Politecnico di Torino

Master's Degree in Artificial Intelligence and Data Analytics

A.Y. 2025-2026

March 2026

Design and Development of a Framework for Extracting Interpretable Symbolic Knowledge from Video Streams Exploiting Core Knowledge

Supervisors:

Stefano Quer
Giovanni Squillero

Candidates:

Giorgia Ghisolfo
Student ID: 332968

Abstract

The ability to generalize knowledge across novel situations is a defining feature of human intelligence. DeepMind's landmark results on Atari games exposed a revealing paradox: deep reinforcement learning agents can achieve superhuman scores yet collapse entirely when trivial changes are introduced to the environment. These systems do not learn in any meaningful sense; they optimize fragile heuristics over raw pixel statistics. As Melanie Mitchell and others have argued, such models lack an internal world model; a structured, causal representation of how objects behave and interact, and this absence is precisely what makes them brittle. The opacity of these systems is not merely an aesthetic limitation. When a neural agent cannot articulate why it acts as it does, it cannot be corrected or trusted in high-stakes settings. In a black-box model, there is no reasoning process to expose. There is only pattern matching, and pattern matching without structure that cannot generalize. The black box problem and the brittleness problem are, at root, the same problem.

This thesis takes a different path. Rather than training a network to approximate behavior from vast amounts of data, the proposed framework actively constructs explicit, human-readable models of environment dynamics directly from video streams. The system begins anonymous visual patches, without pre-defined object categories. The patches are identified solely by fundamental properties such as position and shape which the systems tracks over time using heuristics grounded in Core Knowledge; the fundamental cognitive primitives that underlying human physical reasoning from early infancy. Through rule inference, the system identifies causal regularities in object behavior, reconstructing the underlying mechanics of the environment as a set of interpretable symbolic rules. Every inferred rule can be read, inspected, and understood. As these rules are abstract and encode causal relationships rather than simple perceptual features, they can, in principle, be transferred across different environments. The system does not lose its understanding when encountering a new context; instead, it retains and adapts its symbolic model. This contrasts with deep learning models, whose learned weights are entangled with the specific sensory statistics of the training data and cannot be meaningfully reused. The structured representations produced by this framework are therefore portable across related environments without retraining, mirroring how humans transfer physical intuitions from one context to another.

The symbolic world model is further integrated with a reinforcement learning agent through a domain-agnostic environment wrapper, enabling the agent to be rewarded for generating causally coherent interactions rather than optimizing a handcrafted objective. Evaluated on Atari-inspired environments, this intrinsic causal reward drives the agent to near-optimal performance and, crucially, the learned

policy remains stable under structural perturbations including changes in object size, brick count, and visual appearance applied without any retraining. The same framework transfers to a second domain without architectural modification.

Together, these results suggest that interpretability and competence are not in tension, but they are complementary. A system that reasons causally about its environment can act effectively and explain itself, generalizes where black-box models fail, and lays the groundwork for AI that reasons about the world in a genuinely human-like way. The framework opens concrete research directions in autonomous surveillance, human-computer interaction, healthcare monitoring, and robotics, wherever trustworthy, transparent decision-making matters.

Contents

1 Introduction

In recent years, artificial intelligence has achieved remarkable progress, driven by dedicated research, increased computational power, and the availability of vast amounts of data. Deep learning models have produced outstanding results across a variety of domains [1], from natural language processing and video generation to drug discovery and genomics, occasionally surpassing human performance in specific tasks. These achievements are impressive, yet they also highlight intrinsic limitations. Despite their ability to solve specific problems, deep learning systems often lack a genuine understanding of the world: they learn statistical patterns without forming explicit internal models that represent concepts, hierarchies, or causal relationships. Moreover, most models exhibit limited generalization and brittle behavior, requiring fine-tuning or retraining when the environment changes even slightly. Their opacity further prevents interpretability, which constrains the deployment of AI in domains where reliability and accountability are critical, such as medicine or economics. Motivated by these challenges, this thesis explores the development of a cognitively inspired framework capable of constructing structured and interpretable representations of dynamic environments. The research investigates how to extract symbolic knowledge from raw perceptual input, manage ambiguity and uncertainty in object tracking and interaction modelling, and employ intrinsic motivation to guide adaptive behavior without relying on task-specific rewards. A key goal is to abstract object classes and interaction rules, enabling the system to generalize across environments and transfer learned knowledge to new contexts. The framework is not intended to compete with state-of-the-art performance in specific tasks; rather, it aims to lay the foundations for AI systems that are robust, interpretable, and capable of human-like reasoning about interactive environments.

The main contributions of this work are fourfold. First, it proposes a cognitively inspired architecture that integrates visual perception, symbolic reasoning, and reinforcement learning into a coherent pipeline. Second, it introduces a structured symbolic representation, extracting persistent objects and inferring generalized interaction rules to form an interpretable internal world model. Third, it incorporates a composite intrinsic motivation mechanism, combining event detection, causal impact estimation, and curiosity-driven exploration to promote adaptive behavior without predefined rewards. Finally, the framework emphasizes generalization, abstracting object classes and symbolic rules potentially enabling knowledge transfer across different environmental contexts.

The thesis is organized to guide the reader from background to implementation and evaluation. The following chapter reviews the state of the art in deep reinforcement learning, object-centric and neuro-symbolic approaches, and intrinsic mo-

tivation. Chapter 3 introduces the theoretical foundations, including insights from Core Knowledge theory and cognitive science that inform the framework. Chapter 4 and 5 presents the design and architecture of the system, detailing perception, object reconstruction, symbolic rule inference, and reinforcement learning integration. Chapter 6 describes the experimental evaluation in simplified two-dimensional arcade-like environments and analyses the results. Finally, Chapter 7 concludes the thesis by summarizing the key contributions and implications of the study and discusses the findings, highlights limitations, and outlines directions for future work.

2 Theoretical Background

The field of artificial intelligence has undergone deep transformations since its inception,evolving through distinct paradigms that reflect different philosophical and technical approaches to machine intelligence. This chapter provides the theoretical foundations necessary to understand the framework proposed in this thesis, exploring the evolution of learning paradigms in AI, some techniques of visual perception and the principles underlying symbolic reasoning and reinforcement learning.

2.1 The Evolution of Learning in Artificial Intelligence

2.1.1 Artificial Intelligence Paradigms

The history of artificial intelligence is characterized by the emergence and evolution of three major paradigms, each representing a distinct approach to providing machines the human-like reasoning abilities.
The symbolic paradigm, dominant from the 1950s through the 1980s. This paradigm was grounded in the hypothesis that human intelligence could be replicated through the manipulation of symbols, according to logical rules, to perform reasoning tasks. The systems built on this paradigm operated on explicit knowledge representations, using formal logic and rule-based reasoning to solve problems. These systems are grounded in formal logic and manually designed rule sets. Classic examples include expert systems such as MYCIN and Deep Blue, as well as semantic networks and logic programming frameworks [2]. They are capable of performing reasoning tasks and have achieved notable success in constrained domains where knowledge can be explicitly encoded. However, the symbolic paradigm faced significant limitations due to its high reliance on manually encoded knowledge. Indeed, when faced with tasks requiring pattern recognition, learning from experience, the management of uncertainty or incomplete information, it suffered from brittleness [2, 3].

These challenges gave rise to the connectionist paradigm, which gained prominence in the late 1980s and experienced a resurgence in the 2010s with the advent of deep learning [4]. Inspired by the structure and function of biological neural networks, connectionist systems learn representations directly from data through training processes that adjust connection weights between artificial neurons [4, 5]. This paradigm has achieved remarkable success in tasks such as image recognition, natural language processing and game playing, often surpassing human-level performance in specific domains [1]. Their capacity for robust and interpretable reasoning

remains limited. Deep learning still struggles with structured reasoning, causal inference and factual consistency. Despite the impressive capabilities of neural networks, their operation and internal representation often lacks transparency, giving rise to what is commonly referred to as the "black box problem" [3], which limits interpretability, accountability and trust. Indeed, many of these approaches compromised interpretability and relied on strong assumptions. Moreover, they lacked the rich semantic expressiveness characteristic of symbolic systems. These limitations provide strong motivation for integrating neural and symbolic paradigms.

In short, symbolic systems excel at reasoning but lack at perception, while neural networks excel at perception but struggle with reasoning [2]. This limitation has motivated the emergence of a hybrid paradigm, which seeks to combine the learning capabilities of neural networks with the interpretability and reasoning power of symbolic systems. Neuro-symbolic AI represents this third wave, aiming to bridge the gap between data-driven learning and knowledge-based reasoning as well as between logic-based inference and gradient-based learning. By integrating these paradigms, it aims to potentially offer systems that are both powerful and explainable [6]. Neuro-symbolic methods have demonstrated significant promise across a wide range of application areas, especially in scenarios that demand the combination of data-driven learning and structured symbolic reasoning [6]. This emerging paradigm reflects a synthesis of complementary approaches, bringing together the statistical strengths of deep learning, the formal foundations of symbolic logic and the adaptability of large-scale pretrained models.

Nonetheless, several important challenges remain. These include:

- Ensuring compatibility between the discrete symbolic structures used in symbolic approaches and the continuous vector representations typical of neural networks.
- Enabling the model to dynamically learn, adapt, modify, and refine logical rules.
- Achieving an optimal balance between system complexity and computational efficiency.
- Addressing the absence of unified architectural frameworks by developing a general architecture that systematically and reproducibly integrates neural and symbolic components.

Although several challenges persist, neuro-symbolic AI is widely regarded as a promising direction for developing systems capable of transparent, reliable and generalizable reasoning.

2.1.2 Symbolic and Hybrid Approaches in Artificial Intelligence

Symbolic AI operates on the principle that intelligence can be reduced to the manipulation of symbols representing concepts, objects and relationships. The knowledge

in symbolic systems is typically encoded using various representation schemes, including production rules, semantic networks, frames and logical predicates. These representations enable explicit reasoning processes such as deduction, induction and abduction, allowing systems to draw inferences, explain their decisions and incorporate domain knowledge provided by human experts. The advantages of symbolic approaches include interpretability, ability to incorporate prior knowledge and support for logical reasoning and explanation. However, they also present significant challenges: knowledge acquisition bottleneck, brittleness when facing situations not covered by explicit rule and difficulty in learning from raw sensory data. The manual encoding of knowledge is intensive and may not capture the full complexity of real-world domains [2]. Hybrid approaches seek to overcome these limitations by integrating symbolic and connectionist methods. Various integration strategies have been proposed, including systems where neural networks generate symbolic representations that are then processed by reasoning engines, architectures that embed symbolic knowledge into neural network structures and frameworks that use symbolic reasoning to guide neural learning processes [6]. Recent work in neuro-symbolic AI has demonstrated promising results in domains requiring both pattern recognition and logical reasoning, such as visual question answering, knowledge graph completion, and program synthesis. Following the advent of neuro-symbolic artificial intelligence, a paradigm shift has been induced in the way machines learn. This approach combines deductive, rule-based learning, typical of symbolic artificial intelligence, with the inductive and pattern-recognition capabilities characteristic of neural networks. The result is a hybrid approach that leverages the strengths of both domains, leading to a more comprehensive learning methodology. Neuro-symbolic concepts demonstrate strong compositional generalization: new concepts can be created by structurally combining previously learned or existing ones [7]. This compositionality naturally allows for a decomposition of the learning problem, in order to learn and recognize basic concepts. During the inference phase, these concepts can be recombined to achieve the objectives of more complex tasks. Neuro-symbolic AI is one of the most exciting directions in AI research today. It aims to create intelligent systems capable of learning and reasoning like humans.

2.1.3 Reinforcement Learning: Learning Through Experience

Reinforcement learning (RL) represents a distinct learning paradigm in which an agent learns to make decisions by interacting with an environment. Unlike supervised learning, which requires labeled training data, or unsupervised learning, which seeks to discover patterns in unlabeled data, reinforcement learning is based on the principle of learning from the consequences of actions through trial and error [8]. The agent takes actions, observes their consequences, and receives feedback in the form of rewards. At the beginning of learning, an agent may have limited or no prior knowledge of the environment. In model-free approaches, no explicit model is available, and learning relies entirely on interaction. In contrast, model-based methods assume a known model or aim to learn it explicitly. This flexibility makes reinforcement learning suitable for complex environments, where providing a complete dataset or manually specifying optimal behavior is impractical. The core structure

of Reinforcement Learning is modelled through a Markov Decision Process (MDP), which provides a mathematical framework for sequential decision-making [8]. An MDP is defined by a finite or continuous set of states representing the possible situations an agent can encounter, a set of actions available to the agent and a transition function that specifies the probability of moving from one state to another after taking a particular action. Additionally, a reward function assigns numerical feedback to each state-action pair, quantifying the immediate benefit of a given choice. By observing the current state, selecting an action, and receiving a reward and a new state, the agent incrementally improves its decision-making strategy [9]. Over time, the agent aims to learn an optimal policy, which is a strategy mapping states to actions. The goal of this policy is to maximize the expected cumulative reward across future interactions, balancing short-term gains with long-term benefits through exploration and exploitation. Several agent design approaches exist, including methods based on value estimation and approaches that learn action values directly, such as Q-learning. More advanced techniques combine different strategies to improve learning efficiency. Due to its ability to learn from experience, reinforcement learning has achieved notable success in domains such as game playing, robotics, and resource management [9].

2.2 Computer Vision Techniques

Visual perception is fundamental to many AI applications, enabling systems to extract meaningful information from images and video streams. One of the most relevant goals of computer vision is to mimic human visual perception. Robustness is commonly assessed by comparing algorithmic performance to human-level perception under similar conditions. In this context, robustness refers to the ability to extract task-relevant visual information, even when this information is sparse, corrupted, or significantly different from a previously observed representation, while maintaining tolerance to outliers. This section explores both classical and modern approaches to video analysis and feature extraction.

2.2.1 Classical Techniques for Video Analysis

Before the advent of deep learning, computer vision systems primarily relied on deterministic techniques and manually engineered features. Traditional algorithms such as edge detection methods, threshold-based segmentation, and morphological processing were widely employed to extract structural information from images. Later, handcrafted feature descriptors including SIFT [10] (Scale-Invariant Feature Transform), SURF [11] (Speeded-Up Robust Features), and HOG [12] (Histograms of Oriented Gradient) were introduced to capture local patterns useful for object recognition and matching tasks. Although these methods achieved notable success, they suffered from several inherent limitations, particularly in their sensitivity to noise, changes in illumination, and poor generalization across diverse environments [10]. As a result, significant domain expertise and extensive manual effort were required to design robust feature representations, which constrained their scalability and adaptability to complex real-world scenarios.

Originally, computer vision relied heavily on handcrafted features and algorithmic techniques designed based on understanding of visual perception and signal processing. These principles underpin a classical computer vision pipeline, typically composed of preprocessing, feature extraction, and pattern recognition stages.
Preprocessing techniques include operations such as noise reduction, contrast enhancement, and colours space transformations that prepare raw visual data for subsequent analysis. Edge detection algorithms, such as the Canny edge detector and Sobel operator, identify boundaries between regions with different intensity characteristics, providing fundamental structural information about objects in the scene.
Feature extraction methods in classical computer vision include corner detectors like Harris corners [13] and FAST [14] (Features from Accelerated Segment Test), which identify points of interest that can be reliably detected across different views. Descriptors such as SIFT and SURF provide representations of local image patches that are robust to changes in scale, rotation, and illumination. These features enable tasks such as object recognition, tracking, and image matching. Although these classical techniques have been largely superseded by deep learning methods in many applications, they remain valuable in contexts where computational resources are limited, interpretability is crucial, or training data is scarce. Moreover, the principles underlying these techniques continue to inform modern approaches and provide insights into visual perception.

2.2.2 Visual Perception in Dynamic Environments

The analysis of dynamic environments is particularly relevant for applications such as autonomous navigation, human-robot interaction, video surveillance, and sports analytics. In the context of video games, which serve as testbeds for this thesis, dynamic environments present controlled yet complex scenarios where agents must perceive, reason about, and respond to changing situations in real-time. Understanding dynamic environments requires not only processing individual images but also tracking changes over time and recognizing temporal patterns. Visual perception in such contexts involves several key challenges: detecting and segmenting objects, tracking their motion across frames, recognizing actions and behaviors, and predicting future states.
Object detection and segmentation are fundamental capabilities for understanding scenes. Object detection identifies the presence and location of objects within images, typically using bounding boxes to localize them. Semantic segmentation assigns a class label to each pixel, providing a more detailed understanding of scene structure. Instance segmentation combines both tasks, identifying individual object instances and their precise boundaries.
Temporal reasoning in video analysis requires integrating information across multiple frames. Recurrent neural networks and their variants, such as Long Short-Term Memory (LSTM) networks [15] , have been widely used to capture temporal dependencies. Three-dimensional convolutional networks extend spatial convolutions into the temporal dimension, enabling the extraction of spatiotemporal features directly from video data. In this work, the temporal complexity of video is addressed in a modular manner, by first extracting spatial features from individual frames and delegating temporal modeling to subsequent stages.

2.2.3 Deep Learning for Feature Extraction from Video Streams

The advent of Deep Learning has revolutionized the field of Computer Vision, enabling models to automatically learn hierarchical representations of images and enabling end-to-end learning of feature representations directly from raw pixel data through CNNs (Convolutional Neural Networks).

Convolutional Neural Networks for Visual Representation Learning

Convolutional Neural Networks (CNNs or ConvNets) form the foundation of modern visual recognition systems and represent a major paradigm shift in computer vision. It enabled models to automatically learn hierarchical representations, directly from raw data, where lower layers capture low-level features such as edges and textures, while higher layers represent more abstract concepts and object categories.

Convolutional Neural Networks are designed to process data represented as multidimensional arrays. These structures include one-dimensional signals such as time series and language sequences, two-dimensional data such as images, and three-dimensional data such as videos. The effectiveness of CNNs is rooted in four fundamental architectural principles that exploit the compositional nature of visual data: local connectivity, shared weights, pooling operations, and hierarchical multilayer organization. Unlike fully connected networks, each convolutional neuron is connected only to a local region of the input, significantly reducing the number of parameters and improving generalization. Weight sharing allows the same convolutional filter to detect a specific visual pattern regardless of its spatial location within the image.

A typical CNN architecture is organized as a sequence of stages, each composed of convolutional layers, nonlinear activation functions, such as the Rectified Linear Unit (ReLU), and pooling layers [1]. In convolutional layers, neurons are grouped into feature maps, where each map applies a specific filter across the entire input, producing a representation that highlights the presence of particular local patterns. Pooling layers are responsible for locally aggregating information, reducing the spatial dimensionality of feature maps and introducing a degree of invariance to small translations and distortions in the input. The most common operation is max pooling, which selects the maximum value within a local neighborhood. The alternation of convolution, nonlinearity, and pooling enables the network to progressively construct increasingly abstract and robust representations. In the visual domain, low-level features such as edges and orientations are combined into more complex patterns, which in turn form object parts and ultimately complete objects. This hierarchical aggregation allows the learned representations to remain stable under local variations in position, shape, or appearance. In other words, CNNs leverage convolutional operations to capture local spatial patterns, which are gradually combined across deeper layers to form increasingly abstract representations. This hierarchical learning process allows the network to identify complex visual structures without relying on manually designed features. The effectiveness of CNNs was clearly demonstrated in the ImageNet challenge [16], where deep architectures achieved unprecedented performance, significantly outperforming traditional computer vision approaches. This progress has made it possible to tackle complex tasks

such as image segmentation, with high accuracy.

