allow for selective and explicit retrieval and modification of stored information would be a great advance for most AI models.

- Complex Reasoning and Algorithmic Tasks: As reasoning and more symbolic AI models emerge, there is the need for external memory in order to store intermediate states and algorithmic logic simulation.

Recent advances in Memory-Augmented Neural Networks (MANNs) [65] propose a reorientation of AI architectures, treating memory not as a byproduct of training, but as a core computational resource. These systems partly draw inspiration from psychological theories of memory, such as the Atkinson-Shiffrin model [2], to create machines capable of flexible storage, selective recall, and context-sensitive processing. While there have been contrasting theories, this model is considered a good foundation in studying how human memory works. In this context, memory can be seen as broadly divided into working (short-term) memory and long-term memory, each serving distinct functions:

- `Working Memory` - Temporarily holds information needed for immediate reasoning and task execution. It is capacity-limited and designed for rapid retrieval and manipulation of information relevant to ongoing activities. Working memory is still obscure despite the extensive researches, but one regarded theory is the Baddeley-Hitch model [3], which describe it as a multi-component system:
  - `Central Executive` - Responsible for managing attention and coordinating the activities of the other subsystems. The central executive is believed to be involved in higher-level cognitive functions, such as problem-solving, planning, and reasoning.
  - `Phonological Loop` - Subordinate system that deals with auditory and verbal information and prevents their decay by continuously refreshing it in a rehearsal loop.
  - `Visuo-spatial Sketchpad` - Subordinate system that manage visual and spatial information. Its tasks seems to be related to spatial

understanding and to construction and manipulation of visual images and mental maps, dealing with such phenomena as shape, color, and texture.

  - `Episodic Buffer` - Latest addition to the Baddeley-Hitch model, it is believed to be the subordinate system in communication with long-term memory and to bind phonological, visual, spatial, and semantic information into unitary episodic representation for the central executive to work with.

- `Long-term Memory` - Stores extensive amounts of information over long periods, with virtually unlimited capacity. The current understanding is that the long-term memory is divided in:

  - `Declarative` (Explicit) `Memory` - Conscious and intentional recollection of experiences, facts and concepts. These information can be retrieved explicitly, with different amount of effort based on their strength and complexity. Acquisition, consolidation, and retrieval are the key processes. Declarative memory is further divided in episodic memory, which stores actual past experiences along with records of sensory-perceptual-conceptual-affective processing, and semantic memory, which refer to general impersonal world knowledge built upon past experiences. Semantic memory might contain information about what a pizza is, while episodic memory might contain a specific memory of me enjoying a pizza.

  - `Procedural` (Implicit) `Memory` - Unconscious part of memory that focus on implicit learning skills through repetition. When one individual get better in a task only due to repetition, it is shown that no new explicit memories form, instead unconscious procedural memories emerge. These memories are accessed and used without the need for conscious control or attention. Most motor skills are stored in this part of memory.

Studying these theories, several memory augmentation solutions have been researched to address these needs, including:

- `Retrieval Augmented Generation` [36] - As mentioned above, it is an integration that allow generative models to query an external knowledge base and to add the retrieved sources to the inputs before generating the response. It is a way to allows models to use domain-specific and

access updated information. Unfortunately, these sources are static to the model and they have to be reviewed accurately to not incur in misinformation.

- `Hopfield Network` [27] - A form of recurrent neural network that differ from traditional RNN and as an auto-associative content-addressable memory. It consist of a single layer where each neuron is connected to all the others with a bidirectional weight and where the outputs are fed back into its inputs. Patterns are associatively recalled by fixing certain inputs, and dynamically evolve the network to minimize an energy function towards local minimum that correspond to stored patterns.

- `Neural Turing Machine`s [23] - An extension of the concept of Turing Machine, with a NN controller interacting through attention-based mechanisms with an external modifiable memory that emulate the infinite tape in Turing's vision. This architecture uses an attentional process to read from and write to memory selectively and reason through "rapidly-created variables".

- `Memory-Augmented Transformers` [12] - Frameworks to enhance transformers' ability to process long-context, inspired by human memory processes.

Without memory, AI remains reactive, bound to fixed context windows and forgetful of the past. MANNs promise systems that are not only more efficient but more general, capable of evolving knowledge, grounding their inferences, and building models of the world that persist beyond a single input or task. Ultimately, memory is not an add-on. It is foundational to intelligent behavior. By drawing from cognitive science and computational theory, memory-augmented architectures lay the groundwork for AI systems that reason, remember, and adapt like humans do.

3.2.4 Making Sense of AI: Toward Explainability

Attempts to improve AI explainability have led to the rise of explainable AI (XAI) initiatives. These techniques can be broadly categorized [1] into two main types:

- `Intrinsic interpretability`, where the model is inherently understandable due to its structure. This includes decision trees, linear models, and rule-based systems, where the reasoning behind predictions is explicit.

- `Post-hoc explanations`, which apply explainability methods to complex models (e.g., deep learning and ensemble methods) after training, aiming to explain their internal workings without modifying their architecture.

On the side of Post-Hoc Explanations, researchers are developing techniques to obtain interpretations of what influenced the results of deep learning models:

- `Feature Attribution Methods`, such as SHAP (i.e., Shapley Additive Explanations) [38] and LIME (i.e., Local Interpretable Model-Agnostic Explanations) [59], which identify the most influential features that contribute to the prediction of a model. These methods help uncover biases, improve debugging, and improve user trust.

- `Attention-Based and Saliency Methods`, commonly used in computer vision and NLP, which highlight parts of the input that most influenced the model's decision. For example, in an image classification task, saliency maps can reveal which regions of an image led to a certain prediction.

- `Surrogate models`, where a simpler, interpretable model (e.g., a decision tree or linear regression) is trained to approximate the predictions of a more complex model. This technique is useful for understanding the general behavior of black-box models, but their fidelity to the original model can be limited in highly nonlinear decision spaces.

- `Concept-Based Explanations`, which attempt to align AI reasoning with human-understandable concepts rather than low-level features. Techniques like TCAV (i.e., Testing with Concept Activation Vectors) [30] help to understand which high-level concepts (e.g., "stripes" in an image classifier) influence more the model output.

However, explainability remains an open challenge. Many deep learning models, especially those in NLP and computer vision, are so complex and

nonlinear that even their creators struggle to fully understand how they arrive at specific outputs. This has led some experts to argue that rather than trying to interpret inherently opaque models, AI research should focus on building inherently interpretable architectures that align with human reasoning from the outset. Another aspect to consider is that explainability is often a trade-off with model complexity and capabilities. Highly explainable models (e.g., decision trees) may not always achieve the same level of accuracy and flexibility as deep learning architectures. Researchers are actively working on methods to balance accuracy with explainability, aiming for AI systems that are powerful and understandable.

The goal of Explainable AI is to ensure that humans can trust and collaborate effectively with AI systems, making them more accountable, fair, and aligned with human reasoning; possibly with the same capabilities.

3.2.5 Neurosymbolic Integration: Bridging Logic and Learning

Traditional AI has been shaped by two dominant paradigms: symbolic AI, which relies on explicit rules and logic for structured reasoning, and neural networks, which extract patterns from raw data without predefined structures. While symbolic AI offers explainability and formal reasoning, it struggles with large-scale unstructured data. Neural networks, on the other hand, excel at recognizing patterns in high-dimensional spaces but often function as opaque "black boxes" stuck in their learned distribution.

Neurosymbolic AI seeks to combine the strengths of both approaches, integrating the flexibility of neural learning with the precision and explainability of symbolic reasoning. By embedding structured logic into learning architectures, or conversely, enabling neural networks to manipulate symbols, these hybrid models aim to achieve deeper reasoning, better generalization, and more transparent decision-making.

Neurosymbolic AI has shown promise in different domains, particularly in areas where both reasoning and learning are required. Key researches include:

- `Differentiable Inductive Logic Programming` (dILP) [16] - Traditional Inductive Logic Programming learns symbolic rules from structured data by searching through a combinatorial space of possible logical

hypotheses. This makes it computationally expensive and hard to scale. dILP replaces this discrete search process with differentiable operations, enabling gradient-based optimization. Logical rules are represented as weighted constraints, allowing the system to softly select the best-fitting rules rather than exhaustively searching for all possibilities. The advantage of dILP is that it can learn first-order logic rules from raw data while remaining interpretable, making it useful for reasoning tasks that require structured decision-making.

