

University of Warsaw  
Faculty of Philosophy

Miłosz Tkacz

Record book number: 446250

# Analyzing the Relationship Between Predictive Error of World Models and Performance in Exploration Tasks in Reinforcement Learning

Bachelor's thesis  
in COGNITIVE SCIENCE

Supervisor:  
**dr Jan Kwapisz**  
Faculty of Physics

Warsaw, Sep 2025



## **Abstract**

Recent advances in artificial intelligence can be attributed to data exploitation methods: powerful computational optimizations, new neural architectures enabling generative modeling, and techniques for compressing and learning from massive datasets.

Despite this, the problem of creating truly intelligent systems grounded in stable and robust internal representations allowing extrapolation of knowledge remains yet to be solved. Exploration is the missing piece: the ability to identify what remains unknown, to seek novel experiences, and to maintain stable internal representations that support continuous learning.

This thesis examines how reinforcement learning with world models can contribute to this goal. World models provide agents with predictive capabilities that guide and motivate behavior, but their role in exploration is not yet fully understood. Alongside experiments, this work presents a theoretical synthesis of two major research traditions on world models: intrinsic motivation and learning from imagined trajectories - situating them within broader insights from reinforcement learning and cognitive science.

The central contribution is an experimental analysis of how the predictive error of a world model influences exploration performance in reinforcement learning agents. By systematically varying the accuracy of the world model through controlled update schedules, the study reveals that improved prediction enhances exploration, but with diminishing returns once surprisal is reduced too quickly. The findings indicate that too much accuracy too quickly can reduce exploratory behavior, highlighting the need for a balance between prediction and novelty.

## **Keywords**

Reinforcement Learning; World Models; Artificial Curiosity; Intrinsic Motivation; Exploration–Exploitation; Predictive Coding; Free Energy Principle; Cognitive Science; Philosophy of Mind; Mental Representations.

## **Tytuł pracy w języku polskim**

Analiza zależności między błędem predykcyjnym modeli świata a wydajnością w zadaniach eksploracyjnych w uczeniu ze wzmocnieniem.



# Contents

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| <b>1. Introduction</b>                                                      | 5  |
| 1.1. Chapter Overview                                                       | 6  |
| <b>2. Key Concepts</b>                                                      | 7  |
| 2.1. Reinforcement learning                                                 | 7  |
| 2.2. Exploration-exploitation dilemma                                       | 8  |
| 2.3. Extrinsic and intrinsic rewards                                        | 8  |
| 2.4. Mathematical formalism in reinforcement learning                       | 9  |
| 2.4.1. Optimal control and Bellman equation                                 | 9  |
| 2.4.2. Markov Decision Process                                              | 9  |
| 2.4.3. Direct policy learning                                               | 11 |
| 2.4.4. Model-based RL                                                       | 12 |
| 2.5. World Models                                                           | 13 |
| 2.5.1. What are world models?                                               | 13 |
| 2.5.2. Background and decision-time planning with world models              | 14 |
| 2.5.3. Evidence for world models in humans                                  | 15 |
| 2.5.4. Artificial curiosity                                                 | 16 |
| 2.5.5. Contents of internal representations within world models             | 16 |
| 2.5.6. Efficiency of internal representations                               | 17 |
| <b>3. Selected reinforcement learning algorithms utilizing world models</b> | 19 |
| 3.1. Intrinsic Curiosity Module                                             | 19 |
| 3.1.1. Architecture                                                         | 20 |
| 3.1.2. Strengths and limitations                                            | 21 |
| 3.2. Random Network Distillation                                            | 21 |
| 3.3. Architecture                                                           | 22 |
| 3.4. Strengths and limitations                                              | 22 |
| 3.5. Dreamer                                                                | 22 |
| 3.5.1. Architecture                                                         | 23 |
| 3.5.2. Dreamer-V2 and V3                                                    | 25 |
| 3.5.3. Limitations                                                          | 25 |
| 3.5.4. InDRiVE: Combining Intrinsic Rewards with Imagination                | 26 |
| <b>4. Experiments</b>                                                       | 29 |
| 4.1. Method                                                                 | 29 |
| 4.1.1. Training environment                                                 | 29 |
| 4.1.2. Model architecture                                                   | 30 |
| 4.1.3. Experimental setup                                                   | 31 |

|        |                                                        |    |
|--------|--------------------------------------------------------|----|
| 4.1.4. | Training parameters and hardware . . . . .             | 31 |
| 4.2.   | Results . . . . .                                      | 32 |
| 4.2.1. | Preliminary experiment . . . . .                       | 32 |
| 4.2.2. | Plain PPO, RND and ICM comparison experiment . . . . . | 33 |
| 4.2.3. | Weight update skip experiment . . . . .                | 34 |
| 4.3.   | Discussion . . . . .                                   | 36 |
| 4.4.   | Future Research . . . . .                              | 37 |

# Chapter 1

## Introduction

Recent advancements in artificial intelligence have been driven by breakthroughs in deep learning methods such as residual connections (He et al., 2016), batch normalization (Ioffe and Szegedy, 2015), transformer architectures (Vaswani et al., 2017), and diffusion models (Ho et al., 2020). These paradigm-shifting approaches primarily address the challenge of how to effectively learn from large and complex datasets. They provide solutions to issues such as preserving learning signals in very deep networks, modeling long sequences of data, and efficiently training large-scale architectures.

However, the field of AI research is moving from 'learning from data' to 'learning what data to learn from'. Despite more progress being done every day in the former area, the current limitations of modern AI is heavily based on the latter (Jiang et al., 2022). This shift places the problem of autonomous data acquisition as central to AI research. Elements of this problem have long been studied within the reinforcement learning, where exploration-exploitation dilemma was always one of the main interests in the field (Sutton and Barto, 2020).