Evolution of Deep Architectures for Visual Representation Learning

Architectures for image understanding have evolved considerably since the introduction of AlexNet in 2012 [16]. This model demonstrated the effectiveness of deep convolutional neural networks trained on large-scale datasets such as ImageNet, significantly outperforming traditional computer vision methods and marking a turning point for deep learning in visual recognition tasks. VGGNet [17] further emphasized the importance of network depth, showing that stacking small convolutional filters could improve representational capacity while maintaining manageable computational complexity. Its simple and uniform architecture also made it a widely adopted backbone for transfer learning and feature extraction. ResNet [18] addressed one of the key challenges in training deep networks, namely the vanishing gradient problem. It introduced residual skip connections, which allow gradients to flow more effectively through the network. Furthermore, more recent architectures such as EfficientNet [19] optimize both accuracy and computational efficiency through neural architecture search.
Video understanding requires architectures that can process temporal information. Feature extraction from video streams for downstream tasks often involves using pre-trained networks as feature extractors, transferring knowledge learned on large-scale datasets to specific applications. The features extracted from intermediate or final layers of these networks provide rich representations that can be used for tasks such as action recognition, video classification, and temporal localization of events. The integration of deep learning with computer vision has enabled unprecedented performance in visual recognition tasks, but often at the cost of interpretability. The features learned by deep networks, while effective, do not always correspond to human-understandable concepts, motivating research into explainable AI and the integration of symbolic reasoning with learned representations.

From Feature Extraction to Dense Prediction: Semantic Segmentation Architectures

Early convolutional architectures were primarily developed for image-level classification tasks [19]; however, many real-world applications require a more fine-grained understanding of visual scenes. In particular, tasks such as scene understanding and environment modelling demand dense, pixel-level predictions, where each pixel is assigned a semantic label rather than a single global class. This shift in objectives led to the development of semantic segmentation architectures, which extend the principles of convolutional feature learning to dense prediction problems. A key milestone in this evolution is represented by Fully Convolutional Networks (FCNs) [20], which replace fully connected layers with convolutional ones, enabling networks to process images of arbitrary size and produce spatially structured outputs. Building upon this idea, the U-Net architecture [21] introduced an encoder-decoder structure specifically designed to preserve spatial precision while leveraging deep hierarchical features. The encoder progressively extracts abstract visual representations through convolution and pooling operations, while the decoder reconstructs a dense segmentation map by restoring spatial resolution. A defining characteristic of U-Net is the

use of skip connections between corresponding encoder and decoder layers. These connections allow low-level spatial information to be directly combined with high-level semantic representations, mitigating the loss of fine-grained details caused by repeated downsampling. As a result, U-Net achieves accurate boundary localization while maintaining strong semantic consistency [21]. Due to its effectiveness, architectural simplicity, and ability to perform well even with limited training data, U-Net and its variants have been widely adopted across domains, including medical imaging, robotics, and video game analysis. In particular, U-Net-based models are well suited for environments characterized by structured layouts and clearly defined visual entities, making them a natural choice for perceptual preprocessing in dynamic 2D environments and an effective perceptual backbone for downstream reasoning and decision-making systems.

2.3 Knowledge Representation and Reasoning

The ability to represent knowledge in a structured, manipulable form and to perform reasoning over such representations is central to artificial intelligence. The knowledge representation and reasoning concerned with how knowledge about the world can be formally encoded and manipulated in order to support intelligent behavior. Its primary goal is to design representational formalisms that enable artificial systems to draw logical conclusions, explain their decisions, and reason consistently about complex domains. As emphasized in classical AI literature, effective knowledge representation must balance expressive power, computational efficiency, and interpretability [22]. This section examines approaches to symbolic knowledge representation and their role in building interpretable AI systems.

2.3.1 Symbolic Representations

Symbolic knowledge representation is based on discrete symbols that explicitly encode concepts, entities, and their relationships. This approach stems from a cognitive perspective according to which human reasoning relies on symbolic manipulation, a view that profoundly shaped early artificial intelligence research [23]. Due to both the representations and the inference mechanisms operate on explicit symbols, symbolic systems closely mirror human modes of reasoning and are therefore inherently interpretable. These representations are commonly employed to structure data and metadata, where the semantic meaning of each symbol must be carefully defined and considered during information analysis. Furthermore, since human communication relies predominantly on the use of symbols, symbolic representations also play a fundamental role in the study and analysis of natural language. Symbolic representation involves encoding knowledge using discrete symbols and the relationships between them. Unlike distributed representations in neural networks, where meaning emerges from patterns of activation across many units or is distributed across numerical parameters, symbolic representations explicitly denote concepts and their properties, encoded in well-defined structures. Each symbol has a clear semantic interpretation, which makes symbolic representations particularly suitable for reasoning, explanation, and the integration of expert knowledge [24].

2.3.2 Semantic Networks

Semantic networks are graph-based models designed to organize, and represent knowledge, through interconnected nodes and labeled edges. In this context, nodes correspond to concepts, objects, or entities, while edges denote semantic relations between them, such as *is-a* (classification), *part-of* (composition), or associative links. This structure allows knowledge to be represented in a way that mirrors conceptual organization, making relationships explicit and visually interpretable. One of the central advantages of semantic networks is their ability to represent taxonomic hierarchies. Through hierarchical relations (e.g., canary → bird → animal), properties associated with more general categories can be inherited by more specific ones. For instance, if the concept bird is associated with the property has wings, that property can automatically be inferred for canary without being redundantly specified. This mechanism, known as inheritance reasoning, supports efficient knowledge organization and reduces duplication. From a cognitive perspective, semantic networks were strongly influenced by early models of human memory and conceptual organization. A foundational contribution was provided by M. Ross Quillian (1968), who proposed semantic memory models structured as hierarchical networks in which concepts are connected by labeled relations. His work demonstrated how hierarchical organization could explain verification time in human reasoning tasks, suggesting that cognitive processes may operate through structured relational networks. Later, John F. Sowa expanded the formal foundations of semantic networks by integrating them with logic in his theory of *Conceptual Graphs*, providing a more rigorous semantic interpretation and linking graph structures to formal logic systems. Their graphical notation makes them particularly intuitive, facilitating both human understanding and computational reasoning. Although later symbolic formalisms, such as description logics and ontology languages, provided more formal semantics, semantic networks remain a foundational paradigm in knowledge representation.

2.3.3 Interpretable Models

In knowledge representation, a variety of symbolic models have been developed to describe objects, situations, and events in a structured and interpretable way. Among these, frames and scripts extend semantic networks by introducing richer internal structures capable of capturing stereotypical situations and recurring sequences of actions. A frame represents a stereotypical object or situation through a set of attributes, known as slots, which may be filled with specific values, default assumptions, or references to other frames. This structured representation allows frames to encode both declarative knowledge about entities and procedural knowledge about the actions typically associated with a given context. Scripts, by contrast, are specifically designed to model common sequences of events. They describe how actions usually unfold in familiar situations and are particularly effective for narrative understanding and for predicting behavior in routine scenarios. Both models were originally proposed to account for aspects of human comprehension and memory [25]. A more formal and explicit approach to knowledge representation is provided by ontologies, which define a shared conceptual framework for a given domain. Ontologies specify classes, relations, constraints, and axioms using logic-based languages, often grounded in Description Logic. By enforcing precise

semantics, they enable automated reasoning tasks such as consistency checking, classification, and logical inference. Their standardized nature also supports knowledge sharing and reuse across different systems, making ontologies a central component of areas such as the Semantic Web, intelligent information retrieval, and large-scale knowledge integration [26, 27]. Another major category of symbolic approaches is logic-based representations, which encode knowledge as sets of formal logical statements. First-order logic provides a powerful framework for expressing complex relationships among entities through predicates, variables, and quantifiers, enabling rigorous deductive reasoning. Alongside this, production rules, typically expressed as if-then statements, capture procedural knowledge and support both forward chaining, in which conclusions are derived from known facts, and backward chaining, in which goals are traced back to the conditions required to satisfy them.

These mechanisms have played a foundational role in expert systems and rule-based AI architectures. The strength of logical representations lies in their ability to guarantee consistency, derive logical consequences, and provide clear explanations for inferred conclusions, if inference rules are correctly applied. More broadly, the interpretability of symbolic representations constitutes a key advantage: the meaning of symbols, rules, and relations can be directly understood by humans, facilitating debugging, validation, and trust in AI systems. However, constructing such representations manually or extracting them from raw, unstructured data remains a significant challenge.

Despite these limitations, symbolic knowledge representation continues to be a crucial component of artificial intelligence, particularly in applications that require explainability, structured reasoning, and formal guarantees. Due to purely symbolic systems are not sufficient on their own, their integration with data-driven approaches offers promising directions for the development of interpretable and robust AI systems. Ongoing research in neuro-symbolic methods aims to automate the extraction of symbolic knowledge from data while preserving the powerful reasoning capabilities that symbolic representations provide.

2.4 Reinforcement Learning Principles

This section delves deeper into the principles and methods of reinforcement learning, with particular focus on Deep Q-Networks, which play a central role in the framework proposed in this thesis.

2.4.1 Theoretical foundations

The mathematical foundation of reinforcement learning rests on the framework of Markov Decision Processes (MDPs). An MDP formalizes sequential decision-making problems in which an agent interacts with an uncertain environment and seeks to maximize a cumulative reward over time. An MDP is defined by the tuple (S, A, P, R, γ), where:

- S is the state space, i.e., the set of all possible configurations of the environment.

- A is the action space, i.e., the set of all actions available for the agent in each state.
- $P(s' \mid s, a)$ is the transition probability function, i.e., the probability that, after performing action a in state s, the system transitions to state s'.
- $R(s, a)$ is the reward function, which provides a numerical value quantifying the desirability of transitioning from state s to state s' after performing action a, assigning a scalar value to each state-action pair.
- $\gamma \in [0, 1)$ is the discount factor that determines the relative importance of immediate versus future rewards. Smaller values of γ encourage myopic behavior focused on immediate gains, whereas values closer to 1 promote long-term planning.

A fundamental assumption in MDPs is the Markov property, which states that the future evolution of the system depends solely on the current state and action, independent of the full history of past states. Mathematically:

$$P(s' \mid s, a, s_{t-1}, a_{t-1}, \dots) = P(s' \mid s, a)$$

This memory-less characteristic significantly simplifies the analysis and allows the use of dynamic programming techniques. The agent tries to maximize the total cumulative reward. This cumulative reward is defined as:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k})$$

It is not simply the sum of instant rewards, but the discounted reward over time. Due to the γ it is possible to give more weight to immediate or future rewards.

Policie

In reinforcement learning, a policy π defines the agent's behavior by specifying which action should be taken in each possible state [9]. Formally, a policy can be represented either as a function or as a probability distribution over actions. Two main types of policies can be distinguished. A deterministic policy $\pi(s)$ assigns a single, fixed action to each state. In contrast, a stochastic policy $\pi(a \mid s)$ associates a probability with each available action. The agents progressively learn an optimal policy through continuous interaction with the environment. This learning process requires collecting experience by visiting new states and testing different actions. Through this interaction, the agent refines its strategy in order to maximize the cumulative reward obtained over time. A central challenge in this process is the exploration-exploitation trade-off. Exploration encourages the agent to try new and unfamiliar actions in order to discover potentially better rewards and improve its knowledge of the environment. On the other hand, exploitation focuses on selecting actions that are already known to yield high returns, thus capitalizing on past experience. Achieving an effective balance between these two objectives is crucial for learning a robust and optimal policy.

Value Function

Value functions are used to evaluate policies and play a fundamental role in many reinforcement learning algorithms. The state-value function $V_\pi(s)$ estimates the expected cumulative reward obtained when starting from state s and following policy π.

$$V_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})\right]$$

The action-value function $Q_\pi(s, a)$ measures the expected return after taking action a in state s and subsequently following policy π, are also called Q-values [22].

$$Q_\pi(s, a) = \mathbb{E}\left[R(s, a, s') + \gamma V_\pi(s')\right]$$

The central objective of reinforcement learning is to identify an optimal policy π^* that maximizes these value functions across all states.

Potential and areas of application

MDPs provide the theoretical foundation for a wide range of reinforcement learning algorithms, including value iteration, policy iteration, temporal-difference learning and Q-learning. These methods exploit the mathematical structure of MDPs to iteratively improve policies, either when the environment model is known or when it must be learned from experience. As a result, MDPs play a crucial role in enabling autonomous agents to learn effective behaviors through interaction with complex and uncertain environments. Due to their strong theoretical foundation and flexibility, MDPs are extremely important across numerous application domains. In robotics and automation, for example, robots rely on MDPs to navigate uncertain and dynamic environments. Beyond robotics, MDPs are also widely applied in healthcare for treatment planning and optimization, in finance for trading strategies and risk management, and in game AI for developing intelligent and adaptive agents. This broad applicability highlights the practical significance of MDPs in real-world decision-making problems [9].

The path to success

The primary objective of reinforcement learning is to identify an optimal policy that maximizes long-term rewards. This objective leads to one of the most fundamental results in RL theory: the Bellman Optimality Equation [28]. This equation provides a recursive definition of optimal value functions, both for states and for state-action pairs [29].

- **Optimal State Value Function** $V^*(s)$:

$$V^*(s) = \max_a \sum_{s'} P(s' \mid s, a)\left[R(s, a, s') + \gamma V^*(s')\right] \tag{2.1}$$

- **Optimal Action Value Function** $Q^*(s, a)$:

$$Q^*(s, a) = \sum_{s'} P(s' \mid s, a)\left[R(s, a, s') + \gamma \max_{a'} Q^*(s', a')\right] \tag{2.2}$$

The Bellman optimality equation removes any dependence on a particular policy. Instead, it establishes a self-consistency condition that optimal value functions must satisfy. Specifically, the optimal state-value function is defined as the maximum expected return achievable by selecting the best possible action in a given state. Similarly, the optimal action-value function is expressed as the immediate reward plus the highest expected future return obtainable from the subsequent state [29]. The Bellman Optimality Equation serves as the theoretical cornerstone for a wide range of optimal control algorithms used in reinforcement learning.

2.4.2 Types of Reinforcement Learning

Several methods have been developed to solve Markov Decision Processes (MDPs), differing mainly in how they use information about the environment. These approaches can be broadly classified into model-based and model-free methods. Model-based methods rely on a known or learned model of the environment to predict future states and rewards, allowing agents to plan their actions in advance. In contrast, model-free methods do not assume any prior knowledge of the environment and instead learn optimal behaviors directly through experience and interaction. Understanding this distinction is fundamental for analyzing how different reinforcement learning algorithms operate and adapt to uncertainty.

Model-Based Reinforcement Learning

Dynamic Programming Methods Dynamic programming is a model-based approach and solves MDPs by iteratively improving value functions through:

- **Policy Iteration** alternates between policy evaluation and policy improvement until the policy converges to the optimal policy.
  - *Policy evaluation* consists of estimating the value function associated with a given policy π, typically by computing the expected return obtained when the agent follows that policy from each state. This step answers the question of how good the current policy is, by assessing its long-term performance in the environment.
  - *Policy improvement*, on the other hand, aims to enhance the current policy by exploiting the information obtained during policy evaluation. In this phase, the policy is updated by selecting actions that yield higher expected returns according to the evaluated value function, resulting in a new policy that is guaranteed to be at least as good as the previous one [29].
- **Value Iteration:** updates the value function until it converges to the optimal value function.

$$V_{k+1}(s) = \max_{a \in A} \sum_{s' \in S} P(s' \mid s, a) \left[R(s, a, s') + \gamma V_k(s') \right]$$

where:

- k is the iteration index of the algorithm;
- $V_k(s)$ is the estimated value of state s at iteration k;
- $V_{k+1}(s)$ is the updated value estimate at the next iteration;
- A is the set of available actions;
- S is the set of possible next states;
- $P(s' \mid s, a)$ is the transition probability to state s' given state s and action a;
- $R(s, a, s')$ is the reward received after transitioning to s';
- $\gamma \in [0, 1)$ is the discount factor.

Dynamic programming breaks down complex tasks into smaller subtasks. It then models problems as workflows of sequential decisions made in discrete time steps. Each decision is made based on the possible resulting next state, since dynamic programming considers potential future states and reward signals when selecting actions. Indeed, dynamic programming is a value-based method, meaning it selects actions based on their estimated values according to a policy that aims to maximize its value function. An agent's reward R for a given action is defined as a function of that action a, the current environmental state s, and the potential next state s':

$$R(s, a) = \sum_{s' \in S} R(s, a, s')\, P(s' \mid s, a)$$

where $P_t(s' \mid s, a)$, called the *transition probability function*, denotes the probability that the system is in state s' at time $t + 1$ when the agent chooses action a in state s at time t [9]. Dynamic programming seeks to maximize cumulative rewards by consistently selecting the best possible actions at each state, using recursive relationships such as the Bellman equations [28]. Dynamic programming theory guarantees the existence of at least one optimal stationary policy. However, classical dynamic programming techniques require full knowledge of the reward function and transition probabilities, which are generally unavailable in real-world scenarios.

Model-Free Reinforcement Learning

Monte Carlo methods The Monte Carlo method assumes a black-box environment, making it a model-free approach. Since no prior knowledge of the environment is available, Monte Carlo methods rely entirely on direct interaction to gather information, so these methods are purely experience-driven. These methods sample sequences of states, actions and rewards exclusively through interaction with the environment. As a result, learning occurs through trial and error [29] rather than through predefined probability distributions or transition models. This experiential learning process enables Monte Carlo methods to adapt to unknown environments, but it also requires a large number of episodes to achieve reliable performance. Monte Carlo methods estimate value functions by computing the average return for each state-action pair based on observed episodes. Consequently, Monte Carlo methods must wait until an entire episode (or planning horizon) is completed before updating their value estimates and policies. Within this framework, two main variants can be distinguished: First-Visit and Every-Visit Monte Carlo.

- **First-Visit Monte Carlo** estimates the value of a state by averaging the returns obtained only on the first visit to that state in each episode. This avoids updating the same state multiple times within the same episode, reducing the risk of bias in the results.
- **Every-Visit Monte Carlo**, on the other hand, considers all visits to a state within each episode. The state's value is estimated by averaging the returns obtained in each visit. This method uses more data than First-Visit but may introduce greater correlation between observations.