- `Neural Theorem Provers` (NTPs) [60] - In classical theorem proving, symbolic reasoning engines apply rigid inference rules to derive conclusions from a knowledge base. NTPs replace these rigid operations with neural embeddings, where logical predicates and terms are mapped into a continuous vector space. Instead of explicitly unifying terms via hard matching, NTPs use differentiable unification, allowing for approximate matching of terms in learned latent spaces. This makes NTPs more robust to noisy or incomplete data and enables AI to perform logical reasoning over natural language statements and knowledge graphs without requiring strict symbolic representations.

- `Neural-Symbolic Knowledge Graphs` [39] - Knowledge graphs store relational information in symbolic form (e.g., entities and edges), but they suffer from incompleteness and lack of adaptivity. Neural-symbolic knowledge graphs integrate Graph Neural Networks or transformer-based encoders to learn embeddings that preserve logical structure while enabling inference over missing or uncertain relationships. These models allow AI to extend symbolic knowledge graphs with learned relational patterns, making them suitable for applications where structured and unstructured reasoning must be combined.

- `Neuro-Symbolic Concept Learners` [40] - which learns symbolic concepts from raw perceptual input by separating neural feature extraction from logical reasoning. It first maps input data (e.g., images or text) into latent representations, then uses a probabilistic logic programming layer to induce symbolic concepts from those representations. The key innovation is its ability to perform compositional reasoning, which means that it can generalize beyond training data by combining learned concepts in novel ways. This architecture is particularly useful for vision-and-language models that require explainability and structured understanding.

Despite these advances, neurosymbolic AI remains an evolving field with significant challenges. Scaling these systems to real-world complexity requires refining architectures that seamlessly combine learning and reasoning while maintaining efficiency. Current research focuses on improving symbolic abstraction learning, reducing computational overhead, and developing models that can autonomously acquire and manipulate structured knowledge without relying on predefined symbolic representations.

The long-term goal is to create AI that learns, reasons, and adapts in a way that mirrors human intelligence, capable of not just recognizing patterns, but understanding their meaning and applying logical inference to new problems.

Chapter 4

Proposed Framework

The goal of this chapter is to present the framework developed throughout this thesis, both in its conceptual evolution and in its current implementation. The proposed system aims to infer symbolic and interpretable knowledge about dynamic environments, specifically within the context of Atari-like video games, without relying on deep learning or opaque statistical methods.

Rather than attempting to replicate state-of-the-art performance in game playing, the framework focuses on exploring a cognitive-inspired alternative for building internal world models. The motivation behind this work lies in addressing key limitations of current AI systems, such as the lack of generalization, interpretability, and adaptability to environmental changes.

4.1 Conceptual Foundations

This section provides some insights in the evolution of the themes and the objectives of this thesis, from the analysis of the short-comings of the starting idea to the definition of core design principles that guided the rest of the work. It also includes a brief glossary of terms used in the rest of the text.

4.1.1 Evolution of the Idea

The initial concept for this work originated from a draft of a research paper proposing an evolutionary approach to structured environment modeling, inspired by principles from Core Knowledge theory. The aim of that work

was to obtain human-interpretable representations of dynamic environments, such as simple 2D video games, by combining low-level visual processing with evolutionary learning mechanisms. The underlying hypothesis was that, by evolving rule-based systems guided by heuristics related to innate cognitive capacities, it would be possible to build a symbolic and structured description of the environment, with minimal reliance on data-driven learning.

The evolutionary component was designed to optimize the symbolic representation of the environment by selecting compact sets of object classes and behavioral rules. The encoded heuristics provided the scoring system for this process, while the evolutionary algorithm acted as a search procedure to compress and refine these structures into a minimal explanatory model.

However, a deeper analysis of the system's behavior revealed that the evolutionary component, despite contributing to the exploration of candidate solutions, ultimately hindered progress toward generalization. Since rules and class assignments were evolved and then tested against fixed sequences of patches, the resulting representations, though plausible within each scenario, lacked the flexibility to adapt to new or unseen sequences. This contradicted the broader goal of constructing an internal world model capable of accumulating and reusing knowledge across environments. In contrast, a framework aimed at modeling structured, generalizable knowledge should prioritize the incremental construction of representations, building on previously inferred behaviors in a coherent and extensible way.

Following this shift, the design of the framework moved toward a knowledge-construction approach based on incremental and structured understanding. Rather than relying on global optimization over static data, the goal became to emphasize a step-by-step process, in which knowledge is progressively built through symbolic structures as new behaviors and interactions are observed.

The idea evolved in the direction of a bottom-up perceptual base, anchored in core knowledge, combined with a top-down guidance mechanism, where previously inferred knowledge informs the interpretation of new observations. In this model, symbolic structures such as object classes and behavioral rules are not generated in isolation, but emerge from the interaction between incoming perceptual data and the system's existing internal model.

The current implementation of the framework follow an heuristic approach and serve as a testbed for this approach. Although fragile and limited in

scope, it successfully produced a symbolic representation of the game environment from low-level visual patches, providing a concrete proof of concept for the proposed direction.

The final phase of this work focused on identifying a set of conceptual expansions intended to bridge the gap between the initial implementation and the broader framework envisioned. These proposals aim to refine the modules employed in the first implementation and to add top-down knowledge guidance to the pipeline. While still in an early stage, they draw from recent developments in both cognitive science and AI, and represent a step toward a more complete and cognitively inspired system.

4.1.2 Core Design Principles

Parallel to the framework design, the review of literature on cognitive science helped refine the design principles of the system, with a focus on core knowledge, explainability and generalization.

Core Knowledge

Core Knowledge theory posits that human cognition is grounded in a small number of domain-specific systems. These systems are considered to be evolutionarily derived and to serve as foundational structures for learning and reasoning. Based on extensive research in developmental psychology and comparative cognition, four well-established core systems have been identified: representation of inanimate objects and their physical interactions, representation of agents and goal-directed actions, numerical cognition, and geometric reasoning about spatial layouts. A fifth system, concerning the identification of social partners and group membership, has also been proposed.

Among these, the system for object representation is the most extensively documented and most directly relevant to the present framework. This system provides a minimal set of spatiotemporal principles that allow agents to perceive discrete objects within a continuous stream of sensory input. These principles include:

- **Cohesion**: Objects move as connected and bounded wholes.
- **Continuity**: Objects follow continuous, unobstructed paths.

- **Contact**: Objects interact through direct contact rather than at a distance.

In addition, the system supports basic concepts about shape and boundaries. These capabilities allow even very young infants, and many non-human animals, to segment scenes into meaningful units, track entities across frames, and form basic predictions about their motion and interactions.

In the context of this thesis, Core Knowledge is employed as a source of design principles. It is used to define a minimal, cognitively inspired set of perceptual constraints that guide the system's low-level interpretation of visual input. Specifically, the principles derived from Core Knowledge should serve two purposes: first, to define the building blocks on which knowledge is constructed; second, to provide the symbolic scaffolding upon which new information, such as behavioral rules and class abstractions, can be structured.

Practically, these principles are encoded as heuristic mechanisms in the processes of the framework. Visual patches are grouped and tracked over time according to cohesion, continuity, and contact, forming the basis for object persistence. As the system observes interactions, it begins to infer regularities and structured symbolic representations. In this way, Core Knowledge acts as both a structural prior and a generative constraint: it shapes the form of knowledge that can be acquired, while remaining flexible enough to support open-ended development.

Explainability

Explainability has emerged as a central concern in the development of artificial intelligence systems, particularly as machine learning models have grown in complexity and opacity. While modern neural networks can achieve remarkable performance, their decision processes are inscrutable. This lack of transparency not only undermines trust, but also hinders debugging, evaluation, and integration with other systems. In response, the field of explainable AI (XAI) has proposed a range of strategies to enhance explainability.

Recent literature suggests that a promising direction lies in designing intrinsic interpretability across multiple levels of abstraction: from encoded priors to internal knowledge representation and up to the modeling of the external environment. Such a layered approach promotes transparency not

only in decision outcomes, but also in the underlying mechanisms that generate them.

The framework presented in this thesis is structured to support explainability at every level. Symbolic principles inspired by Core Knowledge serve as explicitly defined priors, guiding the perception process and constraining the form of the information inferred. All internal representations are encoded in human-readable structures, with processing steps that are discrete, traceable, and transparent.

By committing to symbolic and modular representations, the system deliberately trades off some statistical flexibility in favor of interpretability, debuggability, and alignment with human conceptual understanding. This trade-off reflects the core motivation of the framework: to explore how structured knowledge can emerge from perception in a way that is not only functional, but intelligible.

In this context, any future neuro-symbolic integration should adhere to the same design principle. While neural components may eventually replace or enhance specific modules, they should do so without compromising step-by-step transparency or the inspectability of the internal reasoning process.