One class of reinforcement learning approaches to exploration relies on world models. These models equip AI systems with planning capabilities that guide the exploration process. What makes them distinct from classic search-based methods such as Monte Carlo Tree Search is that they can be trained without access to the forward dynamics of the training environment, making them more feasible in real-life scenarios. This research direction draws inspiration from cognitive science, where there is substantial evidence for analogous structures in the human brain that support prediction, planning, and goal-directed behavior (Ha and Schmidhuber, 2018; Friston, 2010). Importantly, cognitive studies also suggest that human internal models are not exact copies of reality, but rather simplified and adaptive abstractions that balance accuracy with usefulness. This perspective motivates the question of whether artificial world models also require such a balance, rather than striving for perfect prediction.

Against this background, one underexplored research question concerns the relationship between the predictive error of a world model and the exploration capabilities of the AI system that relies on it. Although previous research provides evidence that more accurate world models can lead to better task performance (Whitney and Fergus, 2019), it remains unclear whether continuous improvements in prediction accuracy always benefit exploration. There is also little analysis of how a gradual increase in world model accuracy might influence exploration efficiency in practice. This question is particularly important in contexts where policy and world models are trained simultaneously, and the former directly depends on the representations generated by the latter. Such training setup is common in a subfield of reinforcement learning called artificial curiosity (Schmidhuber, 2010). Those methods,

inspired by human experience, utilize internal rewards accumulated from exploring unknown states which leads to exploration. As stressed in Schmidhuber, 2010, the intrinsic reward is based on the predictive error of the world model that arises from the mismatch between estimates of the future states and the real state of the environment. If the predictions are lacking in accuracy, then everything is surprising and there is no meaningful signal of novelty in explored states. Conversely, if the predictions are perfect, then there is no signal at all and agent has no incentive to explore. A good balance between predictive accuracy and surprisal must be maintained in order for the exploration to be efficient.

Building upon the results from past research, this work examines the hypothesis that increasing accuracy leads to improved exploration, but with diminishing returns once a sufficient level of predictive capability is reached. To investigate this, experiments were conducted in which world models performance was systematically weakened by gradually increasing probability of skipping weight updates. This setup allowed for the comparison of prediction error and exploration efficiency under varying levels of world model accuracy. This is motivated by both theoretical considerations from artificial curiosity and empirical evidence that agents benefit from, but are not strictly dependent on perfect prediction.

The central aim of this thesis is therefore to explore the relationship between world model accuracy and exploration efficiency.

## 1.1. Chapter Overview

The remainder of this thesis is organized as follows:

- **Chapter 2 (Key Concepts).** Core background: RL fundamentals, the exploration–exploitation dilemma, intrinsic vs. extrinsic rewards, formalism (Bellman/MDP/policy gradients), model-based RL, world models, cognitive-science links, and artificial curiosity.
- **Chapter 3 (Selected reinforcement learning algorithms utilizing world models).** Survey of algorithms using world models for exploration and/or imagination-based learning, focusing on ICM, RND, Dreamer (V1–V3), and hybrids that combine intrinsic motivation with imagination.
- **Chapter 4 (Experiments).** Experimental setup and findings on how world-model predictive error relates to exploration performance. This chapter includes:
  - Method — environments (MiniGrid/BabyAI), agent architectures (PPO, PPO+ICM, PPO+RND), the weight-update skip protocol, training settings, and metrics;
  - Results — baseline comparisons and the effect of update probability on prediction error and sample efficiency; regression tests of diminishing returns;
  - Discussion — interpretation, limitations (stochasticity, sparse rewards, long horizons), and directions for future work.

## Chapter 2

# Key Concepts

This chapter introduces the theoretical background necessary to understand the role of world models for exploration in reinforcement learning. The discussion combines perspectives from machine learning and insights from cognitive science and philosophy of mind. Reinforcement learning is presented as a computational framework for adaptive behavior, followed by an overview of the exploration–exploitation dilemma and its mathematical foundations. Subsequent sections connect these ideas to the notion of world models, both in artificial agents and in humans, highlighting their relevance for curiosity, quality of internal representations, and the efficiency of learning. Together, these concepts provide an interdisciplinary foundation for the later experimental and analytical parts of this thesis.

### 2.1. Reinforcement learning

Reinforcement learning (RL) is a subset of machine learning that is often defined as learning which actions should be taken to maximize a numerical reward function (Sutton and Barto, 2020) (Francois-Lavet et al., 2018). Every action choice corresponds to mapping from state of the environment to action space, which can be discrete, such as in the game of chess, or continuous as in the cart pole problem. In most interesting cases, we care about sequences of such states and actions, in which action may affect multiple future states, all resulting in optimization of the cumulative reward function.

A useful analogy comes from instrumental reinforcement in behavioral psychology, where behaviors are shaped by their consequences: actions followed by rewards become more likely, while those followed by punishments become less likely. RL formalizes this idea in a computational setting.

Reinforcement learning is simultaneously a problem, set of methods solving it, and a research field that studies them.

It is distinct from supervised learning, where the task is to learn the underlying data distribution given a set of labeled data points used as a source of ground truth. Supervised learning system learns to classify previously unseen data points utilizing information embedded in the training set. In RL a set of ground truth examples from which a system can learn is not known, as it is often impossible to describe a fixed set of situations containing a pattern of action that would allow for creation of generalized policy performing well in an uncertain environment. (Sutton and Barto, 2020)

Lack of supervised examples makes RL akin to unsupervised learning with which it is often mistakenly conflated. In contrast to unsupervised learning, the objective of RL is not to learn the patterns hidden in the dataset, but it is to maximize the reward function that

reinforces certain behavior. This problem setting is often described as a Markov Decision Process, where given past states, agent must decide on actions affecting following states, all within an uncertain environment in which state transitions are a stochastic process. This framework will be described in one of the following sections.