In both variants, the value function is updated according to:

$$V(s) \leftarrow V(s) + \alpha \left[G_t - V(s) \right]$$

where $V(s)$ is the estimated value of state s, α is the learning rate, and G_t is the actual return obtained from time step t.

Temporal Difference learning Temporal-Difference (TD) learning is a hybrid approach that combines the concepts of dynamic programming and Monte Carlo methods. It relies on experience to learn from interaction with the environment as Monte Carlo method, indeed is model-free approach. It also incorporates the bootstrapping idea from dynamic programming by updating value estimates incrementally, based on existing predictions and after each transition, without waiting for the final outcome of an episode. In this way, TD learning algorithms adjust the estimated value of a state and consequently the agent's policy, incrementally after each step, enabling faster and more continuous learning compared to waiting for complete episodes. As its name suggests, a TD learning agent adjusts its policy based on the difference between the predicted reward and the reward received at each state. In other words, while both dynamic programming and Monte Carlo methods focus on the rewards obtained, TD learning additionally evaluates the prediction error between expected and observed rewards. This difference, known as the temporal difference error, is then used to update value estimates for subsequent states immediately, without waiting for the episode to terminate.

$$V(s_t) \leftarrow V(s_T) + \alpha \left[R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \right]$$

where:

- $V(s_t)$ is the estimated value of state s at time step t;
- $\alpha \in (0, 1]$ is the learning rate, controlling the update step size;
- R_{t+1} is the reward received after transitioning from s_t;
- $\gamma \in [0, 1)$ is the discount factor;
- $V(s_{t+1})$ is the estimated value of the next state s_{t+1}.

The temporal difference error δ_t is defined as:

$$\delta_t = R_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$

so that the update rule can be written compactly as:

$$V(s_t) \leftarrow V(s_t) + \alpha\, \delta_t$$

TD learning exists in multiple variants, among which State-Action-Reward-State-Action (SARSA) and Q-learning are two of the most prominent.
SARSA is an on-policy TD algorithm, meaning it evaluates and improves the same policy that it follows during learning, therefore updates the action-value function based on the action taken by the current policy:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\, [R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

where:

- $Q(s_t, a_t)$ is the estimated value of taking action a_t in state s_t;
- $\alpha \in (0, 1]$ is the learning rate;
- R_{t+1} is the reward received after the transition;
- $\gamma \in [0, 1)$ is the discount factor;
- $Q(s_{t+1}, a_{t+1})$ is the estimated value of the next state-action pair, where a_{t+1} is the action chosen by the **current policy**.

Q-Learning is a model-free, off-policy reinforcement learning algorithm that aims to directly approximate the optimal action-value function, independently of the policy used to generate behavior [30]. Q-Learning separates the behavior policy, which directs exploration to gather new experiences, from the target policy, which focuses on exploiting the current value estimates to guide optimal decisions. The behavior policy may include exploratory mechanisms such as ε-greedy action selection, while the target policy is implicitly defined as the greedy policy with respect to the learned action-value function. This separation allows Q-Learning to learn the optimal policy regardless of how the agent behaves during data collection, provided that sufficient exploration occurs. For this reason, the algorithm is often described as exploration-insensitive, meaning that its convergence properties do not depend on the specific exploration strategy adopted, as long as all state-action pairs are visited sufficiently often [9]. The core update rule of Q-Learning is derived from the Bellman optimality equation and is defined as follows:

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

where:

- α is the learning rate,
- γ is the discount factor,

- R_{t+1} is the reward received after taking action a_t in state s_t,
- $\max_a Q(s_{t+1}, a)$ represents the maximum estimated future return obtainable from the next state.

The algorithm updates the action-value function using the maximum estimated reward of the subsequent state, allowing it to iteratively improve its policy based on expected future returns. Since Q-Learning does not require explicit knowledge of the transition probabilities or reward function, it can learn directly from raw experience, making it particularly suitable for environments where a model is unavailable or difficult to construct [31]. Once the optimal action-value function $Q^*(s, a)$ has been learned, the optimal policy is obtained by selecting, in each state, the action that maximizes the estimated Q-value:

$$\pi^*(s) = \arg\max_a Q^*(s, a)$$

For convergence to the optimal solution, it is necessary that every state-action pair is visited infinitely often and that the learning rate decreases appropriately over time. Under these conditions, also called stochastic approximation conditions, the action-value estimates are guaranteed to converge to their optimal values [30].

2.4.3 Deep Q-Networks

Deep Q-Networks (DQN), introduced by Mnih et al. in 2015 [28], revolutionized reinforcement learning by demonstrating that deep neural networks could successfully approximate Q-functions in high-dimensional state spaces. Prior to DQN, Q-learning was limited to small, discrete state spaces due to its reliance on tabular representations. By integrating convolutional neural networks (CNNs) with Q-learning [28], DQN enabled end-to-end learning directly from raw sensory inputs, such as pixel-based game screens. This innovation allowed reinforcement learning agents to achieve human-level performance on several Atari 2600 games using only visual input and game scores as feedback, without any handcrafted features.

The DQN algorithm addresses two key challenges that arise when combining Q-learning with neural networks: instability due to correlations in the sequence of observations and non-stationarity of the target values used for training. In standard online reinforcement learning, consecutive observations are highly correlated. Training a neural network on sequential samples violates the independent and identically distributed (i.i.d.) assumption required for stable stochastic gradient descent, leading to oscillations or divergence. In addition, Q-learning relies on bootstrapped targets derived from the same network being trained. As network parameters change during learning, the target values also change, resulting in a moving target problem that destabilizes training.

DQN employs two critical innovations to stabilize learning [28]. The first innovation was inspired by biological mechanism, the experience replay stores agent transitions, at each time-step, in a replay buffer. In addition, during training, samples mini batches uniformly for training. This technique breaking temporal correlations between past experience, enabling more efficient use of past experience and stabilizes learning through randomized training batches. This allows gradient updates to better approximate the true expectation of the Bellman equation.

The second innovation was targeting networks. This technique maintains a separate copy of the Q-network whose parameters are updated less frequently, providing more stable target values for the loss function, this help to reduces harmful feedback loops and prevents divergence. These techniques enable successful learning in complex domains such as Atari 2600 games, where the state space consists of raw pixel inputs.

Loss Function and Optimization The DQN loss function is designed to minimize the difference between the current Q-value estimates and the target values defined by the Bellman optimality equation. Using a target network with parameters θ^-, the algorithm stabilizes learning by providing a fixed reference for the updates. In practice, DQN can be seen as a method that approximates the Q-function with a deep neural network and iteratively solves the Bellman optimality equation via stochastic gradient descent.

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim \mathrm{U}(D)} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i) \right)^2 \right]$$

where:

- $L(\theta)$ is the loss function to be minimized with respect to the network parameters θ;
- $(s, a, r, s') \sim \mathrm{U}(D)$ denotes that the transitions are sampled uniformly from the replay buffer D;
- r is the immediate reward received after taking action a in state s;
- $\gamma \in [0, 1)$ is the discount factor;
- θ_i^- are the parameters of the target network, kept fixed during updates to stabilize training, at iteration i;
- $\max_{a'} Q(s', a'; \theta_i^-)$ is the maximum Q-value over all possible next actions a', estimated by the target network;
- $Q(s, a; \theta_i)$ is the Q-value estimated by the online network for the current state-action pair (s, a);
- $r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)$ is the TD error, measuring the discrepancy between the current estimate and the bootstrapped target based on the best possible next action [1].

One of the most significant aspects of DQN is its ability to learn directly from raw sensory inputs, requiring little to no prior domain knowledge or handcrafted feature engineering. Experimental results show that the DQN agent is capable of achieving performance levels comparable to, and in some cases exceeding, those of human players across a wide range of Atari 2600 games [28]. These findings highlight the robustness of deep reinforcement learning methods, showing that complex behaviors can emerge solely from interaction with the environment and scalar reward feedback. The model's success across diverse gaming environments indicates its potential applicability to broader decision-making problems beyond the Atari benchmark.

2.5 Atari-like Environments as Testbeds

Over the past decade, two-dimensional video games, especially those from the Atari 2600 game suite, have become standard benchmarks for the evaluation of reinforcement learning and artificial intelligence systems [32]. By offering high-dimensional visual inputs, diverse dynamics, and well-defined evaluation protocols within a controlled setting, these environments bridge the gap between simple toy problems and real-world applications. This section examines the characteristics and inherent challenges that make Atari-like environments particularly suitable as experimental testbeds for AI research.

2.5.1 Challenges of two-dimensional Game Environments

Atari games present diverse challenges that require different capabilities [33]. In this study, we concentrate our analysis on the Atari game as a representative benchmark environment. This choice allows us to highlight several fundamental properties shared by many Atari games suit. More in details, these environments often require strong temporal reasoning. Temporal reasoning is essential in many games, where success depends on understanding sequences of events, anticipating future states, and planning multi-step strategies. In some cases, precise timing of actions is crucial, while in others, rewards are delayed and the consequences of decisions become apparent only after multiple steps. For example, Breakout requires predicting where the ball will land based on its current trajectory and velocity. Furthermore, the agent must plan the paddle's movements several steps in advance, especially when the ball bounces off bricks or edges. This corresponds to the need for multi-step planning and anticipation of future states.

Exploration and credit assignment pose additional challenges. Many Atari games sparse reward structures, in which positive feedback is infrequent, making difficult for learning algorithms to discover effective strategies. Furthermore, long action sequences with delayed outcomes complicate the attribution of credit to individual actions that ultimately lead to success or failure. Partial observability is another important characterizes several games, where the full state is not visible in a single frame. This requires agents to integrate information over time or maintain memory of past observations. For example, in Breakout, even though the agent receives numerical information such as the position and speed of the ball, it does not know the complete future trajectory or any changes in speed due to multiple bounces. This necessitates mechanisms for temporal integration beyond simple reactive policies.

Generalization across games remains a key challenge [33] for the development of general-purpose learning systems. Although deep reinforcement learning has achieved superhuman performance on individual games, transferring of learned knowledge across different games remains difficult. Each game may require different visual features, temporal abstractions, and strategic reasoning, thereby testing the generality and robustness of learning algorithms.

The diversity and complexity of Atari games, combined with their standardized interface and computational tractability, make them ideal testbeds [32] for developing and evaluating artificial intelligence systems. They provide a middle

ground between simple toy problems and real-world applications, offering sufficient complexity to be challenging while remaining amenable to systematic experimentation. The characteristics are particularly relevant to the framework proposed in this thesis, which aims to extract interpretable symbolic knowledge from visual observations and use this knowledge to guide learning. The structured nature of 2D games, characterized by discrete objects obeying consistent rules, provides a suitable domain for testing the hypothesis that combining perceptual processing, symbolic reasoning, and reinforcement learning can lead to more efficient, interpretable, and generalizable AI systems.

3 State of the Art and Challenges

Although artificial intelligence has achieved significant successes in recent years, particularly through deep learning methodologies [1], fundamental challenges persist in the development of systems that are truly intelligent, interpretable, reproducible, and trustworthy. This chapter examines the current state of the art in AI, focusing on the limitations of purely neural approaches and the emerging paradigm of neuro-symbolic integration [34]. We explore the cognitive foundations that inspire alternative approaches, and the position of the framework proposed in this thesis within the broader landscape of contemporary AI research.

3.1 Challenges in Symbolic Interpretation

3.1.1 The Illusion of Understanding

Modern deep learning systems have achieved impressive performance on benchmark tasks, often matching or exceeding human capabilities in specific domains. Large language models such as GPT-4 and BERT demonstrate near-human accuracy in translation, question answering, and text generation, while deep neural networks excel in image classification and game-playing environments. These achievements often create the illusion of genuine understanding [7], masking fundamental limitations in how such systems process information. The impressive performance of modern AI systems has fuelled widespread optimism, with some interpretations suggesting that these models exhibit forms of intelligence or comprehension comparable to human cognition. However, this view has been critically challenged by several scholars. Bender et al. famously described large language models as *"stochastic parrots"*, emphasizing that their outputs are generated through sophisticated statistical pattern matching rather than through semantic understanding or intentional reasoning [35]. According to this critique, LLMs predict the most probable sequences of words based on training data distributions, without possessing an internal model of meaning or the world. A concrete manifestation of this limitation is the phenomenon of hallucinations, wherein language models confidently generate incorrect or nonsensical statements that are nonetheless statistically plausible [31]. While extensive research efforts aim to mitigate hallucinations through architectural improvements, or better datasets, some researchers argue that hallucinations are an intrinsic consequence of the probabilistic nature of these models [36]. This reinforces the distinction between surface-level linguistic competence and genuine understanding. Similar shortcomings emerge in applied contexts such as AI-assisted content moderation. Social media platforms including Facebook, Instagram, and X deploy

deep learning systems to detect hate speech, misinformation, and policy violations at scale. Despite their efficiency, these systems frequently struggle with contextual nuance, irony, and cultural variation, resulting in both over-censorship and under-enforcement. These failures highlight the inability of current AI systems to reason symbolically about intent, meaning, and social context. Melanie Mitchell has extensively analysed these limitations, particularly in her work *A Guide for Thinking Humans* [37]. She demonstrates how contemporary AI systems, spanning conversational agents, image generation models, and image captioning architectures, systematically fail when exposed to adversarial inputs specifically designed to be trivial for humans yet confusing for machines. These failures expose the lack of commonsense reasoning and conceptual grounding in both text-based and vision-based architectures. A paradigmatic example is the Winograd Schema Challenge [38], a benchmark designed to test an AI system's ability to resolve linguistic ambiguity using commonsense knowledge. Consider the sentence: *"The trophy did not fit in the suitcase because it was too big."* Humans effortlessly infer that *"it"* refers to the trophy, based on an intuitive understanding of physical size and spatial relationships. In contrast, AI models often perform poorly on such tasks, achieving accuracy rates below 70% , whereas humans typically score near 100%. This gap illustrates that neural models do not reason about physical properties or causality but rather rely on shallow textual correlations. Mitchell also examines DeepMind's game-playing agents, noting that despite their superhuman performance in specific games, these models remain extremely narrow. Each agent must be trained extensively on a single game, and the knowledge acquired does not transfer to even slightly modified environments. Some studies on the robustness of Deep Q-Networks (DQN) [39] show that minor visual perturbations, such as changing background colours or object positions, can cause complete failure. This stands in contrast to human performance, who grasp the abstract rules of a game and adapt to superficial changes, instead, these agents memorize pixel-level patterns and reward structures without forming a conceptual understanding of the environment.

The Lack of World Models

The limitations observed in language processing extend to other domains, including reinforcement learning and game-playing agents lead to a broader critique: modern AI systems lack an internal world model, like a structured representation of objects, relationships, causality, and dynamics necessary for reasoning and generalization [40] . Without such models, AI systems struggle to apply learned knowledge to novel contexts or perform commonsense reasoning reliably. Gary Marcus echoes this concern, arguing that progress in AI requires moving beyond narrow task optimization toward architectures that integrate structured knowledge and symbolic reasoning [41] . He advocates for hybrid and neuro-symbolic approaches as a promising path forward. As machine learning-based decision support systems become increasingly influential, it is essential for users to be able to critically assess the assistance they provide. A key requirement in this context is interpretability, which refers to the ability to understand and reason about a model's outputs [42]. Despite years of research efforts, progress in this area has been relatively limited. Highly successful models such as deep neural networks, while achieving near-human accuracy in various prediction and classification tasks, largely operate as black boxes [28].

They offer little transparency regarding how and why certain features are favoured over others during training, how patterns and correlations in the training data are internally represented, and how specific computational pathways transform raw inputs into final predictions. This limitation is particularly problematic as machine learning systems increasingly influence high-stakes decisions. In such contexts, interpretability becomes essential: users must be able to understand, question, and trust the outputs of AI systems. Despite extensive research, most high-performing deep neural networks remain opaque black boxes. The tension between predictive power and understanding becomes especially apparent when models are evaluated outside their training distribution. Image classifiers may rely on irrelevant background cues rather than object-defining features; language models may generate fluent but logically inconsistent text; and agents may exploit unintended shortcuts rather than learning the intended task structure. This behavior, often referred to as shortcut learning [43], reflects the tendency of neural networks to optimize for superficial correlations that satisfy training objectives without capturing underlying conceptual structure. Whereas these representations are statistically effective, they lack the compositionality, systematic generalization, and explanatory clarity that characterize human cognition. As a result, modern AI systems can appear highly intelligent while remaining fundamentally brittle, uninterpretable, and unable to truly understand the problems they solve.

3.1.2 The Black Box Problem

The illusion of understanding is closely tied to the black-box nature of deep neural networks. The opacity of deep neural networks poses significant challenges to trust, accountability, and scientific understanding [44, 45]. Modern deep neural network architectures are highly complex nonlinear statistical models with an exceptionally large number of parameters, making direct interpretation of their decision-making virtually impossible. This inherent complexity results in a lack of transparency, as these models are unable to provide explicit and interpretable explanations for the rationale underlying their decisions, which in turn has several important implications. In high-stakes applications such as medical diagnosis, criminal justice, or autonomous vehicles [2, 44], the demand for interpretability stems from the inherent incompleteness of problem formalization. In these contexts, producing an accurate prediction alone is insufficient, as it only partially addresses the original decision-making problem. The inability to explain how and why [42] a system arrives at a given outcome raises significant ethical and legal concerns, undermining trust and accountability. Consequently, stakeholders require not only reliable predictions, but also transparent explanations that clarify the reasoning processes behind those predictions, enabling informed evaluation, justification, and responsible use of automated systems. Despite significant advances in data modelling and learning algorithms, a clear gap remains between the construction of models and the extraction of meaningful knowledge from them. Machine learning models can be represented in many ways depending on the techniques used; however, true knowledge can only be claimed when these representations are understandable to humans. Knowledge extraction inherently involves human cognitive processes, which extend far beyond purely technical considerations. Without interpretability, models risk becoming ineffective, as their output cannot be properly assessed or trusted. For this reason,

interpretability should be regarded as a fundamental requirement [46] for machine learning approaches intended for real-world applications. Moreover, the opacity of neural networks significantly complicates their debugging and improvement. When a model fails, pinpointing the underlying causes and identifying effective corrective measures is often extremely challenging [47]. The absence of explicit and interpretable representations makes it difficult to determine whether errors arise from insufficient or biased training data [42], architectural limitations, or fundamental shortcomings in the learning process. Bias in both data and models is a particularly critical concern, as it can perpetuate harmful stereotypes and produce unfair outcomes [48]. Finally, the lack of transparency also limits the scientific insight that can be gained from successful AI systems. Although building effective models is undeniably valuable from an engineering perspective, understanding the principles that govern their behavior is essential for advancing the field of artificial intelligence. Black-box models may achieve high performance on specific tasks yet fail to provide generalizable insights that could guide the development of more robust, reliable, and capable systems. In response to these concerns, a growing body of research has sought to make sense of AI systems by focusing on their interpretability and explanatory power.