Generalization and Adaptability

The ability to generalize and adapt to novel situations is a hallmark of intelligent behavior. In both biological and artificial systems, the capacity to extend knowledge beyond specific training instances is essential for robust performance in dynamic environments. Despite major advances in statistical learning, many state-of-the-art models remain highly sensitive to distributional shifts, superficial perturbations, or minor variations in input. This brittleness seems to stem from a fundamental limitation: the absence of an internal, structured model of the world that supports flexible reasoning and learning.

Generalization in humans appears to be grounded in abstract representations that capture regularities across experience. These representations are not tied to specific observations, but are organized around symbolic relations, causal patterns, and conceptual structures. From this perspective, adaptability is not simply the ability to re-train quickly, but the ability to reuse and restructure prior knowledge to make sense of new situations with minimal data.

The framework proposed in this thesis is designed to move toward this type of generalization, using internal representations that are symbolic, reusable, and adaptable. These structures are intended to be transferable across similar environments, enabling the system to interpret new scenarios without starting from scratch. The goal is not to maximize generalization through statistical generality, but to promote it through structured abstraction.

At its current stage, generalization and adaptability are still limited by the immaturity of the core knowledge base. However, the framework is conceived with the intent to support cumulative knowledge construction, allowing the system to adapt by reorganizing its internal model in response to new evidence.

In summary, generalization and adaptability in this framework are guiding design principles. They inform the choice of symbolic representations and guide the development of knowledge structures. The aim is to approach a form of intelligence that grows through the accumulation and transformation of structured knowledge, involving contextual and analogical reasoning.

4.1.3 Terminology

- `Frame` - In this context, frames are intended as visual screenshots of the game environment.
- `Property` - Predefined as a core knowledge, it includes a name and a method to compute its value.
- `Patch` - A visually coherent and distinct element extracted from a frame and associated with computed properties. It is anonymous, representing the state in which a non-yet-specified object appear at a given time.
- `Event` - Predefined as a core knowledge, it includes a name, the list of involved properties and a method to test if it gets triggered.
- `Unexpected change` - Each change in property happening from subsequent patches associated to an object. The goal is to have them all explained by rules.
- `Object` - Reconstruction of a sequence of Patches that is considered by the system to have consistency and continuity. Events and unexpected changes are computed considering it as a true object, with a behavior to be explained.

- `Rule` - Symbolic and causal rule that describe a behavior of an object. They are defined by a cause, a delay, an effect and some context-inferred parameters.

- `Class` - Contain the rules and a generalization of the expected properties' behavior. Objects assigned to a class have to conform.

- `Individual` - It is a possible explanation of the processed frames, it includes classes and objects. Together, they form a possible understanding of the environment, with differences between each individual.

4.2 Proposed Implementation

This section presents the current state of the proposed framework, developed following the core design principles introduced above. While still limited in scope and functionality, this version propose a method that allow a symbolic system to extract interpretable representations from a dynamic environment using only low-level perceptual input.

The development focuses on a simplified scenario: a 2D game environment which evolves frame by frame. These frames are processed through a pipeline consisting of object tracking, rule inference and class generalization. The system uses explicit heuristics inspired by the principles of Core Knowledge to generate symbolic knowledge about object identity and behavior, which are then used as guidance in the following iterations.

Although the framework is still evolving and presents several areas for improvement, the results obtained from the preliminary implementation are promising: it demonstrates the ability to generate a structured and interpretable internal model of the environment, correctly identify persistent objects, infer interactions, and recognize primitive classes. This first iteration is not intended as a final solution, but as a minimal and interpretable foundation upon which more robust, generalizable, and adaptive components can be developed.

The following sections describe the key modules of this framework, focusing on their purpose, design, and current limitations.

4.2.1 Patch Extraction

The environment that the system have to comprehend is represented as a sequence of discrete frames extracted from a game environment. These frames have to be processed to obtain a set of non-overlapping visual units referred to as "patches". These patches constitute the system's primary perceptual input: anonymous minimal representations of visual elements that may correspond to objects or object parts within the environment.

Rather than relying on actual visual processing or segmentation algorithms, this initial version of the system operates on patches that are directly extracted from a simulated environment. The simulation provides access to information such as position and shape, which is used to construct simplified patch representations without performing any form of visual recognition. As a result, the system bypasses the challenges of real visual perception, such as noise, occlusion, or ambiguity. This choice reflects the early focus of the research, which prioritized symbolic modeling and knowledge representation over perception. However, as the framework evolved and greater emphasis was placed on cognitive plausibility, the limitations of this approach became more evident.

The current patch extraction strategy serves as a placeholder: it allows for rapid experimentation and controlled testing of higher-level components, but does not align with the long-term vision of perceptual grounding.

Future iterations of the framework are expected to include a visual segmentation module, in which patches are derived from raw pixel input under the guidance of learned knowledge. This step is essential for testing the robustness of the system under realistic perceptual conditions and for further developing the framework's knowledge construction capabilities.

4.2.2 Object Formation

The object tracking module is responsible for reconstructing the identity and continuity of objects over time based on sequences of visual patches. Since the system receives no direct information about object permanence or identity across frames, this component plays a critical role in forming persistent representations of entities within the environment.

In this context, object tracking is performed using a set of interpretable heuristics relying on predictions made using previously inferred knowledge

and on the concepts of spatial and temporal coherence encoded in the core knowledge base. Each object is associated with an object class, which defines a set of behavioral rules. By applying these rules in combination with the object's current properties (such as position, velocity, or state), the system can predict the object's future state in the environment. Each new patch is compared to existing objects based on proximity, shape, and consistency with expected behavior. When a sufficiently close match is found, the object's identity is updated to include the new patch. Specifically, patch-to-object assignments fall into the following categories:

- `Perfect Assignment` - The object predicted the patch
- `Possible Assignment` - A patch and an object are assigned even if they do not perfectly match, defining an unexpected change in the object
- `No Direct Assignment` - An object or a patches has no association, including cases such as completely new object or the reappearance of a disappeared object

Because the system analyzes frames sequentially, ambiguities frequently arise in both patch assignments and preliminary interpretation of changes (distinct from causal explanations handled during rule inference). For example, a change in position may indicate simple movement, or it could suggest an acceleration that will become clearer over subsequent frames. Similarly, the sudden appearance of a patch near another disappearing object raises competing interpretations about continuity, occlusion, or replacement.

To manage such ambiguity, the system can maintain multiple competing interpretations in parallel. These are represented as distinct, non-conflicting configurations of objects and associated behaviors, each differing in assignments, inferred events, or higher-level abstractions. This approach supports flexibility without prematurely committing to a single explanatory path.

Each object is modeled as an episodic reconstruction, specific to the current sequence of observations. It encapsulates a set of assigned patches, associated properties, observed events, and unexplained changes. Within this framework, generalizable knowledge arises through object classes and behavioral rules, while individual objects serve as their episodic instantiations.

In summary, the object formation process serves as the foundational layer for symbolic understanding within the system. By combining heuristic tracking with predictive reasoning and structured ambiguity management, the

system reconstructs coherent, temporally persistent entities from raw perceptual input. These episodic object representations not only enable localized interpretation of events but also ground the abstraction of generalizable knowledge through object classes and behavioral rules, bridging perception and conceptual understanding.

4.2.3 Rule Inference

The rule inference module is responsible for identifying and abstracting general symbolic rules that describe dynamic relationships between events and changes in objects. Its function is to detect regularities and to represent them in a form that can be reused, interpreted, and eventually generalized across environments.

Rule generation operates on the data accumulated during object tracking. As objects evolve across frames, they accumulate both observed events and unexplained changes in their properties. While events are nominal, unexplained changes can involve numerical parameters and require generalization.

When a consistent co-occurrence is detected between one or more events and a change in an object's properties, a symbolic rule is generated. These rules take the form of generalized expressions that relate symbolic events to variable transformations. For example, if an object reverses its direction of motion immediately after a contact, the system may infer a rule such as:

```
Contact_Right -0-> vx[i+1] = -1 * vx[i] + 0
```

In this way, the rule abstracts from the individual instance to a structural regularity.

The generated rules are not tied to a specific episode, but are intended to populate a growing knowledge base of general behavioral patterns. Rules are stored symbolically and may later serve as priors or constraints in the interpretation of new situations.

However, the current implementation relies on correlational analysis, and does not yet reason about causality or counterfactuals. For example, it cannot distinguish between a correlation due to co-occurrence and one based on causal necessity. Addressing this limitation is part of the framework's envisioned development.

In summary, rule inference in the current framework is an initial step

toward structured, reusable symbolic knowledge. It enables the system to detect and represent regularities in the environment in a transparent and generalizable form, while setting the stage for future integration of more sophisticated causal reasoning methods.