## 2.2. Exploration-exploitation dilemma

One of the challenges that sets RL apart from supervised and unsupervised learning is exploration-exploitation dilemma. Data used in training reinforcement learning systems contain neither labeled examples nor a fixed set of representative data points. Conversely, data used in RL systems comes from a small and ever-changing subset of all possible states; hence the RL agents must learn to effectively explore previously unseen states of the environment, later utilizing the knowledge gathered from those experiences. Rewards are often sparse and unavailable, as a result there is often not enough training signal for machine learning systems to learn patterns from their experiences.

Supervised and unsupervised learning methods, depend on representativeness of the training set of data points; such assumption can be considered as solving only the exploitation part of the RL problem setting. Examples provided to those learning systems are a result of a data collection conducted by external learning supervisors. RL resigns from this assumption, considers data collection as the part of the broader learning process, and hence extends the challenge of machine learning by another dimension.

It is often seen as a trade-off, as learning to explore to gather new knowledge for the future comes with the expense of losing opportunity of exploiting short-term rewards. Agents not capable of exploration will fail to adapt to changing environment and will lose opportunities of collecting higher rewards that may lie outside of the state space explored in the past. On the other hand, exploration often comes with a risk of failure, which might lead agents to fail in accumulating reward signals disabling them from learning at all.

## 2.3. Extrinsic and intrinsic rewards

In most reinforcement learning settings, agents are trained with extrinsic rewards, signals explicitly provided by the environment designer to reflect task objectives. Classic examples include reaching the end of a maze, winning a game, or maximizing a score. Extrinsic rewards are often sparse, delayed, or difficult to specify in a way that captures the full range of desired behaviors, which makes learning slow and sample-inefficient.

To address these challenges, researchers have introduced intrinsic rewards, signals generated within the agent itself to encourage exploration or the acquisition of useful knowledge. Intrinsic rewards are typically derived from prediction error, novelty, or information gain, and are designed to push agents toward states that are surprising or informative rather than simply rewarding in an extrinsic sense. They provide a self-supervised training signal that can be present even in the absence of external rewards.

The notion of intrinsic rewards is closely inspired by psychology and neuroscience, where intrinsic motivation refers to behaviors driven by curiosity, exploration, or the inherent satisfaction of learning. In developmental psychology, children engage in exploratory play without explicit external rewards, gradually building models of the world that later support goal-directed behavior (Schmidhuber, 2010). This psychological perspective provides a strong foundation for artificial analogues of curiosity in reinforcement learning.

Recent work has shown that purely intrinsic objectives are sufficient to drive effective learning in complex environments. For example, Burda, Edwards, Pathak, et al., 2018 demonstrated that agents trained solely with intrinsic signals can successfully learn to perform a wide range of tasks, even when extrinsic rewards are completely absent. This suggests that curiosity and novelty-seeking alone can provide a strong basis for autonomous skill acquisition.

In addition, architectures that first pre-train on intrinsic rewards and later fine-tune with extrinsic signals have been shown to improve sample efficiency and transfer. Khanzada and Kwon, 2025 provide evidence that such intrinsically motivated pre-training leads to faster policy convergence compared to learning the task directly with extrinsic rewards. This highlights the complementary role of intrinsic and extrinsic rewards: the former guides exploration and representation learning, while the latter aligns behavior with explicit task goals.

## 2.4. Mathematical formalism in reinforcement learning

### 2.4.1. Optimal control and Bellman equation

The main concern of the optimal control problem is finding optimal control of the dynamic system so that an objective function is optimized. Dynamic programming is considered the only feasible way to solve the optimal control problem. It solves a decision process by breaking it down into a sequence of action steps over time. Relationship between the consecutive action steps is often represented recursively in a backward-induction form called Bellman equation (“Bellman equation”, 2025) ‘

#### Bellman equation

$$V(x) = \max_{a \in \Gamma(x)} \{R(x, a) + \gamma V(T(x, a))\}. \quad (2.1)$$

Where:

- $V(x)$  is the expected return starting from state  $x$  under an optimal policy.
- $a \in \Gamma(x)$  is an action  $a$  taken from the set of possible actions  $\Gamma$  depending on the state  $x$ .
- $R(x, a)$  is a reward of taking action  $a$  at the state  $x$ .
- $\gamma \in [0, 1)$  is a discount factor that encourages instantaneous reward.
- $T(x, a)$  is a transition function that returns a state  $x_{t+1}$  given action  $a_t$  taken in state  $x_t$ .

Such statement of the problem sets  $V(x)$  as function that needs to be learned in order to construct a policy  $\pi(x)$  that chooses action  $a$  associated with the highest value of  $V$ .

### 2.4.2. Markov Decision Process

Markov Decision Process (MDP) is often used as the framework describing the problem of reinforcement learning. It is a discrete and stochastic version of the optimal control problem introduced by Bellman (Bellman, 1957) (Sutton and Barto, 2020). The method takes its name from Markov chains which were the main object of study of Russian mathematician Andrey Markov.

MDP is formally defined as 5-tuple  $(S, A, T, R, \gamma)$ , as introduced in Sutton and Barto, 2020, where:

- $S$  is a state space.
- $A$  is an action space.
- $T : S \times A \times S \rightarrow [0, 1]$  is the transition function, consisting of a set of conditional state-transition probabilities.
- $R : S \times A \times S \rightarrow \mathcal{R}$  is the reward function where  $\mathcal{R}$  is a set of possible rewards in range  $R_{max} \in \mathbb{R}^+$ .
- $\gamma \in [0, 1)$  is the discount factor.