Making AI Transparent: Principles of Explainable Machine Learning

These limitations have spurred research into explainable artificial intelligence (XAI), which aims to make the behavior of machine learning systems more transparent and interpretable [46, 49]. The approaches to interpretable machine learning can be broadly categorized based on when interpretability is achieved. Post-hoc explanation methods attempt to interpret already trained models by constructing a secondary model that provides explanations for the original system. In contrast, inherently interpretable models are designed to be transparent from the outset [47], achieving interpretability by embedding explanatory mechanisms directly into their structures, such as decision trees and rule-based models. Additionally, hybrid approaches combine symbolic reasoning with neural learning to balance predictive performance and interpretability. Despite impressive empirical achievements, contemporary AI systems remain fundamentally limited in their capacity for genuine understanding [7], reasoning, and symbolic interpretation. Their reliance on statistical correlations, lack of internal world models, and opaque decision-making processes constrain their ability to generalize, explain, and reason about complex environments. Future research should therefore prioritize interpretability, commonsense reasoning, and the development of structured internal representations, shifting the focus from narrow benchmark performance toward robust, transparent, and truly intelligent systems.

3.1.3 Learning Paradigms and Explainable AI

The growing demand for explainability in artificial intelligence is not merely a response to ethical or regulatory concerns, but reflects a deeper limitation rooted in contemporary learning paradigms. Most state-of-the-art AI systems rely on statistical learning from large-scale datasets, optimizing performance objectives without explicitly representing meaning, causality, or structured knowledge. As a result,

the challenge of explainability cannot be separated from the way models learn and internally represent information. This section examines Explainable Artificial Intelligence (XAI) as a response to the black-box nature of modern machine learning, highlighting both its achievements and its fundamental limitations. In doing so, it argues that explainability is inseparable from learning mechanisms and representational choices, and that post-hoc explanations alone are insufficient to address the deeper issues of understanding and reasoning in AI systems.

Learning Without Understanding

Modern machine learning systems, particularly deep neural networks, learn by identifying statistical regularities in data rather than by constructing explicit models of the world. Learning is typically framed as the optimization of a loss function over large datasets, with success measured primarily in terms of predictive accuracy. While this paradigm has proven highly effective in narrow domains, it does not guarantee that the learned representations correspond to meaningful concepts or causal structures [3,37]. Consequently, models may rely on spurious correlations, dataset biases, or superficial cues that enable high performance during training but fail to generalize under distributional shifts, attributable to phenomena of this kind. This highlights the gap between statistical success and genuine conceptual understanding [43]. Moreover, because internal representations are distributed and opaque, it is difficult to determine what a model has truly learned, even when its predictions appear correct. This form of learning without understanding directly contributes to the interpretability problem. If a system lacks structured internal representations, explanations can only describe correlations in behavior rather than reasons grounded in knowledge or world models.

Explainable AI: Goals and Approaches

Explainable Artificial Intelligence (XAI) emerged in response to the opacity of high-performing machine learning models, particularly deep neural networks [44]. The primary goal of XAI is to enable humans to understand, trust, and effectively interact with AI systems by providing explanations of their behavior. Interpretability is especially critical in high-stakes domains such as healthcare, finance, law, and autonomous systems, where decisions must be justified, audited, and contested when necessary [50]. Beyond practical considerations, explainability is also essential for scientific understanding, debugging, and the iterative improvement of models. However, explainability is often treated as an external requirement imposed after learning has already occurred. This framing raises a fundamental question: can meaningful explanations be produced if the learning process itself does not involve the acquisition of interpretable knowledge?

Post-hoc Explainability

A large body of XAI research focuses on post-hoc explanation methods, which aim to interpret the behavior of trained black-box models without modifying their internal structure. Prominent examples include feature attribution techniques such as LIME

(i.e., Local Interpretable Model-Agnostic Explanations) [51] and SHAP (i.e., Shapley Additive Explanations) [52], which estimate the contribution of individual input features to a model's prediction. These approaches assist in identifying biases, facilitating debugging, and enhancing user confidence. Other approaches include saliency and attention-based methods, particularly in computer vision and NLP (i.e., natural language processing), which highlight regions of the input that most strongly influence model outputs [53]. For instance, in image classification, saliency maps can indicate the specific areas of an image that influenced a particular prediction. Surrogate models provide another strategy, approximating complex models with simpler, interpretable ones such as decision trees or linear regressions [54]. This method aids in interpreting the overall behavior of black-box models, though its accuracy in reflecting the original model may be constrained in highly nonlinear decision domains. While these methods offer valuable insights into model behavior, they remain fundamentally limited. Post-hoc explanations describe *"how"* a model behaves locally, but not *"why"* it behaves that way in terms of underlying concepts or causal reasoning. As a result, explanations may be unstable, incomplete, or misleading, especially when models operate in highly nonlinear or out-of-distribution settings [55].

Intrinsic Interpretability

In contrast to post-hoc approaches, intrinsically interpretable models aim to achieve transparency by design. These models, including decision trees, rule-based systems, linear models, and symbolic representations, provide explicit reasoning structures that can be directly inspected and understood by humans [42]. Intrinsic interpretability is often achieved by constraining model complexity or imposing structured representations. However, this typically comes at the cost of reduced flexibility or performance when compared to deep neural networks. This trade-off has historically limited the adoption of interpretable models in complex real-world tasks. Recent research has therefore explored hybrid approaches that integrate neural learning with symbolic or structured components, seeking to balance expressivity with transparency [5]. These approaches suggest that interpretability need not be an afterthought, but can emerge naturally from models that learn structured, compositional representations.

Why XAI is Not Enough

Despite significant progress, XAI methods cannot fully compensate for the absence of explicit internal world models. Explanations that operate solely at the level of input-output correlations fail to capture higher-level concepts such as object identity, causality, goals, and physical constraints. Without these structures, explanations remain shallow and task specific. This limitation has led several researchers to argue that explainability must be addressed at the level of representation and learning, rather than through post-hoc analysis alone [56]. In other words, a system can only explain what it has explicitly represented. From this perspective, the challenge of explainable AI is inseparable from broader questions about how knowledge is acquired, structured, and used for reasoning. This insight motivates a shift toward

cognitively inspired approaches that emphasize structured representations, inductive biases, and internal world models topics that will be explored in the following section.

3.2 Making Sense of AI: Toward Explainability

3.2.1 Cognitive Foundations

The limitations of purely data-driven approaches have led researchers to reconsider insights from cognitive science and developmental psychology. Understanding how humans acquire knowledge, form concepts, and reason about the world may provide valuable principles for building more effective and interpretable AI systems.

3.2.2 Human World Modelling and the Origins of Structured Cognition

Human intelligence is characterized by the ability to form internal models of the external world; structured representations that capture the objects, the relationships, and the causal mechanisms that govern observable phenomena. These models enable prediction, planning, and reasoning beyond immediate perceptual experience. The question of how these models is acquired, has been central to cognitive science and developmental psychology. The Core Knowledge theory, developed primarily through research by Elizabeth Spelke, Renée Baillargeon [57], and colleagues, proposes that human cognition is founded on a small set of innate knowledge systems that emerge early in development and provide the foundation for subsequent learning. These core systems are domain-specific, addressing fundamental aspects of the physical and social world, and are hypothesized to be evolutionarily ancient, shared with other species. The existence of core knowledge systems suggests that learning is not a completely domain-general process of association formation, but rather involves specialized mechanisms tuned to specific aspects of the environment. This perspective contrasts with blank slate views of learning and suggests that effective AI systems may benefit from incorporating analogous inductive biases.

3.2.3 Core Knowledge Theory: Cognitive Principles and Psychological Foundations

Core knowledge refers to a set of foundational cognitive systems that emerge remarkably early in human development, often within the first year of life. These systems are not merely innate facts; rather, they consist of organizational principles that guide infants' attention, interpretation of experience, and learning [57]. These systems provide initial structures for interpreting experience in specific domains. In other words, they provide a structured framework that allows infants to extract meaning and patterns before formal instruction or extensive experience. Spelke and Kinzler highlight that traditional views of human cognition have often polarized between two extremes. On one hand, some scholars have described the mind as a general-purpose learning system, capable of absorbing and adapting to

any kind of experience. On the other hand, evolutionary psychologists have argued for a large number of domain-specific systems, each evolved to handle a particular cognitive task. Core knowledge represents a middle ground: humans appear to possess a limited number of specialized, but flexible, cognitive systems, each tuned to certain types of information, which together form the foundation for more complex thought and learning. One of the most extensively studied core systems is object representation. From the earliest months of life, infants show an understanding that physical objects behave in predictable ways. They expect objects to be cohesive, moving as unified wholes; continuous, persisting even when hidden from view, and solid, unable to pass through one another. Violations of these expectations, such as an object seeming to vanish into thin air, elicit surprise and heightened attention, suggesting that infants are actively engaging in reasoning about the physical world. This system also has limits: infants can typically track only a few objects at a time, usually around three or four, a constraint that persists into adulthood when attention is stretched [58]. Interestingly, these principles appear to be universal, observable even in remote cultural groups like the Pirahã of the Amazon, whose language and counting systems differ dramatically from those commonly studied in western contexts [59]. Alongside object representation, infants develop a number sense that allows them to perceive, compare, and manipulate quantities. This system is supported by two complementary mechanisms. The Approximate Number System (ANS) allows infants to estimate quantities, with precision decreasing as numbers grow larger, and the object-tracking system enables exact representation of small sets, typically up to three or four items [58]. Together, these systems form the basis for early arithmetic reasoning and provide a foundation for later formal mathematical learning. Equally important is the spatial system, which allows infants to encode information about locations, distances, and geometric relationships. This knowledge supports navigation, understanding of containment and support, and the prediction of where objects might be in space. Spatial reasoning is not just a precursor to interacting with the physical world, it underlies more sophisticated cognitive processes later in life, including problem-solving and abstract thinking. Finally, infants possess a system for agent representation, which enables them to understand goal-directed behavior and social interactions [58]. Unlike objects, agents do not need to be cohesive or move continuously to be recognized as entities. Instead, infants interpret their actions as driven by goals, often expecting that agents will act efficiently to achieve desired outcomes. They are sensitive to contingent and reciprocal interactions, as well as social cues such as gaze direction, which allow them to anticipate intentions and understand others' actions. This system forms the basis of early social cognition and plays a role in moral reasoning and cooperative behavior throughout life. Across diverse domains, core knowledge systems share several defining characteristics. They emerge early in development, operate on specialized types of entities and their interrelations, and provide a foundational framework that supports the acquisition of increasingly complex knowledge [57]. When the principles underlying these systems are violated, they elicit surprise and heightened attention, thereby promoting enhanced learning and cognitive engagement. Importantly, core knowledge should not be understood as a set of static, preloaded facts; rather, it consists of flexible principles that guide attention, interpretation, and learning from experience. Moreover, studying these systems highlights their evolutionary roots and universality, offering insights not only into human development but also into

the ways cognition is shaped by both biology and experience.

3.2.4 Definition and Relevance for World Understanding

The relevance of Core Knowledge theory for artificial intelligence extends beyond learning efficiency and directly concerns the problem of explainability. The systems that reason about the world through structured internal models, composed of objects, relations, and causal mechanisms, are not only better equipped to generalize, but also to produce explanations that are meaningful to human users. Core Knowledge suggests that effective world understanding depends on inductive biases that privilege certain entities and relations as explanatory primitives, rather than treating all aspects of the environment as equally relevant, learning from scratch through domain-general mechanisms, or modelling perception as an undifferentiated stream of data [57]. In the context of understanding dynamic environments such as video games, core knowledge principles suggest several design principles that directly support interpretability and explanation. In particular, the object persistence and identity enable explanations in terms of enduring entities that interact over time, rather than transient patterns in sensory input. Physical causality supports accounts of system behavior based on cause-effect relations, allowing predictions and outcomes to be justified rather than merely observed. Similarly, representing agents as goal-directed entities provides a basis for interpreting and explaining actions in intentional terms, which is essential for understanding behavior in interactive environments [57]. Finally, compositional structure allows complex scenes and events to be explained as combinations of simpler, interpretable components and relations, rather than as opaque high-dimensional states [7]. By adopting object-centric and relational representations inspired by Core Knowledge, an artificial system could construct explicit internal world models that support not only prediction and control, but also explanation and counterfactual reasoning [60]. These considerations motivate the architectural choices in the framework proposed in this thesis, which emphasizes the extraction of structured, object-based representations from visual inputs and the use of symbolic or relational reasoning mechanisms to model environment dynamics in an interpretable manner.

3.2.5 Cognitive Models and Symbolic Representation

The integration of cognitive principles with symbolic AI offers a path toward systems that combine the learning capabilities of neural networks with the interpretability and reasoning power of symbolic representations. The cognitive architectures such as ACT-R [61] and SOAR [62] have demonstrated the value of hybrid systems that integrate symbolic reasoning with sub-symbolic processing, enabling more efficient learning and generalization. Intuitive physics research in cognitive science has revealed that humans possess sophisticated, often implicit, models of physical phenomena that enable prediction and reasoning about object motion, stability, and interactions. Incorporating similar models into AI systems could enable more sample-efficient learning and better generalization, as systems would not need to learn fundamental physical principles from scratch. The causal reasoning is central to human intelligence, enabling understanding of why events occur and prediction

of the consequences of interventions. Causal models, represented graphically or through structured probabilistic models, provide a framework for integrating symbolic causal knowledge with probabilistic reasoning. The systems that can learn and reason with causal models may achieve more robust generalization and interpretability than purely associative systems. Additionally, compositional semantics suggests that effective knowledge representations should construct complex concepts from simpler primitives. Neural-symbolic systems that learn such compositional representations can better capture the systematic structure of domains, thereby facilitating interpretation and explanation [5].

In summary, developing interpretable AI systems requires methods to extract structured, symbolic representations from raw sensory data while maintaining the flexibility and learning capability of neural approaches. This requires bridging the gap between sub-symbolic distributed representations learned by neural networks and symbolic, compositional representations suitable for reasoning and explanation.

3.3 Positioning of the Proposed Framework

The framework developed in this thesis is positioned at the intersection of several research directions including visual perception in dynamic environments, symbolic knowledge extraction, cognitive modelling, and reinforcement learning. It addresses key limitations of current approaches while building on their strengths. In relation to deep reinforcement learning, the framework addresses two well-known limitations of end-to-end neural approaches: their high data requirements and the opacity of the representations they produce. Although methods such as Deep Q-Networks have achieved strong results in visually complex domains, they typically learn implicit, task-specific representations that provide little explicit insight into the structure of the environment.

The proposed framework departs from standard paradigm by decoupling visual perception from policy learning. Rather than relying on a reinforcement learning agent to simultaneously learn perception, environment structure, and control, perceptual processing is handled by dedicated neural components whose outputs are subsequently transformed into explicit symbolic representations. This shift moves the emphasis away from end-to-end policy optimization and toward structured environment understanding, enabling interpretability and reasoning while preserving the advantages of neural visual processing. In this positioning, deep reinforcement learning is not replaced but reframed: the DQN agent operates on a structured environment representation rather than raw visual input, allowing learning to focus on decision-making and exploration instead of implicit model acquisition. From the perspective of neuro-symbolic AI, the framework reinforces the view that symbolic representations remain valuable when they are grounded in perception rather than manually specified. Unlike approaches that impose symbolic constraints on neural models or use neural networks as auxiliary components within symbolic pipelines, this work emphasizes the extraction of symbolic structures directly from visual experience. In doing so, it consolidates the role of neural networks as perceptual front ends while extending neuro-symbolic integration toward the autonomous discovery of environment rules. The framework embodies core principles of neuro-symbolic

integration by combining neural visual processing with symbolic reasoning. This model serves as an interpretable intermediate representation that bridges perception and action. The framework draws inspiration from Core Knowledge theory and cognitive principles about object representation, physical causality, and compositional structure. However, unlike full cognitive architectures that aim to model the breadth of human cognition, this framework focuses on a specific task; understanding and acting in dynamic visual environments and implements only those cognitive principles most relevant to this domain. The proposed framework makes several distinctive contributions. It demonstrates a complete pipeline from raw visual input to symbolic knowledge extraction, to environment reconstruction to policy learning. It shows how cognitive principles can inform the design of AI systems for dynamic environments. And it offers a concrete instantiation of neuro-symbolic integration in the domain of game playing.

Overall, the proposed framework is not intended as a replacement for existing reinforcement learning systems or other established approaches, but as a model designed to overcome key limitations of previous approaches by emphasizing interpretability and explicit knowledge representation over end-to-end optimization. It contributes to the state-of-the-art by showing how symbolic knowledge can be extracted from visual observations and structured in a form suitable for reasoning and future interaction. The framework offers a pathway toward systems that are both effective in dynamic environments and understandable, highlighting the potential of neuro-symbolic approaches to enhance reasoning, exploration, and future interaction. The following chapter describes in detail the design and implementation of this framework, explaining the architectural choices, algorithms, and integration mechanisms that enable the extraction of interpretable symbolic knowledge from video streams and its use in guiding reinforcement learning in reconstructed environments.

Although, the framework demonstrates a complete conceptual pipeline from raw visual input to symbolic knowledge extraction and policy learning, the reconstructed environment is currently defined at the design level. It serves as an intermediate symbolic representation intended to support reasoning and learning, with a full interactive implementation deferred to future work.

4 Proposed Framework

The aim of this chapter is to present the framework developed for this thesis, with the intent of describing the logical processes that have guided its realization as well as describing its current implementation. The proposed system aims to integrate techniques as visual perception, symbolic reasoning and reinforcement learning to infer, specifically within the context dynamic environments like Atari-like video games, symbolic and interpretable knowledge. The use of techniques such as image segmentation allows the system to develop structural representations of the environment. The result produced is used by the cognitive framework with the intent of extracting symbolic and interpretable knowledge and model of the game environment. This knowledge is then intended to be used to build an environment that leverages the learned knowledge in which a DQN develops gaming skills. During the search for a framework that allows good game performance, overcoming some limitations of approaches entirely based on neural approaches, the focus was on a system that exploits cognition for building models of the internal world, and it is able to face a complete game based on an explicit and manipulable understanding of the environment. The following sections describe more in detail, the various components, workflow, and principles that make this integration an alternative and complementary solution to the modern dominant approaches in artificial intelligence.