4.2.4 Pruning

In the previous phases, the system was able to maintain multiple individuals, each representing a different possible explanation of the environment. However, since this process is intended to iterate through all frames and, eventually, episodes, it must account for the fact that the number of individuals could grow uncontrollably. Keeping all of them alive in parallel would quickly become computationally unfeasible. For this reason, in the current phase, individuals are evaluated based on the internal consistency of their explanations, enabling the system to discard the least coherent interpretations.

Multiple factors are considered when computing a score for each individual. The first is the number of unexplained changes recorded in objects. Since object-patch assignments may differ across individuals, the resulting unexplained changes will also vary. Minimizing these changes corresponds to finding the simplest and most coherent object-patch associations—for example, a ball moving at constant speed is better explained by a single change in velocity than by multiple unexplained position shifts.

The second factor assesses how well each individual adheres to object classes and their behavioral rules. This includes evaluating whether property constraints are respected and how many unexplained changes are correctly predicted by the associated rules.

The final factor considers the overall complexity of the explanation, penalizing individuals that rely on a higher number of distinct rules or object classes. This again favors simpler interpretations of the environment.

All individuals that share the highest score are preserved, while the others are marked for removal. However, this elimination does not occur immediately. Instead, individuals are granted a grace period during which they can potentially provide better explanations—particularly in cases where confirmation requires observing subsequent frames.

Overall, the design of this scoring system aims to balance the exploration of

competing interpretations with the computational cost of maintaining them, allowing the system to remain both flexible and efficient.

4.2.5 Generalization

A fundamental objective is the ability to generalize across diverse game environments without extensive retraining. Generalization, in the context of our framework, refers to the capacity of the structured symbolic representations to adapt seamlessly to slight or significant variations in game dynamics. This contrasts with discussed limitations in deep reinforced learning approaches, which often exhibit brittleness, losing performance even under minimal visual or mechanical changes.

In this context, generalization is achieved through the creation of classes that abstract objects from their episodic existence. Classes are thought as flexible blueprints encoding behavioral patterns of objects and of their properties, defining relations that should be stable in related environments. Specifically, information such as a property always being constant or the fact that all the objects assigned to a class have the same shape can be used to find similar objects in other environments and to help the evolution of their understanding.

In this context, generalization is achieved through the formation of object classes that abstract objects from their episodic existence. These classes act as flexible blueprints, capturing both structural properties and recurring behavioral patterns observed across instances. Simple regularities, such as a property remaining constant or objects consistently reacting in the same way can help the system to recognize similar entities in new environments and to refine its understanding as new observations accumulate.

As previously mentioned, object classes are used to guide object formation by enabling predictions about an object's future state. This allows learned classes to be carried over into new episodes, supporting the interpretation of unfamiliar scenarios. In such cases, objects are not assigned to a class immediately, but may exhibit behaviors that align with existing classes over time. Once a consistent match is found, the association can be established, and any unobserved behaviors can be inferred from the class definition.

The system must be capable of adapting its object classes as new behaviors are observed or as previously coherent patterns begin to diverge. When certain instances consistently violate existing rules or introduce unexplained

changes, this may indicate the need to revise the class, either by refining its rules to incorporate the new behavior or by splitting it into more specific classes to preserve internal consistency. These adjustments allow the system to maintain robust and accurate generalizations, evolving its symbolic structures without discarding previously acquired knowledge.

This mechanisms allow the system to maintain adaptability while preserving interpretability, enabling knowledge gained in one scenario to inform understanding in another. Rather than relearning from scratch, the system can incrementally refine its internal model by comparing new observations to prior symbolic structures. This supports a form of transfer learning that is grounded in explainable components, object classes and their associated rules, rather than opaque statistical features.

Chapter 5

Experimental Evaluation

This section presents a preliminary implementation of the proposed framework, alongside a set of initial experiments designed to assess its effectiveness in supporting explainability, structured representation, and generalization across similar environments.

The experimental setup focuses on a simplified version of the Atari game Arkanoid, selected for its deterministic physics and discrete, interpretable interactions. This controlled environment provides an ideal testbed for observing how the system forms and maintains object representations, infers behavioral rules, and organizes entities into meaningful classes.

The analysis is primarily qualitative, emphasizing the internal consistency and symbolic coherence of the system's outputs. Preliminary observations are also made regarding the model's ability to retain and apply learned structures when exposed to slight variations of the original environment, offering an early indication of its potential for generalization without retraining.

5.1 Code Structure

The project is implemented in python and is organized into modular components. The `core/` folder contains the encoded core knowledge, such as events, properties, and unexplained changes, as well as the structures that guide the formation of knowledge, including rules, objects, classes, and individuals. The `heuristic/` module implements the flow for the evolution of the population. The `arkanoid/` folder provides some simplified implementations of

the game Arkanoid, which are used to generate sequences of frames for the system to process. Utility scripts support debugging with manually crafted patches and handle various auxiliary tasks. The script `main.py` serves as the entry point for executing knowledge extraction.

5.1.1 Core Files

This module defines the key classes and abstractions used throughout the project. Each file typically defines one or more classes that encapsulate specific concepts:

- `property.py`: Defines the Property parent class and some encoded properties (`Pos_x`, `Pos_y`, `Shape_x`, `Shape_y`, `Speed_x`, and `Speed_y`).
- `patch.py`: Defines the Patch class, which represents a visually distinct anonymous element in a frame, capturing properties at a given time step.
- `unexplained.py`: Contains the Unexplained class, which describes internal events occurring within objects, such as property changes or disappearances.
- `events.py`: Contains the Event parent class and some hierarchical encoded events, such as global events, and Contact and its directional variations.
- `rule.py`: Implements the logic to define and test abstract rules that describe changes in object evaluating unexpected changes and events.
- `object.py`: Contains the Object class, representing an entity composed of a sequence of patches across time. It stores properties, events, unexplained changes, global events, and the sequence of assigned patches.
- `class.py`: Defines classes that generalize objects and their behaviors. Each class includes set of rules and a variance dictionary that maps properties to their expected variability.
- `individual.py`: Manages specific explanations, consisting of a set of non-overlapping objects and selected interpretations. This module is mainly relevant for output generation, as individual-level processing is handled in `heuristic.py` in this implementation.

5.1.2 Execution Flow

To provide robust patch sequences for testing the framework, a simplified environment representing the game Arkanoid is implemented in the `arkanoid_complete.py` file. This script records the log of each game session, including the anonymous elements present in the environment, and saves it in the `logs/` directory for later processing.

To generate a new episode, the user runs `arkanoid_complete.py` and plays a game. Upon completion, a log file is automatically saved, containing all relevant elements required by the framework.

The rest of the pipeline is launched from `main.py`, which first handles the conversion of game logs into patch sequences and global events. It then invokes the heuristic method and displays the resulting inferences.

Log extraction is straightforward, as the files already contain the necessary information. The corresponding extraction method is implemented in `utils/need_imports.py`.

Once patch sequences and global events have been obtained, the script `heuristic_initialization.py` is executed.

heuristic_initialization is responsible for the initialization and frame-by-frame evolution of the population based on the input data. Its main steps are outlined below:

- **Population Initialization** - Tthe function creates a new population by assigning one object per patch in the first frame. Each object encapsulates the patch's properties and any associated global events.
  - If previously inferred objects classes are available, the function resumes from that state and use them to guide the identification of objects.
- **Frame-by-Frame Processing** — For each new frame, a series of steps is executed:
  - *Predicted Assignment of Patches to Objects:* Existing objects attempt to predict incoming patches based on property values (e.g., position, shape, or speed) and rules. Matches result in direct assignments and object updates.

- *Evaluation of Object Unexpected Changes:* Heuristics assess whether non-assigned objects exhibit unexplained changes such as motion, duplication, disappearance, or sudden reappearance. This involves:
  - `check_for_speed` and `check_for_property0_changes`, which detect variations in velocity or primary properties and formulate hypothesis.
  - detection of other unexpected changes, including Appearance, disappearance, and overlapping).

  Different views are tested creating new individuals.
- *Rule Inference:* New unexplained changes and events are used to verify and update rules regarding objects.
- *Population Pruning:* Individuals are periodically scored based on the number of unexplained events, the consistency of property evolution, and the quality of inferred rules. Poorly performing individuals are removed to maintain focus on the most plausible hypotheses.
- *Prototype Summarization:* The function `summarize_into_prototypes` groups similar objects based on rule signatures and property variance. These clusters are abstracted into `Class` objects that capture shared behavioral patterns across multiple instances.

- **Conversion to Individuals** After all frames are processed, the remaining object groups are converted into instances of the `Individual` class. Each individual represents a complete interpretation of the input sequence, along with a confidence score.