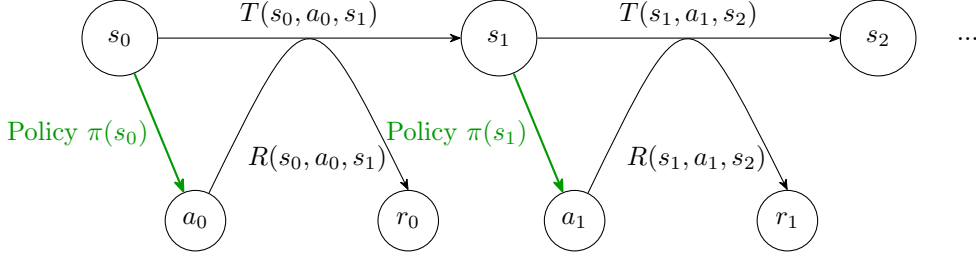


Figure 2.1: Illustration of a Markov Decision Process (MDP), adapted from the structure presented in Francois-Lavet et al., 2018. At each step, the agent takes an action that changes its state and yields a reward.

As in the Bellman equation, we are concerned with finding the value of the  $V$  function.  $V$  function depends on policy  $\pi(s)$  which determines which actions are selected in state  $s$ . This relationship can be modeled in recurrent form as follows.

$$V^\pi(s) = \sum_{s'} T(s, \pi(s), s') (R(s, \pi(s), s') + \gamma V^\pi(s')) \quad (2.2)$$

This can be derived from 2.1 by fixing policy  $\pi$  and rewriting the action  $a$  as being sampled from it ( $\pi(s)$ ). Next, for stochastic state transitions we rewrite the equation as expectation over all possible next states  $s'$  that occur with a probability dependent on the transition function  $T$  and also alter the reward function to be dependent on the next state.

The objective of the RL agent is to learn an optimal policy  $\pi^*(s)$ . The value function following the optimal policy is called  $V^*$  and can formally be defined as (“Markov decision process”, 2025):

$$V^*(s) = \max_{\pi \in \Pi} V^\pi(s) \quad (2.3)$$

There are multiple successful variants of the framework above used in various contexts. The most notable is Q-learning, which modifies V-learning by assuming action  $a$  to be taken first and then following policy  $\pi$ . This modification allows the algorithm to directly approximate the optimal action-value function  $Q^*(s, a)$ , without requiring a model of the environment or explicit knowledge of the current policy. More formally this can be written as:

$$Q^\pi(s, a) = \sum_{s'} T(s, a, s') (R(s, a, s') + \gamma Q^\pi(s', a = \pi(s'))) \quad (2.4)$$

In this formulation, policy is learned implicitly by learning association of state-action pairs with approximate return cumulated throughout the episode. As a result, Q-learning is an off-policy method, enabling learning from actions generated by an exploratory or even unrelated behavior policy, while still converging to the optimal policy (Sutton and Barto, 2020).

Q-learning started as an algorithm for tabular methods that track visited states and update mapping from state, action pairs to Q-function. In practice, it is unfeasible to visit all states and keep mapping of state-action pairs in memory. This issue is solved by learning an approximation of the Q function with neural networks (Francois-Lavet et al., 2018).

### 2.4.3. Direct policy learning

On the contrary to the Q-learning, which by design leads to greedy methods of choosing highest-scoring actions, direct policy learning methods allow learning action distribution from which actions can be sampled leading to explicit exploratory behavior (Francois-Lavet et al., 2018). This is why policy-learning methods are dominant in contexts where exploration is vital for reaching sparse rewards.

Direct policy learning is made possible by the Policy Gradient Theorem (Sutton et al., 1999), which provided a closed-form expression for the gradient of expected reward with regards to policy network weights as shown below:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{s \sim d^{\pi_{\theta}}, a \sim \pi_{\theta}} [\nabla_{\theta} \log \pi_{\theta}(a | s) Q^{\pi_{\theta}}(s, a)] \quad (2.5)$$

Where  $\theta$  are neural network parameters,  $J$  is a loss function,  $Q^{\pi_{\theta}}$  is a Q-function with currently learned policy  $\pi_{\theta}$  and  $d^{\pi_{\theta}}$  is a discounted state distribution where probability of visiting state is given by:

$$d^{\pi_{\theta}}(s_t) = (1 - \gamma) \sum_{t \geq 0} \gamma \Pr(s_t = s | \pi) \quad (2.6)$$

This allows directly learning the action distribution. Method above requires access to Q-function, which in practice is also approximated. Example architecture combining policy learning and Q-function approximation is Actor-Critic network (Konda and Tsitsiklis, 1999). Here two separate networks are trained so that actor parameters take state as an input and return an action distribution, and critic network approximates value of a given state.

For numerical stability reward function is replaced with advantage function  $A^{\pi}$  which is defined as:

$$A^{\pi}(s, a) = Q^{\pi}(s, a) - V^{\pi}(s) \quad (2.7)$$

This describes how a choice of a particular action compares to the baseline value of a given state. Positive values suggest that the choice is advantageous, while negative values point to disadvantage. Operation of subtracting the baseline value of a state leads to decrease in magnitude of the signal sent to the model. As a result, weight updates are more stable and higher learning rate can be used, leading to faster convergence (Francois-Lavet et al., 2018).

To encourage choosing low-probability actions, entropy bonus is often applied.

$$H(\pi) = -\alpha \sum \pi(s, a) \log(\pi(s, a)) \quad (2.8)$$

Where  $\alpha$  is a hyperparameter controlling the magnitude of the bonus.

Applying operations described above leads to the formula:

$$\nabla_{\theta} V^{\pi_{\theta}}(s_0) = \mathbb{E}_{s, a} [\nabla_{\theta} \log \pi_{\theta}(s, a) A^{\pi_{\theta}}(s, a)] + \alpha \mathbb{E}_s [\nabla_{\theta} H^{\pi_{\theta}}(s)]. \quad (2.9)$$

Direct policy learning methods tend to be sample inefficient. As on-policy methods they can only perform a single weight update on collected experience, which is very limiting, because environment simulation is often expensive. Performing multiple steps on the same experience might lead to biased gradient estimates, which destabilize the training and violate assumption that experience data was collected under current policy. Different methods for

improving sample-efficiency by allowing multiple gradient update steps were proposed. This can be achieved by limiting the divergence between old and current policy during weight update. Methods like Trust Region Policy Optimization (TRPO) accomplish that by limiting KL-divergence between old and new policy to be less than hyperparameter  $\delta$  (Schulman et al., 2015). Proposed method requires estimation of second-order derivative which renders it computationally expensive.