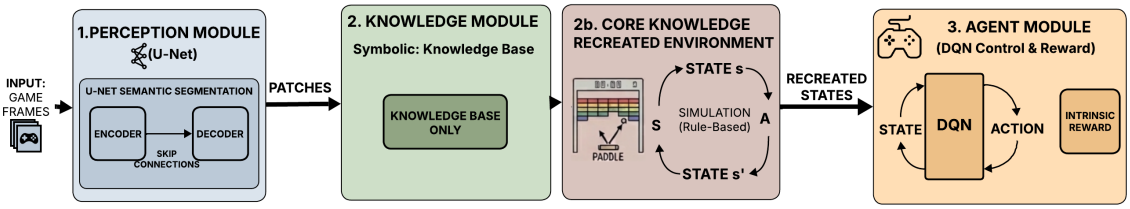

Figure 4.1: The figure illustrates the overall architecture of the system, which consists of three main modules. The Perception Module uses a U-Net for semantic segmentation of raw game frames, extracting structured patches via an encoder-decoder architecture with skip connections. These patches are passed to the Knowledge Module, which builds a symbolic knowledge base and recreates the game environment through a rule-based simulation, producing symbolic states. Finally, the Agent Module will receive these reconstructed states and uses a DQN to select actions, guided by an intrinsic reward signal.

4.1 Conceptual Foundations

This section presents the development of the thesis' themes and objectives, starting with the definition of the fundamental design principles that guided the development

of the work.

4.1.1 Evolution of the Idea

The initial idea for this work originated from a preliminary draft of a research paper that proposed an evolutionary approach to modelling structured environments, drawing inspiration from the principles of Core Knowledge theory. The goal was to obtain human-interpretable representations of dynamic environments, such as simple 2D video games, by combining a preliminary visual processing phase with evolutionary learning mechanisms. Before the application of the evolutionary model, the framework involves the use of a semantic network in charge of extracting the fundamental components of the game environment. This network identifies relevant objects, relationships, and properties within visual patches, providing a first structural organization of perceptual data. This preliminary representation constitutes the conceptual basis from which the evolutionary system starts to build and optimize rules and symbolic categories. The original hypothesis was that by developing a system based on innate cognitive rules, it was possible to construct a symbolic representation of an environment without relying too much on data-driven learning. In this direction, an evolutionary algorithm has been developed those deals with optimizing symbolic representation through a compact sets of object classes and behavioural rules. The encoded heuristics are used to provide a useful metric for this process. The evolutionary algorithm acted as a search procedure to refining, compressing the above structures to provide a minimum explanatory representation of the environment. Specifically, an incremental knowledge construction was sought with the aim of emphasizing learning based on interaction and the observation of new behaviours. In this direction, a bottom-up perceptual basis has developed, thanks to sensory data and a top-down guide in which inferred knowledge informs the interpretation of new observations. As a result of this evolutionary process, object classes and behavioral rules are not identified in isolation but rather emerge from the integration of perceptual inputs with the internal model. Once inferred, these rules provide a symbolic description of the environment The core idea is to recreate the analysed game environment, based on the extracted rules and the previously observed states, and to train a DQN in this reconstructed environment. Although the framework is still at an early stage and limited in scope, it successfully generates a coherent symbolic representation of the game environment from low-level visual inputs, thereby offering a concrete proof of concept for the proposed research direction. The final goal is to develop a system that is highly adaptable to a wide range of 2D game environments, capable of generalizing across the diverse dynamics and structural properties that characterize such domains.

4.2 Proposed Implementation

This section presents the current state of the proposed framework, developed according to the fundamental design principles introduced earlier. Although still limited in terms of scope, functionality, and generalization, this version introduces a method that enables a symbolic system to extract interpretable representations from

a dynamic environment using only low-level perceptual inputs. The development is demonstrated in a simplified scenario, a 2D game environment that evolves frame by frame. These frames are processed initially by a semantic network and then through a pipeline consisting of object tracking, rule inference, and class generalization. Using the principles of Core Knowledge as guidance, the system developed explicit heuristics to generate symbolic knowledge about the identity and behavior of objects, which are then used as a guide in following iterations. The decision to structure the framework as a neural network for visual preprocessing followed by a Core Knowledge. The inspired module for interpreting segmented elements is motivated by the limitations of conventional deep learning in temporal reasoning. Although deep networks excel at pattern recognition within individual frames, they struggle to capture temporal dependencies critical for action-driven decision-making [63]. By combining neural perception with a symbolic reasoning layer, the framework leverages the strengths of deep learning while producing interpretable, structured representations that support temporal reasoning and rule extraction, addressing key limitations of purely neural approaches. The preliminary implementation used yields promising results, showing that the system can construct an interpretable, structured internal representation of the environment, track persistent objects, deduce interactions, and classify basic object types. This first version is not intended as a definitive solution, but as a minimal and interpretable foundation upon which develop more robust, generalizable and adaptive components. The following sections examine the key framework's modules in detail, focusing on their purpose, structure and limitations.

4.2.1 Segmentation

The environment that the system needs to understand is represented as a sequence of frames extracted from the game environment. These frames have to be processed to obtain a set of non-overlapping visual units called "patches". These patches are the primary perceptual input of the system: minimal, anonymous representations of visual elements that can correspond to objects or parts of objects within the environment. The framework relies on a segmentation algorithm that segments the images of game frames generated by a simulated environment. The result of the segmentation, appropriately coded, are used to extract the patches. As a result, the system encounters some of the challenges typical of real visual perception, such as noise, occlusion or ambiguity.

Network Architecture

The segmentation network is a U-Net-based convolutional neural network, specifically designed for multi-class semantic segmentation.

The network follows a symmetric encoder-decoder design: the encoder progressively extracts hierarchical feature representations through convolutional blocks and downsampling, while the decoder reconstructs spatial resolution using transposed convolutions and skip connections. The skip connections between corresponding encoder and decoder layers enable the network to recover fine-grained spatial details lost during downsampling, allowing for precise localization of object boundaries

while maintaining contextual awareness (see Figure 4.2).

The encoder consists of three convolutional blocks, each composed of two consecutive 3×3 convolutional layers followed by ReLU activation functions. These blocks progressively increase the number of feature channels (64, 128, and 256) while reducing the spatial resolution through max-pooling operations with a stride of 2. This design allows the network to capture increasingly abstract representations of the input image, as the spatial resolution decreases while the number of feature channels increases. At the bottleneck, a deeper convolutional block expands the feature dimensionality to 512 channels, enabling the network to learn the most abstract and global features of the image.

The decoder mirrors the encoder structure and employs transposed convolutions for upsampling. At each decoding stage, the upsampled feature maps are concatenated with the corresponding encoder outputs through skip connections. This mechanism preserves spatial information and improves boundary localization in the predicted segmentation masks. Before concatenation, the encoder feature maps are spatially cropped in order to match the size of the decoder features.

The final output layer is a 1×1 convolution that maps the decoded feature maps to the desired number of semantic classes. Since the convolutional operations reduce the spatial dimensions of the feature maps, a bilinear interpolation step is applied to resize the output to the original input resolution. The network outputs dense per-pixel class scores for each semantic category. During inference, the final segmentation mask is obtained by applying an *argmax* operation across the class dimension to assign each pixel to the class with the highest predicted score.

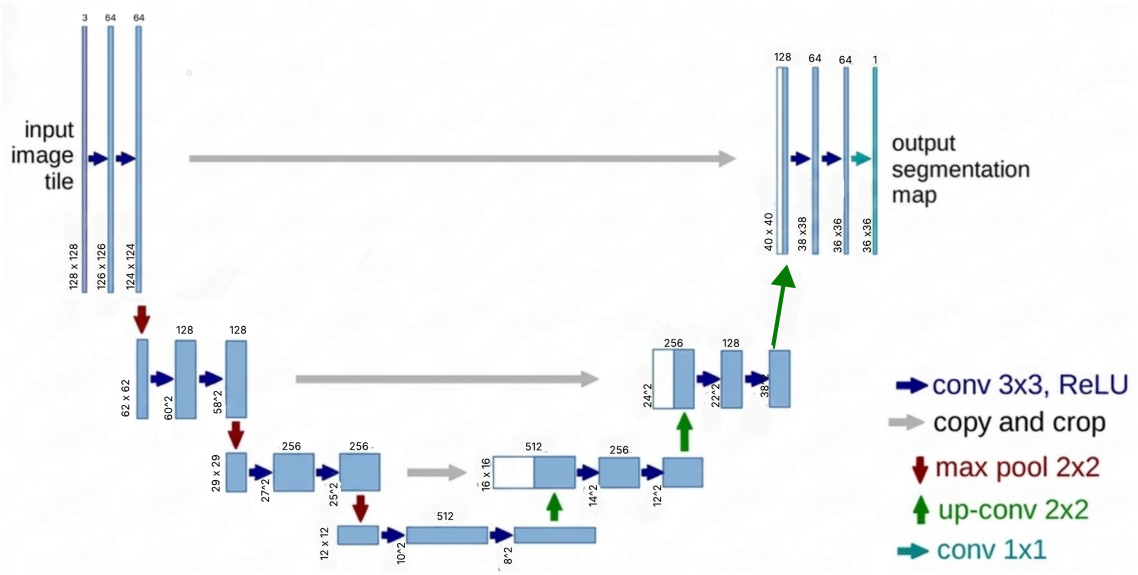

Figure 4.2: The figure shows the U-Net architecture of the proposed framework. The blue boxes correspond to a multi-channel feature map, indeed the white boxes represent the copied feature map.

4.2.2 Symbolic System as Knowledge Construction Module

This chapter describes a system module that was adopted and integrated within the overall framework, rather than being designed and implemented entirely from scratch. The module was incorporated as a pre-existing conceptual and functional component, with only minimal modifications introduced to improve coherence and interoperability with the preceding and subsequent modules of the architecture. Its primary role is to explore a cognitive-inspired alternative for building internal world models specifically, it is a symbolic system that derives interpretable representations of a dynamic environment from low-level perceptual inputs.

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

Patch Extraction

The environment is represented as a sequence of discrete frames extracted from a game-like simulation. Each frame is decomposed into a set of non-overlapping visual units, called patches, which serve as the system's primary perceptual input. These patches are anonymous and minimal representations of visual elements that may correspond to objects or object parts. In the current framework, patch extraction is not performed through direct visual reasoning within the agent. Nevertheless, this component has undergone a slight improvement over the originally proposed model, as patches are no longer obtained directly from the simulated environment.

Instead, raw game frames are processed by the semantic segmentation module, whose outputs are subsequently post-processed by a reconstruction script. This script converts segmented objects into structured attributes, such as position and shape, which are then used to construct simplified patch representations. This component effectively serves as a placeholder perceptual pipeline. Its use reflects the framework's primary focus on symbolic modeling and knowledge representation rather than perceptual learning.

The framework's next development phase will add a visual segmentation module, allowing patches to be derived from raw pixel inputs under the guidance previously acquired knowledge. However, this process is vital for examining how the system handles real-world perceptual challenges.

Object Formation

The object formation module is responsible for reconstructing persistent entities over time by associating sequences of patches across frames. Since the system receives no explicit information about object identity or permanence, this module plays a crucial role in establishing continuity and coherence in the environment representation. Object tracking is performed using a set of interpretable heuristics based on spatial and temporal coherence, as well as predictions derived from previously inferred knowledge encoded in object classes. Each object is associated with a class that defines behavioral constraints and predictive rules. Incoming patches are matched to existing objects according to proximity, shape similarity, and consistency with expected behavior. Patch-to-object assignments are categorized as perfect, possi-

ble or no-direct, allowing the system to represent both expected and unexpected changes. Due to the sequential nature of frame processing, ambiguities frequently arise. In order to manage this uncertainty, the system maintains multiple competing interpretations in parallel, each corresponding to a distinct configuration of objects and inferred behaviors. Objects are modeled as episodic reconstructions tied to the current observation sequence, while object classes and behavioral rules provide the basis for generalizable knowledge. This module, as integrated, establishes the foundational link between perceptual input and symbolic representation.

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. This module operates on data produced during object tracking, it identifies recurring correlations between observed events and unexplained property variations. When consistent patterns are detected, the system generates generalized symbolic rules that capture these regularities in a reusable and interpretable form. These rules are stored in a growing knowledge base and may later inform predictions or constrain future interpretations. In its current form, rule inference relies on correlational analysis and does not yet implement causal or counterfactual reasoning. Despite this limitation, the module represents an important step toward structured, interpretable symbolic knowledge.

Pruning

Throughout the processing pipeline, the system may generate multiple individuals, each representing a distinct interpretation of the environment. In order to prevent uncontrolled growth in the number of maintained interpretations, a pruning mechanism evaluates individuals based on the coherence and simplicity of their explanations. Scoring criteria include the number of unexplained changes, adherence to object class rules, and overall explanatory complexity. Interpretations that achieve the highest scores are retained, while others are scheduled for removal after a short grace period, allowing confirmation from subsequent frames. This pruning strategy balances flexibility and computational efficiency, ensuring that the system can explore alternative hypotheses without incurring prohibitive computational costs.

Generalization

A central objective of the framework is the ability to generalize across different game environments without extensive retraining. Generalization is achieved through the formation of object classes that abstract away from individual episodic instances and encode stable structural and behavioral regularities. These classes act as flexible blueprints that support recognition and interpretation of similar objects in new environments. Learned classes can guide object formation and prediction in unfamiliar scenarios, with class assignments emerging gradually as consistent behaviors are observed. When new observations systematically violate existing class rules, the system can adapt by refining or splitting classes to preserve internal consistency.

This mechanism enables incremental refinement of symbolic knowledge while maintaining interpretability and supporting transfer learning grounded in explicit object classes and rules.

4.2.3 From Symbolic Rules to a Playable Environment

The module presented in the previous section is capable of extracting meaningful symbolic representations from raw perceptual input. The system successfully identifies object classes, tracks their properties over time, and infers a set of interaction rules that closely match the dynamics encoded in the original environments. These rules capture not only static object attributes, but also conditional behaviors triggered by events such as contacts, disappearances, and user inputs, providing an explicit and interpretable description of the environment's underlying structure. In addition, the results obtained in previously executed experimental evaluation suggest that the inferred rules are stable, transferable across episodes, and robust to moderate environmental variations, reinforcing the hypothesis that they could eventually support the automatic generation of executable environments. From a conceptual standpoint, this inferred symbolic representation constitutes a natural candidate for the construction of a fully autonomous game environment. In principle, the object classes and behavioral rules discovered by the system could be compiled into an executable world model, allowing an agent to interact with an environment generated directly from learned knowledge. This capability would represent a crucial step toward closing the loop between perception, knowledge acquisition, and action. However, in the current stage of this work, the automatic synthesis of a playable environment from symbolic rules has not yet been implemented. Despite the inferred rules are sufficiently expressive to describe the observed dynamics, translating them into a complete and robust simulation raises additional challenges, including conflict resolution between rules, temporal coordination, and the handling of incompletely specified or ambiguous behaviors. For this reason, the experiments presented in this thesis adopt a simulated game environment as a proxy for the rule-generated world. A Deep Q-Network (DQN) agent is then trained to interact with this simulated environment, using state descriptions and event-based rewards derived from the system's representations. This approach allows the investigation of how an agent can learn effective control policies over an interpretable and structured state space, without relying on dense, task-specific reward functions. Crucially, the use of a simulated environment does not undermine the long-term objectives of the framework. Rather, it serves as a controlled experimental platform that enables the validation of the proposed learning mechanisms. In this sense, the simulated game world employed in this thesis should be regarded as a placeholder for future developments, in which symbolic rules will directly give rise to the environments in which agents learn and act.

4.2.4 Generic Environment as a Transitional Representation

Generic Environment Design

To ensure domain independence, scalability, and reusability, a generic environment was implemented. The primary objective of this design is to avoid hard-coded assumptions about the environment structure, semantics, or task objectives, allowing the same learning architecture to be deployed across heterogeneous domains with minimal modification. The environment operates under the assumption that the underlying simulation exposes a set of observable attributes. Rather than relying on manually defined state representations, the system employs an automatic state discovery mechanism based on reflection. All observable numerical and boolean attributes of the simulation object are identified dynamically and treated as raw features. These attributes collectively define the agent's observation space, without encoding any semantic meaning such as object identity, spatial relations, or goals. By exposing only unstructured numerical signals, the agent is forced to infer regularities, correlations, and causal dependencies directly from experience. As a result, the learning process remains agnostic to the specific domain and does not depend on expert-designed representations.

Generic event detection and causal abstraction

In parallel with state extraction, the environment implements a fully generic event detection mechanism. Events are defined as significant changes in observable attributes between consecutive time steps. With this strategy is not necessary to impose prior classification, symbolic labeling, or domain knowledge: any attribute variation exceeding absolute or relative thresholds is considered an event. This design reflects the assumption that interaction-relevant structure emerges from change, rather than from static state descriptions. Both discrete transitions (e.g., boolean flips or integer changes) and continuous dynamics (e.g., velocity or position variations) are captured uniformly. Each event is treated as equally informative at detection time, deferring any notion of importance or hierarchy to later stages of learning. Detected events are temporally organized into causal chains, under the hypothesis that clusters of events occurring within short temporal windows correspond to structured interactions between the agent and the environment. Longer chains are interpreted as evidence of richer causal dynamics, and they form the basis of the intrinsic reward signal described below.

4.2.5 Deep Q-Network in a Domain-Agnostic Environment

A Deep Q-Network (DQN) agent is trained to interact with the generic environment described above. The agent receives observations derived exclusively from automatically extracted numerical attributes [1], without relying on hand-crafted object semantics or task-specific encodings. Consequently, the DQN operates on a flat and unstructured state representation, learning control policies directly over raw features.

The underlying goal is to develop an agent capable of adapting to a diverse set of environments without being explicitly provided with environment-specific reward functions. The training follows a standard off-policy DQN framework [28], incorporating experience replay, target network stabilization, gradient clipping, and ε-greedy exploration. However, unlike classical reinforcement learning setups, the agent is not guided by an externally defined reward function tied to task success or failure.

Reward

The underlying goal is to develop an agent capable of adapting to a diverse set of environments without being explicitly provided with environment-specific reward functions. In order to achieve this, rather than relying on sparse or task-specific rewards, the agent is trained with a composite intrinsic reward that reflects the causal structure of the environment. This design choice is inspired by the extensive literature on intrinsic motivation in reinforcement learning, which proposes internal reward signals to drive exploration and autonomous skill acquisition [64]. The composite reward structure combines multiple components that encourage meaningful interaction with the environment. Instead of relying on task-specific signals, the agent is rewarded based on the effects of its actions, including event-driven changes and measures of causal impact.

Event-centered reward

A key innovation of the proposed framework is the adoption of an event-centered intrinsic reward structure. Instead of rewarding goal completion, the agent is rewarded for generating meaningful changes in the environment. This perspective aligns with approaches that replace explicit objectives with measures of novelty, surprise, or interaction richness [65]. Detected events are grouped into causal chains, and rewards are assigned as a function of chain length using a power-law formulation. This choice reflects the intuition that behaviors producing extended sequences of effects are more informative and structurally significant than isolated changes. Importantly, no assumption is made about which events are *"good"* or *"bad"*; reward emerges solely from interaction richness.

Causal impact bonus

To further encourage intentional behavior, the framework introduces a causal impact bonus. For each action, the observed state transition is compared with a counterfactual transition produced by a no-op baseline. The difference between these transitions provides an estimate of how strongly the agent's action influenced the environment. This mechanism rewards actions with a clear causal footprint, discouraging passive or redundant behaviors and promoting actions that actively shape the environment's dynamics. The notion of causality here is purely operational and does not rely on symbolic causal models.