5.1.3 Result Structure

The results are saved in `.txt` format and contain, in order of score, a list of individuals (one if a clear explanation has been found), along with the corresponding list of objects and their associated rules. The rules are presented in a human-readable format, as is the composition of each object. Additionally, ground-truth labels are included to verify whether the system correctly grouped patches into coherent objects. Each individual also includes a summary in terms of object classes.

5.2 Experiments

In this section, some tests made with this preliminary implementations are described, comparing the representation inferred by the system with the rules encoded in the environment.

As mentioned, the system receives frames of anonymous patches from the environment, only containing low-level properties. One episode is intended as the full sequence of frames that are extracted from the same game, representing the sequential evolution of a playthrough. The environments' initial ball position is randomized, to encounter different scenarios. Currently, three environments have been tested:

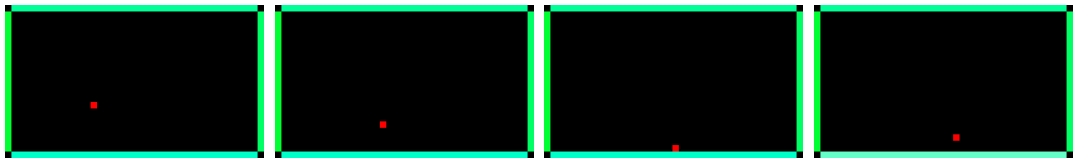
- `arkanoid_simple` — basic example, with four walls and a ball
- `arkanoid_complete` — implementation of the arkanoid game, with four walls, a ball, a paddle and 24 bricks
- `arkanoid_complete_modified` — branch of arkanoid_complete to which some modifications are made on-demand, such as bigger bricks or a quicker ball

Each of these environments saves a log containing elements and global events. These logs can be processed as whole episodes or sequentially frame by frame.

To assess the goodness of the associations, the environments associate to each patch a name which represent the object it belonged to in the scene. These are not used during association and inference, but are printed in the results to evaluate the representation.

the first experiment is made on arkanoid_simple.

The environment doesn't have external interactions (i.e., the paddle), so it is left to run and then stopped after a while, producing a log consisting of 804 frames.



(a) frame 2 (b) frame 11 (c) frame 22 (d) frame 27

Figure 5.1: images from the arkanoid_simple environment

The log is passed as a whole to the system, which converge into one individual, describing five objects and inferring six rules for the object which represent the ball. This representation is then generalized to the following object classes:

```
Class 0:
Property Variance:
        Pos_x: constant
        Pos_y: constant
        Shape_x: constant
        Shape_y: constant
Rules:
        No Rules
same_shape: False
Assigned Objects: [0, 1, 2, 3]
-----
Class 1:
Property Variance:
        Pos_x: variable
        Pos_y: variable
        Shape_x: constant
        Shape_y: constant
        Speed_x: variable
        Speed_y: variable
Rules:
        game_start -0-> vx[i+1] = 0 * vx[i] + 1
        game_start -0-> vy[i+1] = 0 * vy[i] + 1
        Contact_Bottom -0-> vy[i+1] = -1 * vy[i] + 0
        Contact_Right -0-> vx[i+1] = -1 * vx[i] + 0
        Contact_Top -0-> vy[i+1] = -1 * vy[i] + 0
        Contact_Left -0-> vx[i+1] = -1 * vx[i] + 0
same_shape: True
Assigned Objects: [23]
```

The behaviors encoded in this simple environment are comprehended by the system, which understand that:

- the ball start moving at the start of the game
- the ball invert its horizontal speed when making contact on the right or on the left
- the ball invert its vertical speed when making contact on the top-side or on the bottom-side

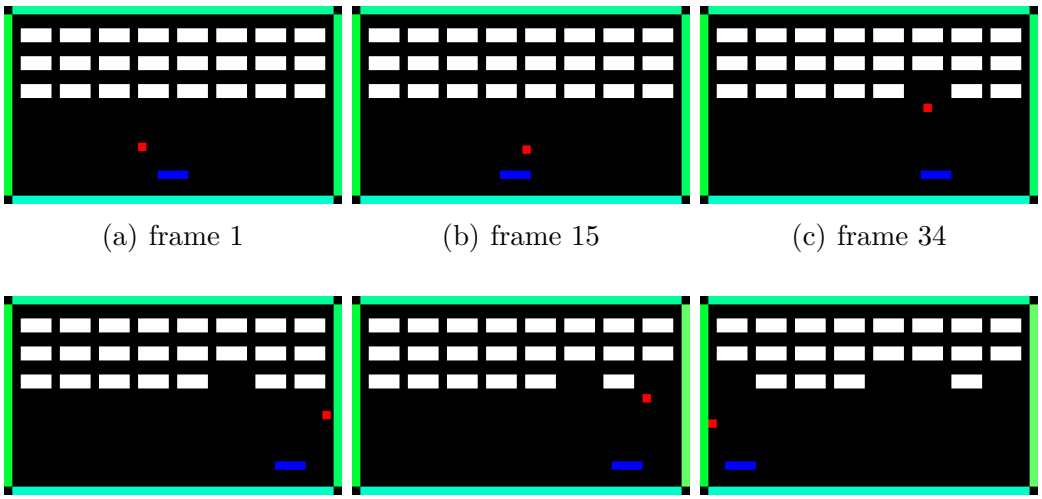
The second experiment involves the arkanoid_complete environment.

This time the game can actually be played by moving the paddle. The recorded log is of a lose game with 498 frames.

The system converge to one individual, describing 30 objects and inferring six rules for the object which represent the ball, two rules for the paddle and one for the bricks. The generalization of the representation is the same for the ball and for the walls, while two more object classes are generated:

```
Class 1:
Property Variance:
        Pos_x: variable
        Pos_y: constant
        Shape_x: constant
        Shape_y: constant
Rules:
        left_arrow_pressed -1-> pos_x[i+1] = 1 * pos_x - 2
        right_arrow_pressed -1-> pos_x[i+1] = 1 * pos_x + 2
same_shape: True
Assigned Objects: [28]
-----
Class 3:
Property Variance:
        Pos_x: constant
        Pos_y: constant
        Shape_x: constant
        Shape_y: constant
Rules:
        Contact -2-> Disappearance
same_shape: True
Assigned Objects: [21, 27, 18, 6, 12, 24, 20, 9, 11, 15, 8, 4,
5, 7, 10, 13, 14, 16, 17, 19, 22, 23, 25, 26]
```

Notably, if left_arrow_pressed and right_arrow_pressed are not recorded in the logs, the representation is the same, but with no rules for the object representing the paddle. This means that the system tries to infer what it can, but that some behaviors need by nature more information to be inferred; a system that does not possess information about the state of the left key cannot infer the fact that it is the reason for which the paddle moves to the left or not. At the same time, these rules are important, as they could guide an agent that uses this representation as an internal model to make decisions based on the predicted evolution of the environment after his choice of using left_arrow_pressed.



(a) frame 1 (b) frame 15 (c) frame 34

(d) frame 70 (e) frame 80 (f) frame 181

Figure 5.2: Images from the arkanoid_complete environment showing key events

To show the evolution of the Object Classes the third experiment uses a log of arkanoid_complete stopping the processing of frames based on the events happening on screen. The log is of a win in 1501 frames, shown in the images.

At frame 0, there is one individuals that describes 30 objects (one for each patch) and one object class with no specialization.

```
Class 0:
Property Variance:
        Pos_x: constant
        Pos_y: constant
        Shape_x: constant
        Shape_y: constant
Rules:
        No Rules
same_shape: False
Assigned Objects: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]
```

At frame 1, the ball start moving and the system is uncertain; there are four individuals, diverging in how to explain the change in the object representing the ball, if with a move or with a change in speed (for both axes).

At frame 2, the ball continues its motion; this means an higher score for the individual that correctly predicted this behavior. The individuals are still four, but one is stronger and the others live their grace period (1 frame in this experiment).