Two years later the Proximal Policy Optimization (PPO) method was proposed (Schulman et al., 2017), since then it became one of the most influential direct policy learning methods. PPO stabilizes training by limiting the magnitude of the objective for weight updates through a clipped surrogate objective:

$$\mathbb{E}_t \left[ \min \left( \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t^\pi, \text{clip} \left( \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1 - \epsilon, 1 + \epsilon \right) A_t^\pi \right) \right] \quad (2.10)$$

Probability ratio measures the likelihood that a given action is taken compared to the old policy. Clipping is applied to it to keep the weight update magnitude low enough to avoid divergence of policy. Despite its simplicity, it was empirically proven to perform well across a wide range of tasks.

#### 2.4.4. Model-based RL

Another approach to reinforcement learning that will be the focus of my work is model-based RL (MBRL). This class of solutions solves the RL problem with planning by combining world models, discussed in the next section, and by storing a global solution which corresponds to policy  $\pi^*$  (Latyshev and Panov, 2023). Those methods allow RL systems to plan ahead using the predictive capabilities of world models. By learning forward dynamics of the environment in which the RL agent is placed, it can estimate states of the environment following the observed one. By having direct access to a state-space estimates within learned latent space and combining it with policy, MBRL systems are capable of planning several steps into the future. Latent spaces are explored by sampling trajectories and this leads back to the difficulty of balancing exploration and exploitation. To explore MBRL system must select trajectories that have low certainty (high variance). On the other hand, exploitation requires choosing trajectories that have known, high reward associated to them (Francois-Lavet et al., 2018). More formally this can be described as RL system with:

- forward dynamics model  $\mathcal{M}$
- $\mathcal{S}_t = s_{t-n}, \dots, s_t$  is a state history with up to  $n$  previous states.
- $\mathcal{Z}$  is a latent space from which  $z_t, z_{t+1}, \dots, z_{t+h}$  can be generated using model  $\mathcal{M}$
- Actions  $\mathcal{A}^h = a_1, \dots, a_h$  where  $a_1$  leads to transition from  $z_t$  to  $z_{t+1}$
- $\tilde{\mathcal{S}}$  are states estimates  $s_{t+1}, \dots, s_{t+h}$  calculated based on their latent space representations.

Formalism above can be extended to ensemble  $\mathcal{M}^k$  of  $k$  models. This can be achieved by exploring latent spaces  $\mathcal{Z}^{1 \dots k}$  by generating multiple trajectories independently from separate models and using them directly or combining them into a single state estimate (Latyshev and Panov, 2023).

Several advantages of model-based reinforcement learning (MBRL) over model-free approaches can be highlighted: (1) MBRL methods can generalize to new tasks within the same environment; (2) they enable learning from state transition data, making them independent

of the specific policy being trained; and (3) they are generally more sample-efficient. Conversely, MBRL based methods struggle when their predictive capacity is limited. Increasing their planning horizons allows their performance to match their state-of-the-art model-free counterparts (Whitney and Fergus, 2019).

## 2.5. World Models

### 2.5.1. What are world models?

The concept of world models has its philosophical roots in the notion of mental models, one of the first works comprehensively introducing this concept was "The Nature of Explanation" by Kenneth Craik in 1943. Craik defined a model as a "physical or chemical system which has a similar relation-structure to that of the process it imitates" (Craik, 1943).

Although Craik frequently used the term "mechanism," his emphasis on relation-structure points more naturally toward a functional account of explanation. In this view, a model explains external phenomena not by replicating their material composition, but by preserving the relevant input-output relations and causal dependencies. In other words, the model mirrors the structure of external processes in such a way that it can generate analogous effects to those observed in the real system.

Craik illustrated this with the example of Kelvin's tide predictor: although the device bears no resemblance to an actual tide, it combines oscillations of different frequencies in a way that accurately reproduces the variations in tidal amplitude at a given location over time. This line of thought resonates with more recent discussions distinguishing functional from mechanistic accounts of explanation, where functional analyses describe capacities in terms of input-output relations, while mechanistic accounts emphasize the causal organization of underlying structures (Milkowski, 2013).

The goal of this idea was to explain the process of thought, which Craik decomposed into three stages:

1. Translation of external process into words, numbers or other symbols.
2. Arrival of other symbols through reasoning, deduction and inference.
3. Retranslation of these symbols into external processes.

In humans, the translation would occur through the sense-organs activating the neural pathways in the brain. Then through neural activations, mental models would conduct symbolic reasoning, and lastly the retranslation would occur through activation of motor-organs. The main role of such models in animals would be to try out various alternatives of actions (Craik, 1943).

This notion was first explored in early connectionist work such as Sutton and Barto, 1981, where adaptive networks were shown to build internal representations of their environment. This idea was extended in Sutton and Pinette, 1985, which demonstrated that a neural network could learn a world model of a maze through exploration and later exploit it once rewards were introduced, anticipating what is now called latent learning. A decade later, Sutton, 1991 introduced the Dyna architecture, the first reinforcement learning framework to integrate learned world models with planning capabilities, allowing policy updates from both real and imagined experience. Shortly after, Schmidhuber, 1991 proposed how world models could also serve as a basis for intrinsic motivation, introducing curiosity and boredom mechanisms in reinforcement learning systems, as described in more detail in later sections.