Curiosity-driven bonus

Intrinsic motivation is further reinforced through a curiosity mechanism based on predictive modelling. A learned forward model attempts to predict the next state [66] given the current state and action. The prediction error of this model serves as an intrinsic reward signal, encouraging the agent to explore regions of the state space that are novel, complex, or difficult to predict. This formulation aligns with the view that learning progress and model error are proxies for informational gain. The forward model is trained online, ensuring that curiosity adapts dynamically as the agent's understanding of the environment improves.

Evaluation Perspective

Due to no extrinsic goal is defined, performance is not evaluated in terms of task completion. Instead, evaluation focuses on survival time, behavioral stability, and the agent's ability to maintain coherent interaction with the environment over extended periods. This choice reflects the underlying philosophy of the framework: learning is assessed by the richness and persistence of interaction rather than by predefined success criteria. In this implementation, temporal reasoning emerges through several mechanisms. The discount factor encourages long-term planning by valuing future rewards, while the replay buffer allows the agent to learn from past sequences of interactions. Additionally, a forward dynamic model explicitly predicts future states, enabling the agent to anticipate the consequences of its actions. The causal event tracker further reinforces multi-step strategies by rewarding chains of temporally connected events. Together, these components allow the agent to handle delayed rewards, learn action timing, and develop long-term strategies.

5 Implementation

This chapter describes the concrete implementation of the proposed framework, detailing the end-to-end processing pipeline, the architectural design choices, and the implementation of each system module. The framework integrates visual perception, symbolic reasoning, and reinforcement learning into a unified and modular architecture capable of discovering structure, causality, and control strategies from raw sensory data, validated in a dynamic game environment.

5.1 End-to-End Pipeline

The implemented system follows a fully automated end-to-end pipeline that processes raw visual observations and transforms them into symbolic knowledge and action policies. The pipeline is designed to progressively increase the abstraction level of information, moving from pixels to semantic objects, then from symbolic events to decision-making strategies (see Figure 4.1).
The pipeline operates through the following stages:

- **Visual input acquisition**
  Frames are captured from the environment or loaded from logged gameplay sessions, each frames represents the full observable state of the environment at a given time step.
- **Image preprocessing**
  Input images are resized to a fixed spatial resolution and normalized using standard ImageNet statistics [67]. This ensures numerical stability and compatibility with convolutional neural networks pretrained or inspired by standard vision architectures.
- **Semantic segmentation**
  A deep convolutional model assigns a semantic class to each pixel, producing dense segmentation masks that identify relevant entities such as the ball, paddle, bricks, and background (see Figure 5.1a).
- **Patch extraction and object tracking**
  Segmented regions are converted into patches and tracked across consecutive frames, forming persistent object hypotheses with temporal continuity.
- **Symbolic abstraction and reasoning**
  Visual objects are mapped into symbolic entities described by properties,

events, and inferred rules. A population-based heuristic process evaluates alternative symbolic explanations and selects the most coherent ones (see Figure 5.1b).

- **Reinforcement learning**
  A reinforcement learning agent interacts with the environment through a generic symbolic interface. The rewards are derived from intrinsic measures of causality, novelty, and event generation (see Figure 5.1c).

- **Logging and persistence**
  Intermediate representations, learned models, and symbolic populations are saved, enabling reproducibility, debugging, and incremental refinement.

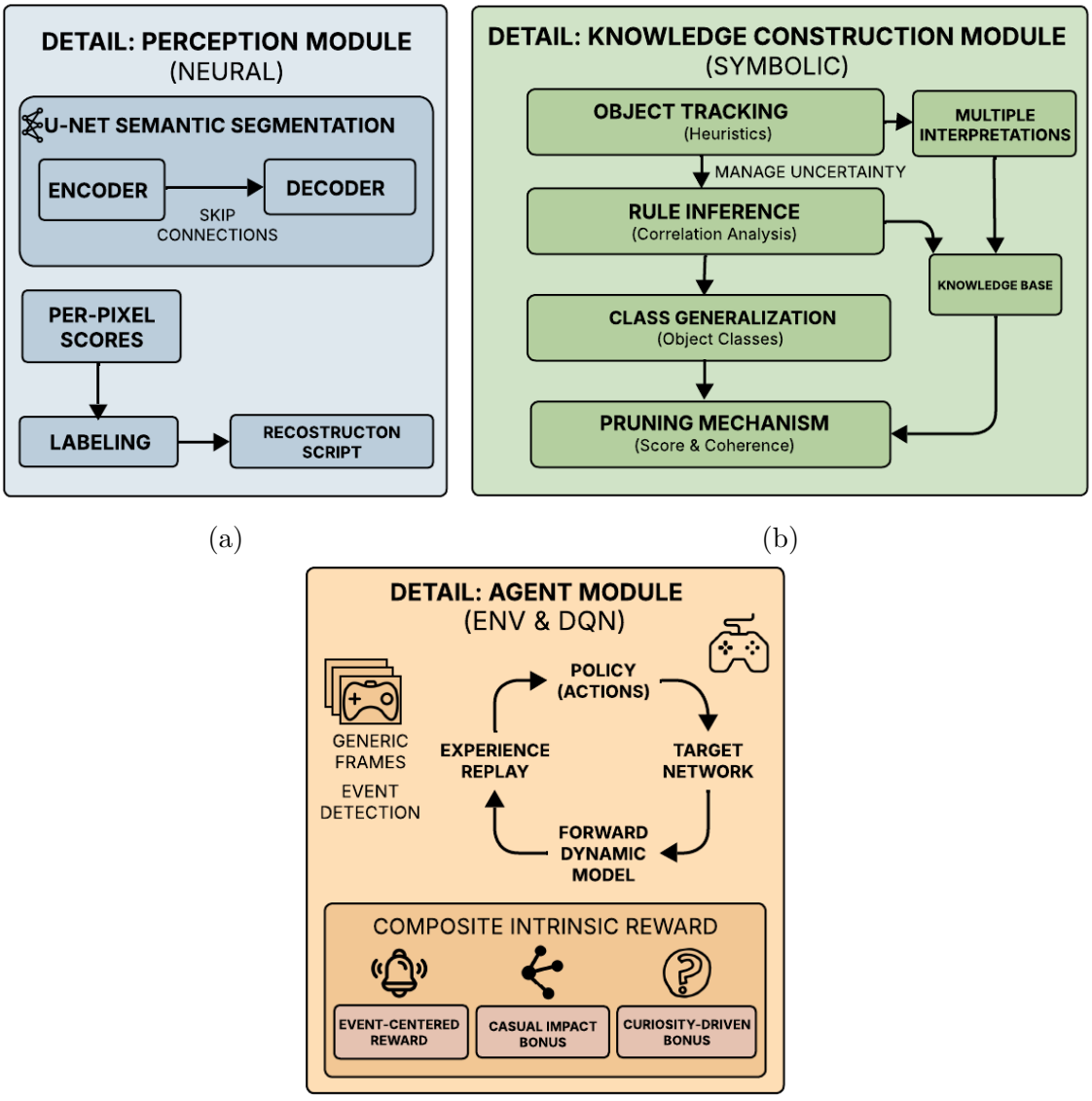

Figure 5.1: The framework consists of (a) a neural perception layer for raw data processing, (b) a symbolic module for knowledge base construction, and (c) a DQN agent utilizing intrinsic motivation for environment-agnostic policy learning.

This pipeline enables learning and reasoning without relying on handcrafted semantic labels or domain-specific reward functions. While semantic labels are used internally by the segmentation module to generate class maps, they are not, actually, provided as supervision for symbolic reasoning or reward assignment; the agent and the symbolic engine operate purely on observed state changes and event structures.

5.2 General Architecture of the Framework

5.2.1 Design Choices

The architecture of the framework is guided by four fundamental design principles, each contributing to its flexibility, robustness, and transparency.

The first principle, modularity, ensures that each component of the system is implemented as an independent, self-contained module with clearly defined interfaces. This design allows vision, symbolic reasoning, and reinforcement learning components to be developed, tested, or replaced in isolation, facilitating experimentation and incremental improvements without disrupting the functionality of the overall system.

Domain agnosticism represents a second cornerstone: the symbolic reasoning and reinforcement learning modules are intentionally designed to operate over arbitrary simulations, without assuming any prior knowledge of game mechanics, object semantics, or reward structures. This capability allows the framework to generalize across diverse environments and tasks, enhancing its adaptability and reducing the need for manual engineering.

A third guiding principle is incremental knowledge construction; whereby symbolic knowledge is not static but continuously refined. As the system observes additional trajectories, it dynamically discovers new events, rules, and abstractions, progressively building a richer internal model of the environment. This incremental learning mechanism mirrors human-like cognitive processes, allowing the agent to improve its understanding over time.

Finally, interpretability sets the framework apart from conventional end-to-end black-box approaches. By exposing intermediate symbolic representations, including object properties, causal relationships between events, and inferred rules, the system allows researchers and practitioners to inspect, analyse, and validate the agent's reasoning process. Importantly, these symbolic components do not directly influence the DQN agent's decisions; rather, they provide a means to quantify how much the symbolic model has captured about the environment. This transparency not only facilitates debugging and knowledge extraction but also provides a foundation for explainable AI, ensuring that the agent's decisions can be understood and trusted.

5.2.2 Selected Techniques and Algorithms

The framework combines several established techniques in a novel configuration:

- U-Net [21] convolutional architecture for pixel-wise semantic segmentation.

- Heuristic population-based symbolic induction for rule discovery.
- Automatic state and event extraction based on attribute introspection.
- Deep Q-Learning [28] for policy learning.
- Intrinsic motivation mechanisms, including curiosity and causal impact rewards.

The interaction between these techniques allows the agent to learn meaningful behaviors even in the absence of explicit task rewards.

5.2.3 Modular Structure of the System

Visual Input Acquisition and Preprocessing Module

The frame acquisition module is responsible for efficiently handling visual input data and preparing it for subsequent processing stages. It supports both online acquisition of frames from live simulations and the loading of previously recorded gameplay logs. Each frame is resized to a fixed resolution to ensure a consistent input size for the neural network and is converted into a normalized tensor to facilitate stable training.

The corresponding segmentation masks are loaded as integer-valued images, where each pixel represents a semantic class identifier, allowing the network to learn to associate spatial regions with meaningful object categories. A custom dataset class encapsulates the logic for pairing each image with its corresponding mask, while PyTorch data loaders handle batching, shuffling, and splitting into training and validation sets. A fixed random seed is used to ensure reproducibility of the dataset splits, which is crucial for consistent evaluation of model performance. By abstracting low-level data handling, this module provides a clean interface for the feature extraction stage, allowing the focus to remain on the learning task rather than on preprocessing logistics.

Perception Module

The perception module performs semantic segmentation using a U-Net architecture [21], which is highly effective for dense pixel-wise prediction tasks. The architecture follows the classic encoder-decoder paradigm with skip connections (see Figure 4.2), enabling the model to combine high-level semantic information with fine-grained spatial details .

Training is conducted with a multi-class cross-entropy loss, optimized using the Adam optimizer. In order to improve stability and generalization, the training process incorporates data augmentation, input normalization, small batch sizes, and early stopping based on validation loss. Evaluation metrics include pixel accuracy and mean Intersection over Union (mIoU) [20], which quantify agreement between predicted and ground-truth segmentation maps.

Overall, this module abstracts the pipeline from raw video frames to class-labeled segmentation maps, producing dense semantic representations that form the foundation for downstream tasks. Its design supports reproducible training and efficient inference, combining robust data handling with a well-established deep learning architecture.

Symbolic Reasoning Module

The symbolic reasoning module converts low-level perceptual outputs into high-level symbolic representations. Each segmented object is transformed into a symbolic entity characterized by:

- A unique identifier.
- A set of observable properties (e.g., position, shape)
- A history of detected events.
- Associations with inferred behavioral rules.

Object persistence across frames is handled through patch tracking, allowing the system to reason about temporal continuity and object identity.
Rather than injecting predefined domain knowledge, the system employs a heuristic initialization process that generates a population of candidate symbolic explanations. Each individual in the population represents a distinct hypothesis about how objects behave and interact over time. Individuals are evaluated based on their ability to explain observed events while minimizing unexplained changes. Scoring functions reward coherence, temporal consistency, and explanatory completeness. The best-performing individuals are retained, serialized, and summarized into higher-level prototypes. These prototypes act as emergent core knowledge, capturing regularities such as object dynamics or interaction patterns without explicit supervision.

Reinforcement Learning Agent

The reinforcement learning agent is implemented using a fully generic symbolic environment designed to work across different simulations without requiring manually engineered features.

The environment, automatically, thought a specific module, extracts all observable numerical and boolean attributes from the simulation object, forming a state representation. Each attribute represents an observable property, such as positions, velocities. Events are detected by comparing consecutive states and identifying significant changes in attribute values. A causal event tracker groups events occurring within a temporal window into causal chains. Longer chains yield higher intrinsic rewards, encouraging the agent to discover action sequences that produce rich and structured dynamics. Effectively, the agent is rewarded not only for individual events but also for sequences of events that demonstrate causal influence, promoting more complex behaviors.

The agent is trained using Deep Q-Learning with experience replay, target networks, and ε-greedy exploration. All these techniques are introduced to improve stability and efficiency during training, especially:

- **Experience Replay**: Transitions (state, action, reward, next state, done) are stored in a buffer and sampled randomly in batches for training. This breaks correlations between consecutive experiences, reduces variance, and allows the agent to reuse past experiences multiple times, improving sample efficiency.
- **Target Networks**: A separate target Q-network is maintained and updated periodically with the weights of the main Q-network. This provides a stable target for Q-value updates, mitigating oscillations and divergence that can occur when the network is updated using its own rapidly changing estimates.
- **ε-Greedy Exploration**: Actions are selected according to an ε-greedy policy, where with probability ε a random action is chosen to encourage exploration, and with probability $1 - \varepsilon$ the action with the highest predicted Q-value is selected. Over time, ε decays to favor exploitation of the learned policy while still maintaining some exploration to prevent local optima.

A central aspect of the implementation is the use of intrinsic reward signals. The total reward at each time step is composed of several components:

- **Event-centered reward:** Proportional to the number of detected events and the length of their causal chains.
- **Causal impact bonus:** Measures how much the outcome of an action differs from a no-op baseline, encouraging interventions that meaningfully affect the environment.
- **Curiosity-driven bonus:**Derived from the prediction error of a learned forward dynamics model, which estimates the next state given the current state and action.
- **Event density bonus:**Optionally rewards large overall changes in the state vector, promoting exploration of diverse configurations.

Formally, the total reward R_{total} can be expressed as:

$$R_{\text{total}} = R_{\text{event}} + w_{\text{causal}} R_{\text{causal}} + w_{\text{curiosity}} R_{\text{curiosity}} + w_{\text{density}} R_{\text{density}} \tag{5.1}$$

This reward structure allows the agent to learn even in environments with sparse or absent external rewards.

5.2.4 Optimizations and Efficiency Considerations

The implementation of the framework incorporates several strategies aimed at improving both computational efficiency and the stability of the training process.

One of the key measures is early stopping during the segmentation model training. By monitoring the validation loss and halting training when improvements

plateau, the framework prevents overfitting while reducing unnecessary computational load.

In the reinforcement learning component, the dimensionality of the observation vector is intentionally limited to reduce computational overhead and accelerate training. Although the state extractor is capable of detecting and encoding numerous attributes from the simulation, only a small subset of numerical features is included in the state vector passed to the agent. This reduces the computational overhead per step and accelerates training.

Another optimization is the adaptive update of the target network in the DQN agent. Early in training, the target network is updated more frequently to promote faster convergence, while updates become less frequent as the agent stabilizes. This balance between rapid learning and stable evaluation mitigates the risk of oscillations in Q-values. Additionally, the agent uses a replay buffer from which moderate-sized batches are sampled for training. This approach mitigates correlations between consecutive transitions, allows efficient reuse of past experiences, and balances gradient accuracy with computational cost. Collectively, these optimizations allow the framework to remain tractable despite its multi-layered architecture. By combining standard deep learning practices with domain-agnostic symbolic reasoning techniques, the system achieves a balance between expressiveness and computational efficiency, ensuring that training and evaluation can be conducted on realistic hardware resources.

6 Experimental Evaluation

This chapter presents the experimental evaluation of the proposed framework, with particular emphasis on the reinforcement learning experiments based on causal intrinsic rewards. Although preliminary tests on symbolic model induction are briefly discussed to contextualize the approach, the core contribution of this work lies in the design and validation of a domain-agnostic causal reward mechanism integrated into a Deep Q-Network (DQN) agent and tested on different arcade-style environments. The evaluation addresses five main questions:

1. Does per-frame semantic segmentation provide a reliable perceptual grounding for symbolic knowledge extraction?
2. Can structured event logs bridge the gap between visual scene understanding and symbolic world representation?
3. Can the system extract meaningful structural knowledge from interaction logs?
4. Can a fully generic, intrinsic, causally grounded reward drive effective reinforcement learning?
5. How do different intrinsic components (causal impact, curiosity, event density, shaping) affect performance?

6.1 Datasets and Testing Scenarios

The experimental evaluation was conducted on procedurally generated arcade-style environments designed to provide controlled yet structurally rich interaction dynamics. All environments were deterministic unless otherwise specified and were implemented to expose structured state variables rather than raw pixel observations.

Two main domains were considered:

- A simplified Arkanoid-like environment with ball, paddle, and destructible bricks.
- A Pong-like environment with a ball and two paddles.

For the symbolic representation experiments, three progressively complex Arkanoid variants were used:

- **Minimal Environment**: ball and walls only.
- **Full Environment**: ball, paddle, and bricks.
- **Modified Variants**: larger ball size, altered velocities, and stochastic bounce dynamics.

For reinforcement learning experiments, the following configurations were evaluated:

- **Arkanoid**
  - Baseline Arkanoid with 24 bricks.
  - Modified Arkanoid versions for generalization testing:
    * Increased ball size.
    * Increased number of bricks.
    * Altered visual properties (colors).
- **Pong**: environment used for cross-domain validation.

Each RL configuration was trained for up to 5000 episodes, with periodic checkpoint evaluation over 100 test episodes per model. Importantly, in generalization experiments, the trained model was evaluated without retraining under structural modifications.