At frame 3, the ball continues its motion. This time only one Individual is left, describing 30 objects and two object classes, one representing the ball and one not specialized.

```
Class 0:
Property Variance:
        Pos_x: constant
        Pos_y: constant
        Shape_x: constant
        Shape_y: constant
Rules:
        No Rules
same_shape: False
Assigned Objects: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]
-----
Class 1:
Property Variance:
        Pos_x: variable
        Pos_y: variable
        Shape_x: constant
        Shape_y: constant
        Speed_x: variable
        Speed_y: variable
Rules:
        game_start -0-> vx[i+1] = 0 * vx[i] + 1
        game_start -0-> vy[i+1] = 0 * vy[i] + 1
same_shape: True
Assigned Objects: [30]
```

At frame 9, the paddle has moved and a new object class has been formed:

```
Class 1:
Property Variance:
        Pos_x: variable
        Pos_y: constant
        Shape_x: constant
        Shape_y: constant
Rules:
        left_arrow_pressed -1-> pos_x[i+1] = 1 * pos_x - 2
same_shape: True
Assigned Objects: [34]
```

At frame 15, the ball has bounced off the paddle and a new rule has been added:

```
Contact_Bottom -0-> vy[i+1] = -1 * vy[i] + 0
```

At frame 34, the ball has bounced off a brick, which disappeared. One new rule is generated for the disappeared brick.

```
Contact_Bottom -1-> Disappearance
```

The other bricks are still in the not specialized object class.

One the other hand, one rule for the ball is modified, making a wrong assumption. As it has seen two contacts resulting in the same outcome, it generalized them into one rule.

```
Contact -0-> vy[i+1] = -1 * vy[i] + 0
```

At frame 70, the ball has bounced off the right wall. This make for a change in the rules assigned to the object representing the ball.

```
Contact_Bottom -0-> vy[i+1] = -1 * vy[i] + 0
Contact_Right -0-> vx[i+1] = -1 * vx[i] + 0
Contact_Top -0-> vy[i+1] = -1 * vy[i] + 0
```

At frame 80, another brick has disappeared. Now, all the bricks are put inside the same prototype, but still with the rule

```
Contact_Bottom -1-> Disappearance
```

At frame 182, the ball has made contact with the left wall and a new rule for the ball has been added.

```
Contact_Left -0-> vx[i+1] = -1 * vx[i] + 0
```

At frame 348, the ball hit a brick from the left and the system generalize the rule for bricks to

```
Contact -1-> Disappearance
```

From this point, the objects' prediction match the new patches and the system representation is stable.

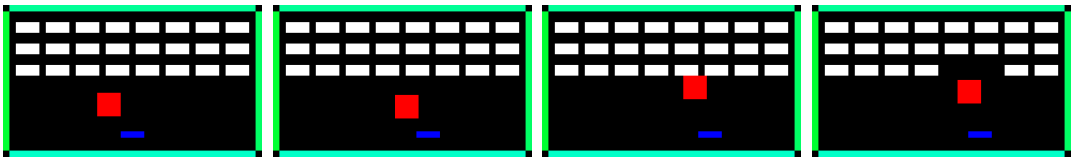
Another scenario is tested crafting an intangible brick. The ball first overlap behind the brick, then it briefly disappear from view, to reappear overlapping on the other side and continuing its movement. The heuristics are still immature, but a rule

```
Contact -0-> Overlap
```

is created, which stops the system from assuming that the ball disappeared when behind the brick. This kind of behavior could benefit from more complex rules and should be carefully studied when implementing the perception part of the framework, possibly using top-down information to infer the existence of patches hidden by other elements.

Using the arkanoid_complete_modified environment, some experiments are made to test the generalizability of the representation found by the system.

First, the ball size is fivefold. The log of the game contains 186 frame. Notably, with the ball this big more than one brick is destroyed at the same time most of the times. The representation found is the same as the environment with no changes, which is good, since the knowledge obtained from the original could be used to infer the evolution of the modified version.



(a) frame 2 (b) frame 15 (c) frame 24 (d) frame 26

Figure 5.3: images from the arkanoid_complete_modified environment, with a ball fivefold its original size

Then, another experiment involve modifying the ball speed, specifically only its horizontal component. The result is similar, with only the rule on starting speed getting modified to

```
game_start -0-> vx[i+1] = 0 * vx[i] + 2
```

This is good because the rules describing the rest of the behaviors are unchanged, but it emphasize the need for context-aware parameters in the rules, so that they can be adapted to various scenarios.

To test a case for which knowledge cannot be inferred, arkanoid_complete_modified is changed so to make the ball vary its speed almost randomly after each contact. In this case, the system finds the other rules as before, but cannot infer meaningful behavior for the ball bounces

```
Class 2:
Property Variance:
        Pos_x: variable
        Pos_y: variable
        Shape_x: constant
        Shape_y: constant
        Speed_x: variable
        Speed_y: variable
Rules:
        game_start -0-> vx[i+1] = 0 * vx[i] + 1
        game_start -0-> vy[i+1] = 0 * vy[i] + 1
same_shape: True
Assigned Objects: [30]
```

Going further, some preliminary test are made to transfer learning to new eipsodes.

First the object classes are formed on arkanoid_simple and used to guide the understanding of arkanoid_complete. The same two logs as before are used to make a comparison; the representation follow a similar evolution, but the ball is associated to its object class at frame 3 and it remains stable for the rest of the episode. In contrast, before, 182 frames had to be processed before obtaining a stable representation.

Using an episode from arkanoid_complete as base show similar results, with the system stabilizing at frame 34 (i.e., when the first brick disappear), instead of frame 348 as before. Similarly, the addition or subtraction of bricks doesn't change the understanding.

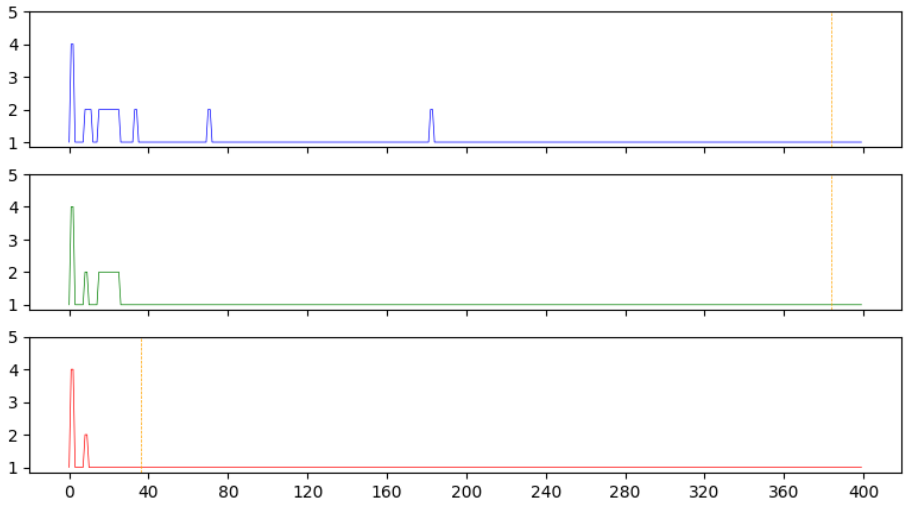


Figure 5.4: Top: population evolution on arkanoid_complete starting from scratch; Middle: population evolution drawing from arkanoid_simple; Bottom: population evolution drawing from arkanoid_complete; the orange lines mark where the representation got stable

By tweaking arkanoid_complete_modified, a test scenario is created with two different sizes for bricks. In this case, the first group of bricks is assigned to the object class representing a brick after one is destroyed at frame 40; the rest after a brick of the second group is destroyed at frame 110. Differently, the usage of different sizes for all bricks confuses the system, which doesn't have common ground to associate them, making so that the bricks are assigned only when being destroyed.

Another test is made passing from arkanoid_complete to the version of arkanoid_complete_modified with changed horizontal speed. At first the system cannot assign the ball to the correct object class, creating a new one instead.

```
Class 2:
Property Variance:
        Pos_x: variable
        Pos_y: variable
        Shape_x: constant
        Shape_y: constant
        Speed_x: variable
        Speed_y: variable
Rules:
        game_start -0-> vx[i+1] = 0 * vx[i] + 2
        game_start -0-> vy[i+1] = 0 * vy[i] + 1
same_shape: True
Assigned Objects: [30]
```

Then, after bouncing off the paddle, the system completes the class with the information from the original.

```
Class 2:
Property Variance:
        Pos_x: variable
        Pos_y: variable
        Shape_x: constant
        Shape_y: constant
        Speed_x: variable
        Speed_y: variable
Rules:
        game_start -0-> vx[i+1] = 0 * vx[i] + 2
        game_start -0-> vy[i+1] = 0 * vy[i] + 1
        Contact_Bottom -0-> vy[i+1] = -1 * vy[i] + 0
        Contact_Right -0-> vx[i+1] = -1 * vx[i] + 0
        Contact_Top -0-> vy[i+1] = -1 * vy[i] + 0
        Contact_Left -0-> vx[i+1] = -1 * vx[i] + 0
same_shape: True
Assigned Objects: [30]
```

These tests only show preliminary results, as the system clearly needs a more complex knowledge base and the integration of methods to facilitate generalization between episodes, but they still works as a proof of concept for the proposed framework. Specifically, the results shows how the system require few interactions to make assumptions and how these can be changed when faced with different scenarios.