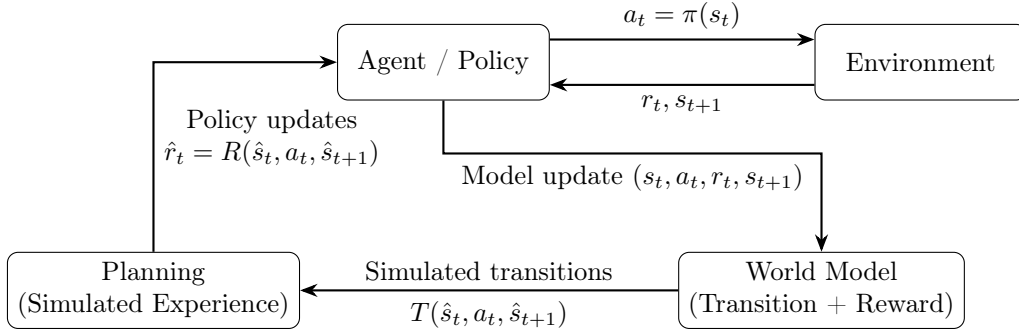


Figure 2.2: The Dyna architecture (Sutton, 1991), where the agent interacts with the real environment while simultaneously improving a learned world model. The model enables planning through simulated experience with learned transition  $T$  and reward  $R$  models, allowing policy updates beyond direct interaction.

In RL the term mental model is mostly replaced by the notion of world model. The distinction seems to depend mainly on the context in which the term is used. For example, Schmidhuber in his 2018 work "World Models" starts by explaining the idea of mental models, then switches to using the term "world model" without reintroducing it (Ha and Schmidhuber, 2018). This points to the fact that world model and mental model is the same idea differentiated by the field of study. The former is used in cognitive science, while the latter is used in reinforcement learning.

### 2.5.2. Background and decision-time planning with world models

Sutton and Barto, 2020 in chapter 8.8 introduced crucial distinction between two types of planning in RL:

- background planning - type of planning used in Dyna (Sutton, 1991) allowing imagining trajectories at training time and learning policy from them.
- decision-time planning - computes multi-step plan of action at inference time.

The difference lies in when planning happens. In background planning it only happens at the training time, while in decision-time planning happens just before choosing an action in inference time.

Although both are commonly referred to as "planning," this terminology can be misleading. In the case of background planning, no explicit lookahead is performed during inference. Instead, the system leverages imagined trajectories during training, which implicitly equip the policy with long-horizon competence. Thus, the agent's action choice at inference time is reactive rather than the result of an explicit search.

By contrast, decision-time planning is more naturally aligned with the everyday meaning of planning: the agent explicitly generates, evaluates, and executes multi-step action sequences at inference time.

Given this discrepancy, it may be useful to reserve the term planning exclusively for decision-time approaches, while referring to background planning instead as learning from imagined trajectories. This distinction would emphasize the fundamentally different computational processes at work, avoid terminological confusion, and help clarify discussions of methods such as AlphaZero (decision-time planning) (Silver et al., 2017) versus Dreamer (background trajectory learning) (Hafner et al., 2020).

### 2.5.3. Evidence for world models in humans

The notion that humans maintain internal “world models” is strongly supported by findings in cognitive psychology and neuroscience. Multiple empirical studies give evidence that perception and action are guided by anticipatory models of the environment rather than by passive reception of sensory input (Clark, 2013). Studies on animal brains show that retinal ganglion cells use spatial-temporal predictions of visual stimuli. They suppress nearby signals to highlight only those that diverge from the assumption that visual input remains similar across space and time (Hosoya et al., 2005). This gives evidence of bottom-up predictive signal influencing perception, different studies give evidence for predictive capacity found in top-down pathways in the brain. For example (Hohwy et al., 2008) explain the famous “binocular rivalry” effect by alternation between different hypotheses explaining contradictory sensual information.

The free energy principle of brain function proposes that incoming signals are continuously compared with internally generated predictions, and that the mismatch (prediction error) drives both learning and conscious perception. The goal of a living organism is then to decrease the free energy which can correspond to surprisal. This can be achieved either by updating its posterior distribution meaning refining its best guess about the external environment, or by changing the external environment through its actions (Friston, 2010).



Figure 2.3: Illustration of inattention blindness: participants focusing on counting basketball passes often fail to notice a person in a gorilla suit crossing the scene (Simons and Chabris, 1999). This shows that mental models in humans ignore significant part of sensual data irrelevant to the task at hand.

Importantly, human world models are not exact copies of the sensual data but are shaped for usefulness. Herbert Simon’s concept of bounded rationality (Simon, 1957) emphasizes that cognition exploits simplified representations suited to decision-making under resource constraints. Empirical phenomena such as change blindness (Rensink et al., 1997) and inattention blindness (Simons and Chabris, 1999) show that large alterations in visual scenes often go unnoticed if they are irrelevant to current goals. Collectively, these results indicate that human internal models are adaptive abstractions: they prioritize task-relevant features and predictive utility over complete accuracy.

This perspective has direct relevance for reinforcement learning. Just as humans rely on simplified yet useful models for prediction and action, artificial agents benefit from world models that capture the aspects of the environment relevant for control, while ignoring irrelevant complexity (Pathak et al., 2017). The analogy strengthens the interdisciplinary connection between cognitive science and model-based approaches in machine learning.

#### 2.5.4. Artificial curiosity

Artificial curiosity is an idea from Schmidhuber, 1991 that aims to solve the problem of reinforcement learning with non-stationary environment requiring constant adaptation of RL system. In order to thrive, such systems must constantly adapt their behavior to shifts in distribution of their observations. Schmidhuber proposed an idea of artificial neural network that uses predictive error of learned world model as a source of intrinsic rewards. Such system would direct its actions toward states that are unpredictable and hence more surprising or interesting. As more novel states are visited, world model is refined and its prediction error goes down. Consequently, model is encouraged to visit subsequent interesting states allowing constant refinement of the world model and constant exploratory behavior of the RL system.