6.1.1 Evaluation Metrics: Accuracy, Interpretability, Scalability

Evaluation was conducted along three orthogonal axes: task performance, structural coherence, and interpretability. The performance was measured using win rate measured as the percentage of episodes successfully completed, average number of bricks destroyed, average episode length (steps, in this context, one step is equal to one frame), survival time (only for pong), and stability across evaluation runs. These metrics quantify whether intrinsic reward is sufficient to produce goal-directed behavior without task-specific supervision. The symbolic layer was evaluated in terms of coherence of induced object classes, consistency of transition rules across episodes, stability under parameter changes, and correct abstention in stochastic conditions. These criteria assess whether the symbolic module produces well-formed and reliable representations of the environment. At the RL layer, interpretability was assessed through behavioral robustness under structural modifications, invariance to visual features, and generalization across object count variations. Rather than focusing solely on reward maximization, this axis investigates whether learned policies reflect structural regularities of the environment.

6.2 Experimental Results

6.2.1 Semantic Grounding and Symbolic Log Reconstruction

A key preliminary step in our framework is the extraction of symbolic representations from visual input. The first question addressed is whether per-frame semantic segmentation can provide a reliable perceptual grounding for symbolic knowledge extraction, and whether structured logs can serve as an interface between visual understanding and symbolic reasoning.

From Semantic Masks to Symbolic Logs

In principle, one might attempt to translate segmentation maps directly into symbolic logs compatible with the inference module. However, early experiments showed that this naive mapping is insufficient due to two main factors:

1. Noise in segmentation outputs: even state-of-the-art segmentation models produce spurious predictions, small artifacts, or intermittent missing detections, which can disrupt the identification of consistent object instances over time.

2. First-generation symbolic module limitations: the initial symbolic reasoning pipeline expects structured, temporally coherent inputs. Raw masks alone do not encode relational information or temporal consistency, leading to fragmented event chains and breakdown in rule induction.

To overcome these challenges, it was developed a refinement pipeline that converts semantic masks into structured interaction logs while preserving object identity and temporal continuity.

Symbolic Log Reconstruction

Since the environment is a controlled synthetic domain, the geometric properties of all game objects are known a priori. This allows segmentation output to be systematically corrected to match the expected real-world shapes and dimensions. For example, a brick missing a corner pixel is completed, and slightly malformed contours are regularized. The semantic network inference is thus not treated as ground truth, but as a noisy signal that is refined against a structural prior derived from the environment itself.

The reconstruction process proceeds in two main stages. In the first stage, mask parsing and object extraction, each frame is analysed to identify individual instances of balls, paddles, walls, and bricks using connected-component analysis. Each instance is assigned a unique identifier, particularly important for bricks, which can be numerous, and position, size, and hitbox information are extracted and normalized into a structured object description. In the second stage, frame-wise log assembly, each frame is transformed into a structured dictionary encoding four fields: elements, containing the objects present with their properties; commands, serving as

placeholders for agent actions at this stage; events, initialized as empty lists for later derivation; and *global_state*, capturing environment-level metadata such as score and object velocity. This pipeline ensures that the symbolic module receives high-quality, temporally coherent logs, mitigating the effects of segmentation noise and module sensitivity.

In addition, this procedure provides two main advantages:

- **Bridging perception and symbolic reasoning**: structured logs encode both visual properties and inferred dynamics, enabling the symbolic module to operate without direct pixel-level information.
- **Noise resilience**: correcting malformed contours, completing missing pixels, and discarding spurious artifacts, compensate for residual segmentation errors, allowing rule induction to focus on causal regularities rather than transient noise.

6.2.2 Preliminary Validation: Symbolic Representation Learning

This section presents a set of experiments conducted using the preliminary version of the Symbolic reasoning module. The goal is to evaluate the quality of the internal representations learned by the model by comparing them with the ground-truth rules governing the simulated environment.

An episode corresponds to the complete temporal sequence of frames generated during a single gameplay session, capturing the dynamic evolution of the environment over time. In order to ensure exposure to diverse interaction patterns, the initial position of the ball is randomized at the beginning of each episode.

At this stage of development, three different environments have been considered for evaluation. First, a minimal version with only walls and a ball, after a complete Arkanoid environment with paddle and bricks, and in the end, modified variants with altered physical parameters (e.g., larger ball, increased speed, stochastic bounce). Across these scenarios, the system demonstrated the ability to:

- Group patches into consistent object classes.
- Infer transition rules linking events (e.g., contacts) to state updates.
- Generalize structural rules across instances of the same object class.
- Adapt parameters when physical quantities change (e.g., initial velocity).

In deterministic settings, the inferred rules matched the true simulator dynamics. When stochasticity was introduced (random bounce speed), the system correctly refrained from extracting stable deterministic rules, highlighting a sensitivity to statistical regularity.

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

Interpretation:

- Bricks disappear when hit, reflecting object removal in the environment.
- Velocity inversion occurs upon *element* contact.

Additionally, transfer experiments showed that initializing object classes from previously processed episodes drastically reduced convergence time, confirming that structural representations are reusable across similar environments.

The experiments reported here should be considered exploratory. Nevertheless, they provide a meaningful proof of concept for the proposed framework. In particular, the results highlight the system's ability to formulate structural hypotheses after a relatively limited number of interactions, and to revise or adapt these hypotheses when exposed to variations in the environment.

6.2.3 Reinforcement Learning module

The central contribution of this work is the design of a fully generic reinforcement learning environment wrapper capable of:

- Automatically extracting observable state variables.
- Detecting significant state transitions as events.
- Building short causal event chains.
- Generating intrinsic rewards from structural interaction patterns.

Arkanoid Experiments

A systematic evaluation was conducted across multiple reward configurations, each trained for up to 5,000 episodes on the Arkanoid environment. For this framework, one episode corresponds to one complete game, composed of individual steps where each step advances the simulation by exactly one frame.

In this first set of experiment, the game consists of 24 destructible bricks that the agent must eliminate within a 60-second time limit to avoid infinite loop situations. For each configuration, three model checkpoints were preserved during training: the model achieving the highest cumulative reward (*best_reward*), the model with the best time-efficiency score, meaning the goal of surviving longer in the environment (*best_time*), and the final model at the end of training (*last_generic*). Final performance was evaluated using four complementary indicators: win rate, mean number of bricks destroyed per episode, average episode length (in steps), and consistency across evaluation runs.

The three reward configurations tested were: pure causal and curiosity without shaping or density (*no_shaping_no_density*), causal reward combined with a handcrafted paddle-ball alignment bonus (*yes_shaping_no_density*) and causal reward augmented with an event-density component rewarding the magnitude of state

transitions (*no_shaping_yes_density*). These were benchmarked against a reference model (*WINNING_MODEL*) trained with full supervision, which achieved a 100%-win rate and an average episode length of approximately 1,791 steps.

Configuration 1: Pure Causal + Curiosity (no_shaping_no_density)

This configuration produced the most consistent and robust results across all checkpoints. The *last_generic_model*, representing the agent's behavior at the end of the full training run, achieved a 99% win rate with an average of 23.98 bricks destroyed per episode and a mean episode length of approximately 1,687 steps across winning runs. The *best_reward* checkpoint also performed strongly at 81% wins, and the *best_time* checkpoint reached 77%.

In both cases, failures were exclusively caused by episodes where the agent became stuck in degenerate loops rather than showing any systematic policy breakdown, as evidenced by the extremely high step counts (reaching up to ∼484,000 steps) in non-winning episodes. This loop behavior indicates the agent retained ball contact but failed to progress toward brick destruction, rather than losing the ball entirely. Importantly, because the reward depends only on event chains, the agent has no prior notion of which events are strategically more valuable. For instance, a sequence involving two wall collisions generates the same intrinsic signal as a sequence consisting of a wall collision followed by a brick hit. The system does not encode predefined relevance; it simply maximizes structured interaction.

The central outcome of this configuration is that an intrinsic reward signal grounded solely in causal interaction structure is sufficient to achieve near-optimal control. Even in the absence of any task-specific or externally engineered reward, the agent autonomously acquires behaviors such as correctly deflecting the ball, maintaining prolonged exchanges, and progressively eliminating all bricks.

This finding supports the core hypothesis of the thesis: the causal structure embedded in state-transition sequences provides sufficient informational richness to drive effective policy learning in this domain, even without explicit goal-oriented reward shaping.

Configuration 2: Shaping Enabled - No Density

The introduction of a handcrafted shaping term rewarding paddle-ball alignment produced markedly degraded and unstable performance. The *best_reward* checkpoint failed completely, achieving a 0%-win rate across all 100 test episodes, with each episode terminating in fewer than 30 steps. Inspection of the step counts confirms the agent was losing the ball almost immediately after launch, suggesting the policy learned to maximize the alignment reward at the expense of ball retention. The *best_time* and *last_generic* checkpoints recovered partially, reaching win rates of 54% and 53% respectively, but both remained well below the unshaped configuration. This degradation is interpretable as a form of reward hacking [68]. The alignment bonus creates a shortcut in the reward landscape: the agent can accumulate shaping reward by positioning the paddle near the ball's projected contact point without necessarily developing the long-horizon behavior required to destroy all bricks. When the shaping signal dominates early training, it crowds out the

structural curiosity signal that would otherwise drive exploration of causal interaction chains. The result is a policy that is locally well-calibrated for paddle positioning but globally brittle. This finding has an important methodological implication: handcrafted shaping signals can be harmful when a strong intrinsic structural reward is already present, because they introduce competing learning pressures that distort the trajectory of policy improvement.

Configuration 3: No Shaping - Density Enabled

Adding an event-density component, a reward term proportional to the magnitude of state change across consecutive frames, produced the highest peak performance of any configuration. The *best_time* checkpoint achieved a 100%-win rate with an average of 24.00 bricks destroyed per episode and a mean episode length of approximately 1,804 steps, essentially matching the supervised reference model. This represents a complete and reliable solution to the environment under purely intrinsic motivation.

However, the across-checkpoint variance was substantially higher than in the no-density configuration. The *best_reward* model achieved only 52% wins and exhibited a characteristic bimodal failure pattern: some episodes terminated in under 100 steps, suggesting immediate ball loss, while others ran for hundreds of thousands of steps, suggesting loop behavior. The *last_generic* model reached 57% wins, with similar instability. This variance suggests that the density reward accelerates early exploration, helping the agent discover productive interaction patterns more quickly, but introduces optimization instability that makes the final learned policy less reliable. The density term rewards novelty in state transitions independently of their causal structure. While this promotes rapid coverage of the state space, it may also encourage the agent to seek dynamically rich states that are not strategically useful for brick destruction, thereby disrupting the convergence properties that the pure causal signal provides.

Configuration 4: Shaping Enabled - Density Enabled

The simultaneous activation of the handcrafted paddle-ball alignment bonus and the event-density term produced a markedly different dynamic compared to the previous configurations. Unlike the shaping-only setup, which collapsed due to reward hacking, and the density-only configuration, which showed high variance across checkpoints, this combined formulation achieved both high peak performance and strong final reliability.

The *best_time* checkpoint reached a 100% win rate, destroying all 24 bricks in every episode with an average episode length of approximately $1,761$ steps. This performance fully matches the supervised reference model. Crucially, unlike the density-only configuration, the success here is not transient: the *last_generic* checkpoint maintains a 96% win rate, indicating that performance remains stable at the end of training rather than appearing only as a temporary optimum.

The *best_reward* checkpoint achieved a lower but still substantial 78% win rate. As in other density-based settings, non-winning episodes display extremely high step counts ($\sim$430,000 steps), indicating degenerate loop behavior rather than immediate

ball loss. This confirms that the dominant failure mode remains structural stagnation rather than control breakdown. From a learning-dynamics perspective, the density component appears to mitigate the pathological effect observed when shaping is used in isolation. In the shaping-only configuration, the alignment bonus dominated early training, encouraging locally optimal paddle positioning without long-horizon brick-clearing behavior. When density is added, however, the agent continues to receive reward for dynamically rich state transitions, preventing premature convergence to alignment-focused strategies. The resulting behavior is both globally competent and locally precise. The agent consistently maintains long interaction chains, progressively eliminates bricks, and rarely collapses into early termination patterns. The presence of occasional loop episodes suggests that optimization instability is not entirely removed, but its frequency is substantially reduced relative to density-only training. The central implication of this configuration is that intrinsic reward components are not inherently antagonistic; rather, their interaction depends on balance. While shaping alone distorts the learning landscape, and density alone introduces variance, their combination can produce near-perfect performance with high reliability.

Table 6.1: Performance comparison across all model configurations.

Model	**Win Rate**	**Avg Bricks**	**Avg Steps (wins)**
WINNING_MODEL (reference)	100%	24.00	1,791
no_shaping_no_density – best_reward	81%	23.75	1,638
no_shaping_no_density – best_time	77%	23.03	1,998
no_shaping_no_density – last	99%	23.98	1,6878
yes_shaping_no_density – best_reward	0%	0.00	24
yes_shaping_no_density – best_time	54%	22.18	1,766
yes_shaping_no_density – last	53%	20.21	1,611
no_shaping_yes_density – best_reward	52%	16.48	1,794
no_shaping_yes_density – best_time	**100%**	24.00	1,804
no_shaping_yes_density – last	57%	21.06	1,662
yes_shaping_yes_density – best_reward	80%	23.44	1,835
yes_shaping_yes_density – best_time	**100%**	24.00	1,765
yes_shaping_yes_density – last	97%	23.88	1,780

Final consideration

Two configurations produced models that approach the supervised reference: shaped causal reward with density (*yes_shaping_yes_density*) and causal reward with density (*no_shaping_yes_density*). Both incorporate the same intrinsic density modulation, yet they differ in the presence of an additional shaping term.

Empirically, the configuration without shaping achieves perfect performance and stable convergence, whereas the shaped counterpart, although capable of reaching high win rates, exhibits greater variance and reduced robustness across checkpoints (see Figure 6.1). This contrast suggests that the density component alone, while capable of driving strong performance, provides a purely magnitude-based signal: it rewards large overall state transitions without distinguishing whether they con-

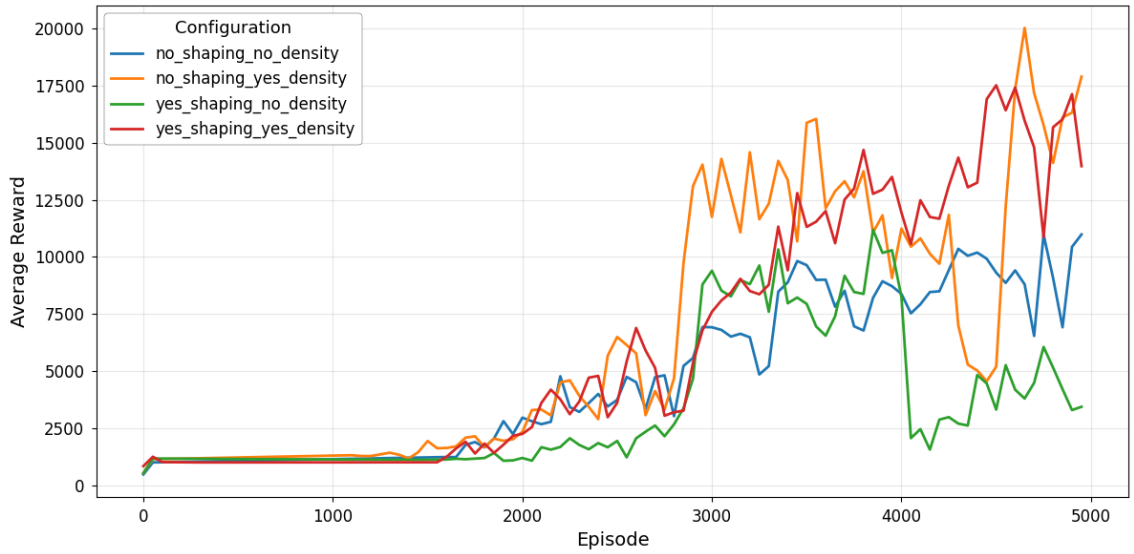

Figure 6.1: The figure shows the average reward over 5,000 training episodes for four different configurations of the DQN agent on the Arkanoid environment. All configurations start with similar low rewards and begin to diverge around episode 2,000. The configurations with density reward (*yes_density*) tend to achieve higher average rewards, with *no_shaping_yes_density* reaching the peak around episode 4500. The *yes_shaping_no_density* configuration shows the weakest performance in the later episodes.

tribute to long-term task structure. When shaping is introduced alongside density, it does not conflict with the intrinsic reward. Instead, it adds a mild directional bias that can regularize exploration by encouraging locally coherent behaviors, without overriding the causal signal. In other words, the density term amplifies state variation in a domain-agnostic manner, whereas shaping introduces weak task-aligned structure. Their combination does not distort the optimization landscape; rather, it slightly constrains it in a way that can improve stability. Empirically, this is reflected in the high and consistent performance of the *yes_shaping_yes_density* configuration, which remains competitive with the purely density-driven variant.

However, the shaping component presupposes deeper prior knowledge of the environment. Designing an effective shaping reward requires assumptions about task structure, desirable intermediate behaviors, or implicit notions of progress. In this work, shaping was introduced only as an experimental probe, to test whether explicit guidance could complement with the intrinsic causal formulation. However, the primary objective of the proposed framework is to remain environment-agnostic. The agent should rely solely on the structural properties of interaction dynamics, without handcrafted bonuses tailored to a specific game.

Extending the comparison, two configurations ultimately match or approach the supervised reference: pure causal reward (*no_shaping_no_density*) and causal reward with density (*no_shaping_yes_density*). The crucial distinction between them lies not in peak performance but in training stability. The pure causal configuration converges reliably and maintains strong performance across checkpoints, while the density-augmented configuration reaches its optimum only transiently during training before destabilizing [69].

In summary, these experiments demonstrate that the causal interaction structure captured by the proposed intrinsic reward is both necessary and sufficient for learning to play Arkanoid without any external reward signal. The agent must not only establish contact with the ball but maintain extended chains of coordinated interactions across dozens of bricks, operating over long temporal horizons. The fact that this competence emerges without shaping and, without task-specific priors, reinforces the central claim: structured intrinsic motivation grounded in causal interaction dynamics is both robust and general, and does not require environment-dependent reward engineering.

6.2.4 Generalization Under Structural Modifications

To further evaluate the structural robustness of the learned policy, the best-performing Arkanoid model was tested under controlled modifications of the environment, without performing any retraining. The goal was to determine whether the learned behavior depends on superficial perceptual features or whether it captures deeper causal regularities of the domain.

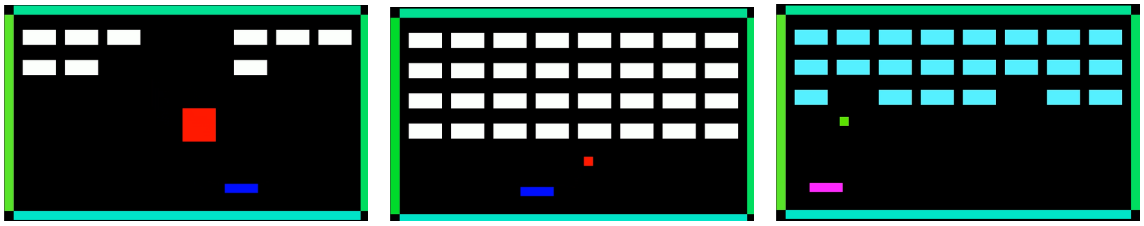

Figure 6.2: From left to right: the three stylistic modifications introduced in the environment — increased paddle size, increased number of bricks, and altered object colours.