Chapter 6

Conclusion and Future Work

This thesis has investigated critical limitations in contemporary artificial intelligence approaches, specifically their brittleness, lack of reasoning, and inherent opacity. It discussed how LLMs doesn't use language in a meaningful way and how current deep reinforcement learning models fail to develop transferable, structured knowledge; with both problems highlighting their reliance on fragile, statistical pattern-matching methods rather than genuine causal understanding or conceptual reasoning.

Drawing from a interdisciplinary background, including philosophy, cognitive science, and AI research, this thesis identified a growing consensus around the need to move beyond purely sub-symbolic approaches. Scholars like Melanie Mitchell and Gary Marcus have emphasized the importance of combining structured, symbolic reasoning with neural methods to build more robust, interpretable systems.

In response, this work introduced the foundations for a framework inspired by Core Knowledge Theory. The proposed system constructs structured representations of dynamic environments by identifying entities through fundamental properties and tracking their behaviors symbolically. By applying abstract classes and rules, it shows potential for generalizing across related environments, directly addressing the limitations of current methods.

Preliminary results suggest that this structured approach improves interpretability, robustness, and explainability in dynamic contexts. However,

challenges remain: the current implementation still lacks context-based reasoning, operates with an immature knowledge base, and risks computational explosion. Furthermore, integration of causal reasoning remains an open research question.

Future work will explore solutions to these limitations, some of which are proposed below.

6.1 Future Directions

The current framework demonstrates the potential for constructing symbolic, interpretable models of dynamic environments by grounding object tracking and rule inference in core knowledge. While initial results show promise in generalizing behavior and recognizing recurring structures across similar game instances, several areas remain open for further development. These proposed expansions aim to extend the framework's capabilities, robustness, and applicability.

6.1.1 Perception Module

A first critical addition concerns the introduction of an explicit perception module into the framework. The inclusion of a semantic segmentation mechanism, guided by information derived from the predicted state of known objects, would enable the system to extract anonymous patches from raw frames. Once an object has been defined and its behavior modeled, its predicted future state in the environment can be used to direct attention mechanisms during the segmentation process, focusing computation on specific regions of the screen where the object is expected to be. This integration of perceptual input with top-down conceptual priors would significantly improve the applicability and learning capability of the system.

6.1.2 Memory Integrations

Another key expansion involves the integration of a memory system. In its current form, the framework stores rules and object classes without a clear structure. This represents a significant limitation, particularly when learning

across multiple episodes or adapting to dynamic environments. This memory system should encompass both semantic and episodic components. While the semantic memory can function as a structured repository for inferred object classes and generalized rules, the episodic memory could be designed to store detailed information about specific and significant sequences of frames. This would require the implementation of mechanisms for identifying patterns that merit retention and efficiently retrieving past experiences based on contextual similarity. The addition of such a memory structure would open new possibilities for analogical and counterfactual reasoning, allowing the system to compare current situations with prior experiences and, eventually, to simulate alternative outcomes. More broadly, it would support the progressive refinement of knowledge over time, enabling the system to evolve a coherent and context-sensitive understanding that persists across episodes.

6.1.3 Environmental States

A further conceptual enhancement involves modeling not just dynamics but also states of the environment. Many real-world scenarios rely on hidden or latent states to modulate object behavior. For example, a door might be "locked" or a ball might be "energized". These are not directly visible in the frame but critically shape how objects interact. To account for this, the framework should support episode-specific states: abstract variables that can be modified by rules and, in turn, trigger rule conditions. These variables would be initialized either explicitly or inferred from object behavior and would evolve over time as consequences of interactions. Crucially, they would allow rules to have memory across frames and context-dependence, making it possible, for instance, to define that a collision with a power-up changes the behavior of the ball in subsequent frames. This mid-level abstraction between visual input and symbolic rules is essential for modeling complex, state-dependent dynamics.

6.1.4 RL Integration

Another extension involves the integration of the framework into a Dyna-style reinforcement learning architecture, a direction that has already begun to be explored. The core idea is to employ the symbolic internal representation developed by the system as a predictive engine to guide the agent's internal simulation phase. In a traditional Dyna architecture, the agent

maintains a model of the environment and uses it to simulate trajectories, evaluating the outcomes of possible actions before interacting with the actual environment. In this context, the structured representation inferred by the framework would serve as that internal simulator. As the agent observes the environment, the incoming frames are encoded into states, capturing all currently recognized objects and their properties. When the agent performs an action, it is interpreted as an event within the representation, triggering the relevant symbolic rules. These rules update the internal state accordingly, simulating the expected evolution of the environment based on learned object behaviors and interactions.

This simulation process could be iterated, allowing the agent to explore hypothetical future states entirely within its internal world model. In doing so, the symbolic framework functions not only as a planning mechanism but also as an evaluative tool. When discrepancies arise between predicted outcomes and actual observations, they reveal limitations or inaccuracies in the internal model, prompting further refinement. In this sense, the symbolic representation becomes both an engine for decision-making and a testable hypothesis about the environment's dynamics, one that evolves as the agent gains experience.

Bibliography

[1] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)". In: *IEEE Access* 6 (2018), pp. 52138–52160.

[2] Richard C. Atkinson and Richard M. Shiffrin. "Human memory: A proposed system and its control processes". In: *The Psychology of Learning and Motivation*. Ed. by Kenneth W. Spence and Janet Taylor Spence. Vol. 2. Academic Press, 1968, pp. 89–195.

[3] Alan D. Baddeley and Graham Hitch. "Working memory". In: *The Psychology of Learning and Motivation*. Ed. by Gordon H. Bower. Vol. 8. Academic Press, 1974, pp. 47–89.

[4] Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. *LLMs Will Always Hallucinate, and We Need to Live With This*. 2024. arXiv: 2409.05746 [stat.ML]. URL: https://arxiv.org/abs/2409.05746.

[5] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018). URL: https://arxiv.org/abs/1806.01261.

[6] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[7] Emily M. Bender et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 610–623. ISBN: 9781450383097. DOI: 10.1145/3442188.3445922. URL: https://doi.org/10.1145/3442188.3445922.

[8] Murray Campbell, A. Joseph Hoane Jr., and Feng-hsiung Hsu. "Deep Blue". In: *Artificial Intelligence* 134.1-2 (2002), pp. 57–83. DOI: 10.1016/S0004-3702(01)00129-1.

[9] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2020. URL: https://arxiv.org/abs/2002.05709.

[10] François Chollet. *On the Measure of Intelligence*. 2019. arXiv: 1911.01547 [cs.AI]. URL: https://arxiv.org/abs/1911.01547.

[11] François Chollet. "The Abstraction and Reasoning Corpus". In: *arXiv preprint arXiv:1911.01547* (2019).

[12] Zihang Dai et al. *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*. 2019. arXiv: 1901.02860 [cs.LG]. URL: https://arxiv.org/abs/1901.02860.

[13] Stanislas Dehaene. "The Number Sense: How the Mind Creates Mathematics". In: *British Journal of Educational Studies* 47.2 (1999), pp. 201–203.

[14] Grégoire Delétang et al. *Neural Networks and the Chomsky Hierarchy*. 2023. arXiv: 2207.02098 [cs.LG]. URL: https://arxiv.org/abs/2207.02098.

[15] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* (2018). URL: https://arxiv.org/abs/1810.04805.

[16] Richard Evans and Edward Grefenstette. "Learning Explainable Logical Rules with Neural Logic Programming". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018.

[17] Jiajun Fan. *A Review for Deep Reinforcement Learning in Atari:Benchmarks, Challenges, and Solutions*. 2023. arXiv: 2112.04145 [cs.AI]. URL: https://arxiv.org/abs/2112.04145.

[18] Emilio Ferrara. "Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies". In: *Sci* 6.1 (Dec. 2023), p. 3. ISSN: 2413-4155. DOI: 10.3390/sci6010003. URL: http://dx.doi.org/10.3390/sci6010003.

[19] Kunihiko Fukushima. "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". In: *Biological Cybernetics* 36 (1980), pp. 193–202. DOI: 10.1007/BF00344251.

[20] Quentin Garrido et al. *Intuitive physics understanding emerges from self-supervised pretraining on natural videos*. 2025. arXiv: 2502.11831 [cs.CV]. URL: https://arxiv.org/abs/2502.11831.

[21] Ian Goodfellow et al. "Generative Adversarial Networks". In: *arXiv preprint arXiv:1406.2661* (2014). URL: https://arxiv.org/abs/1406.2661.

[22] Clive W. J. Granger. "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica: Journal of the Econometric Society* 37.3 (1969), pp. 424–438.