In his later work, Schmidhuber specified the minimum requirements for implementing artificial curiosity as:

1. Adaptive world model equipped with predictive capabilities, compressing knowledge accumulated from past experiences.
2. A learning algorithm that allows detecting novel patterns.
3. Intrinsic rewards generated in novel states.
4. An optimization which allows learning actions maximizing intrinsic rewards (Schmidhuber, 2010).

From an information-theoretic perspective, the central role of world models in artificial curiosity is to identify and exploit compressible structure in the data. A sequence of observations is said to contain a pattern if it can be described by a shorter code than the sequence itself, for example by predicting future components from past ones. In contrast, pure noise is incompressible and therefore uninformative: it cannot be predicted, learned, or used to improve behavior. For this reason, artificial curiosity emphasizes the existence of learnable patterns within the environment. Intrinsic rewards are generated not by arbitrary unpredictability, but by novelty that is discoverable regularities that were initially unknown to the agent yet can be learned and compressed into the world model. This ensures that curiosity-driven exploration is not wasted on randomness but is directed toward meaningful structure in the data (Schmidhuber, 2010). Conversely, failing to distinguish learnable patterns from noise leads to what is called "noisy TV problem" (Pathak et al., 2017). It is a phenomenon in which novelty-seeking agent is attracted to the random patterns within the observations and is unable to leave the vicinity of the source of randomness. Methods for dealing with this problem will be described in next chapter.

#### 2.5.5. Contents of internal representations within world models

World models are most often trained by optimizing reconstruction loss of encoded states visited by agent following actions sampled from policy.

$$\mathcal{L}_W = \frac{1}{N} \sum_{t=1}^N \|s_t - \hat{s}_t\|_2^2 \quad (2.11)$$

Where  $s_t$  are observed states at time  $t$  and  $N$  is number of experienced states within training batch. This and other analogous training objectives turn world model learning into supervised learning problem.

One method of extracting intrinsic rewards from these models are based on uncertainty measured by taking difference of predictions of the next state, and true values of these states. This requires constant access to the true states of the environment in order to generate intrinsic signal. Other family of methods utilize ensemble of world models and calculate variance of their predictions. This approach allows measuring model uncertainty without constant access to true states (Latyshev and Panov, 2023).

Depending on the design of world model encoded state can differ in its contents. For example pixel-reconstruction model aims to predict exact image representation of the next state given state-action pairs. Such objective leads to models that are sensitive to alterations of visual information leading to instability when changes in color, light etc. are introduced. On the other hand, architectures like Intrinsic Curiosity Module (ICM), learn inverse kinematics of the environment in order to ignore all information irrelevant to agents actions (Pathak et al., 2017). Features encoded by this model describe observations in terms of how agents behavior may affect them, which as a result leads to internal representations that remain stable when visual noise is encountered.

The meaning of surprisal, or predictive error, depends crucially on the nature of the state representation used by the world model. When states are encoded at the level of raw pixels, prediction error often reflects visual irregularities such as random noise, changes in lighting, or background motion. In contrast, when intrinsic rewards are derived from forward models such as the Intrinsic Curiosity Module (ICM), surprisal corresponds more directly to action-related uncertainty: the model is surprised when it cannot reliably predict the consequences of the agent’s own actions. Thus, the quality of internal representations determines whether predictive error signals are informative for exploration or are wasted on irrelevant variability.

Richer representations that incorporate higher-level features of the environment, or even information about the agent itself, could in principle support forms of self-awareness: the agent would not only model the external world, but also its own role and capacities within it. Some researchers have suggested that extending world models in this direction may eventually provide computational substrates for rudimentary self-introspection or even self-consciousness, where the agent’s predictions include both the environment and its own internal states (Schmidhuber, 1991).

### 2.5.6. Efficiency of internal representations

Richer world models that incorporate both external dynamics and aspects of the agent itself raise the question of how efficiently such information can be represented. The Fractured Entangled Representation (FER) hypothesis argues that deep learning methods based on stochastic gradient descent often learn unstable and overly complex internal representations. Such representations fail to exploit the symmetries and regularities present in the data Kumar et al., 2025. Kumar et al advocate for putting more emphasis on deep learning methods that are capable of Unified Factored Representation (UFR) learning. They show that compositional pattern producing networks (CPPN) (Stanley, 2007) can be used in conjunction with NEAT algorithm (Stanley and Miikkulainen, 2002) to learn internal representations that are simpler, modular, have greater capability to generalize and take better use of the regularities found in the data. Their findings expand the research on the world model quality by demonstrating that even though neural architectures can produce the same output given the same input, efficiency of their internal representations should also be taken into account.

It is also theorized that UFR is a consequence of a serendipitous discovery process. Algorithms that do not directly optimize any objective have a greater chance of learning solid and more versatile representations. This can be attributed to the fact that the greatest dis-

coveries do not happen as optimization towards the objective, but rather are consequences of undirected and unpredictable search process (Stanley and Lehman, 2015). InDRiVE architecture (Khanzada and Kwon, 2025) although not fully objective-less gives evidence that serendipitous pre-training process focused on intrinsic rewards and exploration may lead to better representations that allow fine-tuning for downstream tasks in more sample-efficient manner. Details of this study will be described in greater detail in the next chapter.

## Chapter 3

# Selected reinforcement learning algorithms utilizing world models

The use of world models in reinforcement learning has given rise to several distinct research traditions. One line of work, originating with Schmidhuber, 1991, focuses on curiosity and intrinsic motivation: agents exploit prediction errors or novelty signals from their world models to guide exploration in environments with sparse external rewards.