Three types of modifications were introduced: an increase in ball size, an increase in the number of bricks, and variations in the visual appearance of objects such as colors (see Figure 6.2). In the first experiment, the ball size was increased by a factor of five, significantly altering collision geometry and raising the probability of multiple brick contacts within a single frame. Despite this substantial physical change, the agent maintained stable behavior and continued to solve the task reliably. This result is particularly meaningful given that the observation space encodes normalized physical variables, positions and velocities, and that the intrinsic reward is derived exclusively from state transitions and causal impact. The policy is therefore not tied to specific geometric assumptions about object scale; instead, it reacts to relational dynamics such as ball-paddle alignment, collision-induced velocity changes, and brick disappearance events, demonstrating clear invariance to object size as long as the underlying causal interaction structure remains consistent. In the second experiment, the number of bricks was increased beyond the original 24, affecting episode length, event frequency, and interaction density. Performance remained high even without retraining, with the agent successfully clearing the environment in most test runs. This outcome suggests that the learned policy does not memorize fixed spatial layouts but generalizes across variations in object count. Since the intrinsic reward depends on event chains and causal impact rather than explicit score signals, the agent remains motivated to generate meaningful interactions regardless of how many bricks are present the objective internalized by the

network is not destroy exactly 24 bricks, but rather something closer to maximizing structured causal engagement with the environment. Finally, additional tests confirmed that modifying the colours of the ball, paddle, and bricks has no effect on performance. This invariance follows directly from two core design choices: the observation extractor operates on numerical state attributes rather than raw pixels, and the reward mechanism depends exclusively on state transitions and causal differences. As a result, the policy is entirely insensitive to changes in color, texture, or visual theme. Taken together, these experiments highlight a fundamental property of the proposed framework: the learned behavior is grounded in causal structure, not perceptual appearance.

6.2.5 Pong Experiments

To test domain generality, the same framework was applied to Pong without structural modification of the intrinsic reward mechanism. The techniques developed for Arkanoid transfer effectively to Pong (see Figure 6.3), confirming that the approach is not tied to a specific game architecture. However, model performance in Pong is noticeably lower, which can be attributed to the fact that Pong generates fewer simultaneous events compared to Arkanoid, providing a sparser set of interaction opportunities for the agent to exploit. Training logs show gradual increases in survival time and event chain complexity. After 1000 episodes, the average survival time reached approximately 1.3 seconds, indicating that the agent had begun exploiting interaction regularities but had not yet converged to expert-level behavior. Although Pong learning is slower than Arkanoid, the same intrinsic reward structure drives improvement without any task-specific reward definition. This supports the claim that the proposed reward mechanism captures generic properties of interactive dynamics rather than game-specific objectives.

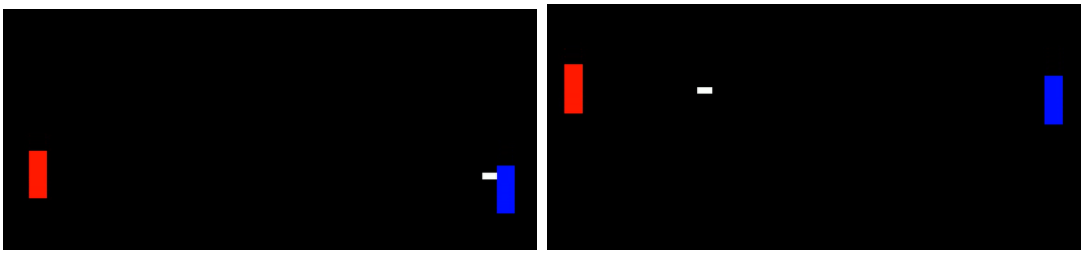

Figure 6.3: The figures represent the Pong environment.

6.2.6 Critical discussion of the results

The experimental results highlight several important findings concerning both the representational layer and the reinforcement learning dynamics enabled by the proposed framework.

First, the symbolic modeling component confirms that the system captures consistent object-centric causal dynamics in deterministic environments and adapts parameters under structural modifications. The induced rules remain stable when physical quantities such as velocity magnitude or object size change, provided that

the underlying causal mechanisms are preserved. This establishes that the representation layer encodes structural regularities rather than superficial statistical correlations.

Second, the reinforcement learning experiments demonstrate that purely intrinsic, causally grounded reward signals can drive complex goal-directed behavior. The agent successfully learns to solve Arkanoid without any handcrafted task reward. In fact, the experiments show that adding shaping signals can be detrimental when intrinsic structural rewards are already informative, suggesting that externally imposed heuristics may interfere with the emergence of structurally coherent policies. Event-based causal interaction appears to function as a powerful exploration driver, encouraging sustained engagement with meaningful environment dynamics. The strongest empirical result is the near-perfect performance (up to 100%-win rate) in Arkanoid using only domain-agnostic intrinsic rewards. This suggests that maximizing structured causal interaction with the environment can implicitly approximate goal-directed behavior in tasks where success depends on maintaining coherent interaction chains over time. Crucially, the generalization experiments further reinforce this interpretation. The best-performing Arkanoid model was evaluated under structural modifications, including increased ball size, increased number of bricks, and variations in visual appearance, without any retraining. Performance remained stable across these changes. This invariance is particularly significant. The agent does not rely on fixed geometric assumptions (such as a specific ball size), nor does it encode expectations tied to a fixed number of bricks. Instead, it continues to generate effective behavior when the interaction density increases or when collision geometry slightly changes. Likewise, modifications to colours or visual attributes do not affect performance, since the observation space is defined over normalized physical state variables rather than pixel-level features.

7 Conclusion and Future Works

This thesis has examined fundamental limitations in contemporary artificial intelligence systems, particularly their brittleness, limited reasoning capabilities, and lack of transparency. Modern large language models often operate through large-scale statistical pattern extraction rather than grounded semantic understanding, while deep reinforcement learning agents typically fail to construct transferable, structured knowledge about the environments in which they operate. In both cases, the dominant paradigm remains heavily sub-symbolic and correlation-driven, with limited capacity for explicit causal modelling or conceptual abstraction.

Drawing on insights from philosophy, cognitive science, and artificial intelligence research, this work aligns with a growing body of scholarship arguing for hybrid architectures that integrate symbolic structure with neural computation. Researchers such as Melanie Mitchell and Gary Marcus have emphasized the importance of combining statistical learning with structured reasoning in order to achieve robustness, interpretability, and systematic generalization.

Inspired by Core Knowledge Theory, this thesis leverages the foundations of a framework that constructs structured representations of dynamic environments by identifying entities through fundamental properties and modelling their interactions symbolically. Rather than relying solely on end-to-end optimization, the system abstracts object classes, infers interaction rules, and builds a reusable structural representation of the environment.

Preliminary empirical results indicate that this structured approach improves interpretability, robustness under environmental modifications, and explainability of learned behavior in dynamic domains. At the same time, several challenges remain: the current implementation lacks deep context-based reasoning, operates with a still limited and immature knowledge base, faces potential combinatorial growth in rule induction, does not yet fully integrate formal causal reasoning mechanisms, and does not perfectly incorporate the perceptual module. These limitations define a rich space for future research.

7.1 Future Directions

The framework, in its present form, demonstrates that symbolic, interpretable models of dynamic environments can be constructed by grounding object tracking and rule inference in structural priors. Although the initial validation in controlled game environments shows promising generalization and robustness, further development is necessary to extend the framework's scalability and applicability.

7.1.1 Perception Module

A first essential extension concerns the integration of an explicit perception module tightly coupled with the symbolic representation layer. Although a perceptual element has been introduced in the current implementation, segmentation outputs are refined post hoc into structured logs rather than being directly extracted as object patches. A more advanced architecture would introduce a bidirectional loop:

- Bottom-up perception extracts candidate object patches from raw frames.
- Top-down predictions from the symbolic model guide attention toward expected spatial regions.

Once an object has been modelled symbolically, its predicted future position could inform attention mechanisms in the segmentation stage, focusing computation on areas where the object is expected to appear. This integration of perceptual input with conceptual priors would significantly improve robustness to noise and partial observability, while enabling scaling to real-world visual domains.

7.1.2 RL Integration

A further extension of this work involves embedding the existing framework within a fully model-based reinforcement learning architecture, inspired by a Dyna-style approach. In the current implementation, the agent relies primarily on off-policy Q-learning and incorporates a forward model only indirectly through curiosity-based reward shaping. While this provides some predictive information, the agent does not yet use the symbolic internal representation to plan or simulate future trajectories explicitly.

In order to achieve a fully model-based setup, the symbolic internal representation could be leveraged as a predictive engine for internal rollouts. In this configuration, the agent would maintain an internal model of the environment derived from the structured rules and object representations. Incoming observations would be encoded into symbolic states capturing the relevant entities and their properties, and potential actions could be evaluated by simulating their effects within this internal model. This refinement would allow the agent to anticipate the consequences of actions before executing them in the real environment, using the symbolic framework not only for decision-making but also as a testable hypothesis of environmental dynamics. Discrepancies between predicted and actual outcomes would then inform continuous refinement of the internal model. This approach could preserve the existing modules, event detection, causal chains, curiosity, and density-based shaping, while augmenting the system with planning capabilities and enhanced interpretability, progressively transforming it into a fully model-based agent.

7.1.3 Generalization Across Environments

Another important avenue for future research is the evaluation and enhancement of the framework's ability to generalize across different environments. Although

current experiments demonstrate promising transfer within similar game instances, extending the symbolic representation and rule abstraction mechanisms to support entirely new domains remains largely unexplored. Future work could investigate how the structured knowledge captured by the framework, such as object classes, interaction rules, and causal chains, can be adapted or reused in novel environments, providing robust generalization while minimizing retraining. This could involve mechanisms for abstracting high-level patterns, aligning entities across domains, or leveraging meta-learning strategies to accelerate adaptation to previously unseen scenarios.

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015. [Online]. Available: https://doi.org/10.1038/nature14539

[2] B. Liang, Y. Wang, and C. Tong, "Ai reasoning in deep learning era: From symbolic ai to neural–symbolic ai," *Mathematics*, vol. 13, no. 11, p. 1707, 2025.

[3] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018. [Online]. Available: https://arxiv.org/abs/1801.00631

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] A. S. d'Avila Garcez, L. C. Lamb, and D. M. Gabbay, *Neural-Symbolic Cognitive Reasoning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. [Online]. Available: https://link.springer.com/book/10.1007/978-3-540-73246-4

[6] A. S. d'Avila Garcez, T. R. Besold, L. De Raedt, P. Földiak, P. Hitzler, T. Icard, and D. L. Silver, "Neural-symbolic learning and reasoning: Contributions and challenges," in *AAAI Spring Symposia*, Mar. 2015, pp. 18–21. [Online]. Available: https://www.aaai.org/Library/Symposia/Spring/ss15-01.php

[7] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, p. e253, 2017.

[8] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[9] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*. Springer, 2006, pp. 404–417.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 886–893.

[13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, 1988, pp. 147–151.

[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision (ECCV)*. Berlin, Heidelberg: Springer Berlin Heidelberg, May 2006, pp. 430–443.

[15] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5929–5955, 2020. [Online]. Available: https://doi.org/10.1007/s10462-020-09825-6

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014, submitted on 4 Sep 2014; last revised 10 Apr 2015 (v6). [Online]. Available: https://arxiv.org/abs/1409.1556

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[19] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds., vol. 9351. Springer, Cham, 2015, pp. 234–241. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28

[22] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Englewood Cliffs: Prentice-Hall, 1995.

[23] A. Newell and H. A. Simon, "Computer science as empirical inquiry: Symbols and search," in *ACM Turing Award Lectures*. Association for Computing Machinery, 2007, p. 1975. [Online]. Available: https://dl.acm.org/doi/book/10.5555/1283922

[24] R. J. Brachman and H. J. Levesque, *Knowledge Representation and Reasoning*. San Francisco, CA, USA: Elsevier, 2004.

[25] M. Minsky, "A framework for representing knowledge," in *MIT-AI Laboratory Memo 306*, Jun. 1974. [Online]. Available: http://hdl.handle.net/1721.1/6160

[26] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993. [Online]. Available: https://doi.org/10.1006/knac.1993.1008

[27] F. Baader, Ed., *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge, UK: Cambridge University Press, 2003.

[28] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[29] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2015.

[30] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3–4, pp. 279–292, 1992. [Online]. Available: https://doi.org/10.1007/BF00992698

[31] L. Huang *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, Jan. 2025. [Online]. Available: http://dx.doi.org/10.1145/3703155

[32] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013. [Online]. Available: https://doi.org/10.1613/jair.3912

[33] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling, "Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents," *Journal of Artificial Intelligence Research*, vol. 61, pp. 523–562, 2018. [Online]. Available: https://doi.org/10.1613/jair.5699

[34] A. S. d'Avila Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neuro-symbolic artificial intelligence: The state of the art," *arXiv preprint arXiv:1905.06088*, 2019. [Online]. Available: https://arxiv.org/abs/1905.06088

[35] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "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. [Online]. Available: https://doi.org/10.1145/3442188.3445922

[36] S. Banerjee, A. Agarwal, and S. Singla, "Llms will always hallucinate, and we need to live with this," *arXiv preprint arXiv:2409.05746*, 2024. [Online]. Available: https://arxiv.org/abs/2409.05746

[37] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "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. [Online]. Available: https://doi.org/10.1145/3442188.3445922

[38] H. J. Levesque, E. Davis, and L. Morgenstern, "The winograd schema challenge," in *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*. AAAI Press, 2012, pp. 552–561. [Online]. Available: https://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4492

[39] X. Qu *et al.*, "Minimalistic attacks: How little it takes to fool a deep reinforcement learning policy," *arXiv preprint arXiv:1911.03849*, 2020. [Online]. Available: https://arxiv.org/abs/1911.03849

[40] M. Mitchell, "Llms and world models - part 1," https://aiguide.substack.com/p/llms-and-world-models-part-1, 2023, accessed: 2026-02-28.

[41] G. Marcus, "The next decade in ai: Four steps towards robust artificial intelligence," *arXiv preprint arXiv:2002.06177*, 2020. [Online]. Available: https://arxiv.org/abs/2002.06177

[42] C. Molnar, *Interpretable Machine Learning*. Lulu Press, 2020. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[43] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, pp. 665–673, 2020. [Online]. Available: https://doi.org/10.1038/s42256-020-00257-z

[44] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017. [Online]. Available: https://arxiv.org/abs/1702.08608

[45] C. Molnar, M. T. Ribeiro, and W. Samek, "Interpreting black-box models: A review on explainable artificial intelligence," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–37, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s12559-023-10179-8

[46] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *ACM Computing Surveys*, vol. 53, no. 3, pp. 1–37, 2020. [Online]. Available: https://doi.org/10.1145/3359786

[47] A. Vellido, J. D. Martín-Guerrero, and P. J. G. Lisboa, "Making machine learning models interpretable," in *Proceedings of the 2012 International Conference on Artificial Neural Networks (ICANN)*. Springer, 2012, pp. 23–29. [Online]. Available: https://doi.org/10.1007/978-3-642-31537-4_4

[48] P. Radanliev, "Intelligenza artificiale: Riflessioni sul passato e sguardo al prossimo cambio di paradigma," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 37, no. 7, pp. 1045–1062, 2025. [Online]. Available: https://doi.org/10.1080/0952813X.2024.2323042

[49] S. Chakraborty *et al.*, "Interpretability of deep learning models: A survey of results," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, San Francisco, CA, USA, 2017, pp. 1–6. [Online]. Available: https://doi.org/10.1109/UIC-ATC.2017.8397411

[50] Z. C. Lipton, "The mythos of model interpretability," *Communications of the ACM*, vol. 61, no. 10, pp. 36–43, 2018. [Online]. Available: https://doi.org/10.1145/3233231

[51] M. T. Ribeiro, S. Singh, and C. 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. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[52] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. [Online]. Available: https://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions

[53] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2014. [Online]. Available: https://arxiv.org/abs/1312.6034

[54] M. Craven and J. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 1996, pp. 24–30.

[55] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 9505–9515. [Online]. Available: https://arxiv.org/abs/1810.03292

[56] Y. Bengio, Y. LeCun, and G. Hinton, "Deep learning for ai," *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3448250

[57] E. S. Spelke and K. D. Kinzler, "Core knowledge: Cognitive principles and psychological foundations," *Developmental Science*, vol. 10, no. 1, pp. 89–96, 2007. [Online]. Available: https://doi.org/10.1111/j.1467-7687.2007.00569.x

[58] L. Feigenson, S. Dehaene, and E. Spelke, "Core systems of number," *Trends in Cognitive Sciences*, vol. 8, no. 7, pp. 307–314, 2004. [Online]. Available: https://doi.org/10.1016/j.tics.2004.05.002

[59] M. C. Frank, D. L. Everett, E. Fedorenko, and E. Gibson, "Number as a cognitive technology: Evidence from pirahã language and cognition," *Cognition*, vol. 108, no. 3, pp. 819–824, 2008. [Online]. Available: https://doi.org/10.1016/j.cognition.2008.04.007

[60] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 45, pp. 18 327–18 332, 2013. [Online]. Available: https://doi.org/10.1073/pnas.1306572110

[61] J. R. Anderson, M. Matessa, and C. Lebiere, "Act-r: A theory of higher level cognition and its relation to visual attention," *Human–Computer Interaction*, vol. 12, no. 4, pp. 439–462, 1997. [Online]. Available: https://doi.org/10.1207/s15327051hci1204_5

[62] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial Intelligence*, vol. 33, no. 1, pp. 1–64, 1987. [Online]. Available: https://doi.org/10.1016/0004-3702(87)90050-6

[63] S. Shah, H. Goel, S. S. Narasimhan, M. Choi, S. P. Sharan, O. Akcin, and S. Chinchali, "A challenge to build neuro-symbolic video agents," *arXiv preprint arXiv:2505.13851*, 2025. [Online]. Available: https://arxiv.org/abs/2505.13851

[64] P.-Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? a typology of computational approaches," *Frontiers in Neurorobotics*, vol. 1, p. 6, 2007. [Online]. Available: https://doi.org/10.3389/neuro.12.006.2007

[65] J. Lehman and K. O. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–223, Jun. 2011. [Online]. Available: https://doi.org/10.1162/EVCO_a_00025

[66] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 2778–2787. [Online]. Available: https://proceedings.mlr.press/v70/pathak17a.html

[67] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 248–255. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206848

[68] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016. [Online]. Available: https://arxiv.org/abs/1606.06565

[69] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros, "Large-scale study of curiosity-driven learning," *arXiv preprint arXiv:1808.04355*, 2018. [Online]. Available: https://arxiv.org/abs/1808.04355