[23] Alex Graves, Greg Wayne, and Ivo Danihelka. "Neural Turing Machines". In: *arXiv preprint arXiv:1410.5401* (2014).

[24] Juan Camilo Gutiérrez, Adarsh Subbaswamy, and Elias Bareinboim. "Causal Reinforcement Learning using Observational and Interventional Data". In: *Proceedings of the 39th International Conference on Machine Learning (ICML)*. 2022, pp. 7957–7970.

[25] Udesh Habaraduwa. *Inductive Models for Artificial Intelligence Systems are Insufficient without Good Explanations*. 2024. arXiv: 2401.09011 [cs.LG]. URL: https://arxiv.org/abs/2401.09011.

[26] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 9729–9738. DOI: 10.1109/CVPR42600.2020.00975. URL: https://arxiv.org/abs/1911.05722.

[27] John J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities". In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558.

[28] Lei Huang et al. "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions". In: *ACM Transactions on Information Systems* 43.2 (Jan. 2025), pp. 1–55. ISSN: 1558-2868. DOI: 10.1145/3703155. URL: http://dx.doi.org/10.1145/3703155.

[29] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[30] Been Kim et al. "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)". In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*. 2018, pp. 2668–2677.

[31] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *arXiv preprint arXiv:1312.6114* (2013). URL: https://arxiv.org/abs/1312.6114.

[32] Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. "CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training". In: *International Conference on Learning Representations (ICLR)*. 2018.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25 (NIPS 2012)*. Curran Associates, Inc., 2012, pp. 1097–1105. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[34] Shane Legg and Marcus Hutter. "Universal Intelligence: A Definition of Machine Intelligence". In: *CoRR* abs/0712.3329 (2007). arXiv: 0712.3329. URL: http://arxiv.org/abs/0712.3329.

[35] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. "The Winograd Schema Challenge". In: *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*. 2012. URL: https://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4492.

[36] Patrick Lewis et al. "Retrieval-augmented generation for knowledge-intensive NLP tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[37] Seppo Linnainmaa. "The Representation of the Cumulative Rounding Error of an Algorithm as a Taylor Expansion of the Local Rounding Errors". Master's Thesis. MA thesis. University of Helsinki, 1970.

[38] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[39] Robin Manhaeve et al. "DeepProbLog: Neural Probabilistic Logic Programming". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2018.

[40] Jiayuan Mao et al. "Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences from Natural Supervision". In: *International Conference on Learning Representations (ICLR)*. 2019.

[42] Gary Marcus. *The Next Decade in AI: Four Steps Towards Robust Artificial Intelligence*. 2020. arXiv: 2002.06177 [cs.AI]. URL: https://arxiv.org/abs/2002.06177.

[43] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Chapter 1, Section 1.2. San Francisco: W. H. Freeman, 1982. ISBN: 978-0716712848.

[44] Warren S. McCulloch and Walter Pitts. "A Logical Calculus of the Ideas Immanent in Nervous Activity". In: *The Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133. DOI: 10.1007/BF02478259.

[45] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969. ISBN: 978-0262130431.

[46] M. Mitchell. *Artificial Intelligence: A Guide for Thinking Humans*. Farrar, Straus and Giroux, 2019. ISBN: 9780374715236. URL: https://books.google.it/books?id=65iEDwAAQBAJ.

[48] Melanie Mitchell and David C. Krakauer. "The debate over understanding in AI's large language models". In: *Proceedings of the National Academy of Sciences* 120.13 (2023). ISSN: 1091-6490. DOI: 10.1073/pnas.2215907120. URL: http://dx.doi.org/10.1073/pnas.2215907120.

[49] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: 1312.5602 [cs.LG]. URL: https://arxiv.org/abs/1312.5602.

[52] Allen Newell, J.C. Shaw, and Herbert A. Simon. "Report on a General Problem-Solving Program". In: *Proceedings of the International Conference on Information Processing*. 1959, pp. 256–264.

[54] Judea Pearl. *Causality: Models, Reasoning and Inference*. 2nd. Cambridge University Press, 2009.

[55] Xinghua Qu et al. *Minimalistic Attacks: How Little it Takes to Fool a Deep Reinforcement Learning Policy*. 2020. arXiv: 1911.03849 [cs.LG]. URL: https://arxiv.org/abs/1911.03849.

[56] Alec Radford et al. "Improving Language Understanding by Generative Pre-Training". In: *OpenAI Blog* (2018). URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[58] Aditya Ramesh et al. "Zero-Shot Text-to-Image Generation". In: *arXiv preprint arXiv:2102.12092* (2021). URL: https://arxiv.org/abs/2102.12092.

[59] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2016, pp. 1135–1144.

[60] Tim Rocktäschel and Sebastian Riedel. "End-to-end Differentiable Proving". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017.

[61] Robin Rombach et al. "High-Resolution Image Synthesis with Latent Diffusion Models". In: *arXiv preprint arXiv:2112.10752* (2021). URL: https://arxiv.org/abs/2112.10752.

[62] Frank Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Report No. VG-1196-G-8, Cornell Aeronautical Laboratory. Washington, DC: Spartan Books, 1962.

[63] Frank Rosenblatt. "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain". In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: 10.1037/h0042519.

[64] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Nature* 323 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[65] Adam Santoro et al. "Meta-learning with memory-augmented neural networks". In: *International conference on machine learning (ICML)*. PMLR. 2016, pp. 1842–1850.

[66] Bernhard Schölkopf et al. "Toward causal representation learning". In: *Proceedings of the IEEE* 109.5 (2021), pp. 612–634.

[67] Edward H. Shortliffe et al. "Computer-Based Consultations in Clinical Therapeutics: Explanation and Rule Acquisition Capabilities of the MYCIN System". In: *Computers and Biomedical Research* 8.4 (1975), pp. 303–320. DOI: 10.1016/0010-4809(75)90009-9.

[68] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529 (2016), pp. 484–489. DOI: 10.1038/nature16961. URL: https://www.nature.com/articles/nature16961.

[69] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 2256–2265. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[70] Elizabeth S. Spelke and Katherine D. Kinzler. "Core Knowledge". In: *Developmental Science* 10.1 (2007), pp. 89–96. DOI: 10.1111/j.1467-7687.2007.00569.x.

[71] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. 2nd. MIT Press, 2000.

[72] Alan M. Turing. "Computing Machinery and Intelligence". In: *Mind* LIX.236 (1950), pp. 433–460. DOI: 10.1093/mind/LIX.236.433.

[73] Alan M. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society* s2-42.1 (1936), pp. 230–265. DOI: 10.1112/plms/s2-42.1.230.

[74] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*. Curran Associates, Inc., 2017, pp. 5998–6008. URL: https://arxiv.org/abs/1706.03762.

[75] Christopher J.C.H. Watkins. "Learning from Delayed Rewards". PhD thesis. University of Cambridge, 1989. URL: http://www.gatsby.ucl.ac.uk/~dayan/papers/cjchthesis.pdf.

[76] Paul J. Werbos. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences". Ph.D. Dissertation. PhD thesis. Harvard University, 1974.

[77] Peter West et al. *The Generative AI Paradox: "What It Can Create, It May Not Understand"*. 2023. arXiv: 2311.00059 [cs.AI]. URL: https://arxiv.org/abs/2311.00059.

[78] Ronald J. Williams. "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning". In: *Machine Learning* 8.3-4 (1992), pp. 229–256. DOI: 10.1007/BF00992696.

Web and Other Sources

[41] Gary Marcus. *The Moral Compass of AI: Gary Marcus's Guide to Steering Technology Towards Human Values*. July 2024. URL: https://www.linkedin.com/pulse/moral-compass-ai-gary-marcuss-guide-steering-bc19e/.

[47] Melanie Mitchell. *LLMs and World Models – Part 1*. https://aiguide.substack.com/p/llms-and-world-models-part-1. 2023.

[50] Benjamin Murphy. "Rationalism and Empiricism: Will the Debate Ever End?" In: *Think* 9.24 (2010), pp. 35–46. DOI: 10.1017/S1477175609990200.

[51] David G. Myers and C. Nathan DeWall. *Psychology*. 13th. New York: Worth Publishers, 2019. ISBN: 9781319050627.

[53] OpenAI. "GPT-4 Technical Report". In: *arXiv preprint arXiv:2303.08774* (2023). URL: https://arxiv.org/abs/2303.08774.

[57] Sridharan Narayanan Rajarshi Roy. "How Explainable AI Reduces Bias". In: *International Journal of Intelligent Systems and Applications in Engineering* (2021). URL: https://ijisae.org/index.php/IJISAE/article/view/6902/5803.