Another line, introduced by Sutton, 1991, employs world model dynamics for planning by generation of trajectories. This idea has since developed into two separate directions. The first, which will be referred to as learning from imagined trajectories, corresponds to what Sutton describes as background planning: the agent uses the model to imagine trajectories used to update the policy during training, but at inference time action selection is purely reactive. Recent advances such as Dreamer (Hafner et al., 2020) are an example this approach. The second direction is decision-time planning, where the agent explicitly constructs and evaluates action sequences using the world model during inference. Systems like MuZero (Schrittwieser et al., 2020) fall into this category.

Given these distinctions, the use of world models in reinforcement learning can be subdivided into three complementary applications: (1) curiosity and intrinsic motivation, (2) learning from imagined trajectories, (3) decision-time planning.

Therefore, the present chapter focuses on curiosity-driven exploration and learning from imagined trajectories, as these two approaches have not only matured substantially, but have also recently converged in novel architectures that integrate intrinsic motivation with imagination-based training (Khanzada and Kwon, 2025). In contrast, decision-time planning with learned world models is excluded from this discussion. This is partly for practical reasons of scope, but also because this line of work does not connect as naturally to the progression described above. Moreover, decision-time planning remains comparatively less developed, as it is rarely applied in settings where environment dynamics must be learned rather than assumed to be known (Schrittwieser et al., 2020). This is related to the fact that these methods are significantly less sample-efficient than learning from imagined trajectories (Hafner et al., 2020).

### 3.1. Intrinsic Curiosity Module

The Intrinsic Curiosity Module (ICM) was introduced by Pathak et al., 2017 as a mechanism for generating intrinsic rewards in reinforcement learning. Its design builds on the observation that prediction error in a learned world model can serve as a proxy for novelty: states that are

hard to predict are likely to be unexplored and thus interesting for the agent. This architecture with small modifications was used for experimental part of this work.

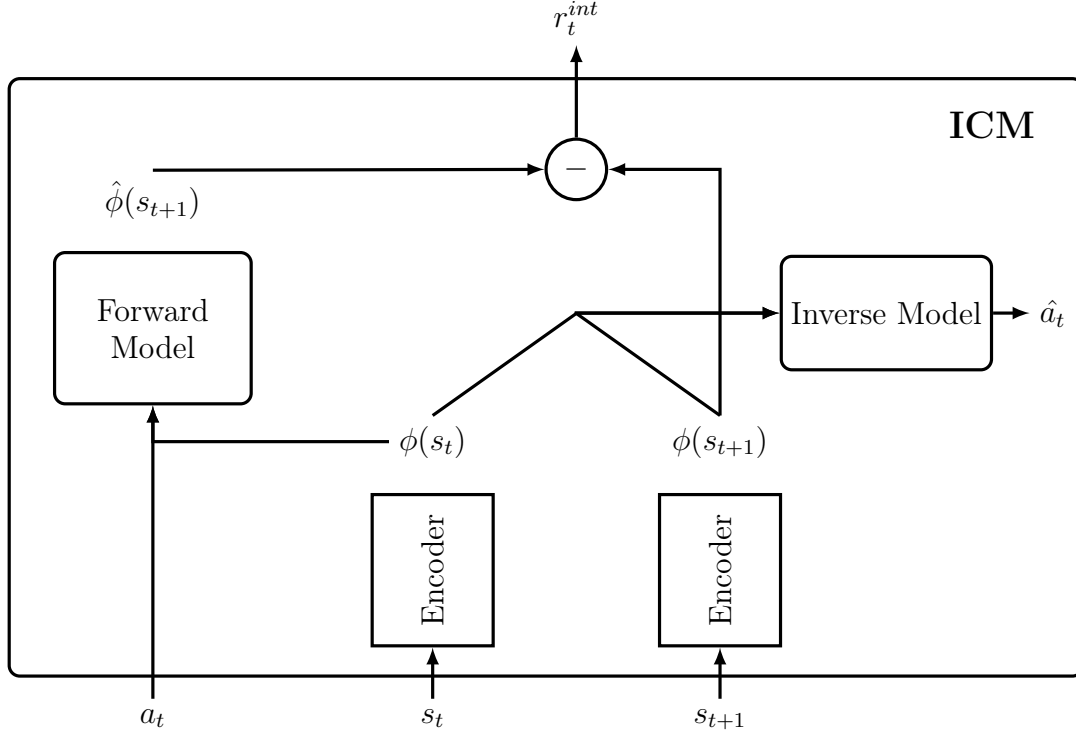


Figure 3.1: Schematic of the Intrinsic Curiosity Module (adapted from Pathak et al., 2017). Forward Model and Inverse Model are trained for different tasks while using the same encoder. This allows state representation to contain both action-relevant and state prediction information. Reward is computed as a difference of forward model predictions and encoded next state.

### 3.1.1. Architecture

The ICM consists of three components:

1. **Feature encoder:** maps high-dimensional observations (e.g., raw pixels) into a compact latent state representation  $\phi(s_t)$ .
2. **Inverse dynamics model:** predicts the action  $a_t$  taken between consecutive latent states  $(\phi(s_t), \phi(s_{t+1}))$ . This forces the representation to capture features of the environment that are controllable by the agent, helping to filter out irrelevant noise.
3. **Forward dynamics model:** predicts the next latent state  $\hat{\phi}(s_{t+1})$  given  $\phi(s_t)$  and  $a_t$ . The intrinsic reward is defined as the prediction error of this forward model, i.e.,

$$r_t^{int} = \frac{1}{2} \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|^2. \quad (3.1)$$

Both networks are optimized jointly with:

$$\mathcal{L}_{ICM} = \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}} \left[ \beta \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|^2 + (1 - \beta) \ell_{CE}(a_t, \hat{a}_t) \right], \quad (3.2)$$

Where  $\beta$  controls forward loss and inverse loss components importance,  $\ell_{CE}$  is a cross entropy loss. And  $\mathcal{D}$  is a distribution of experience accumulated by following policy. Authors suggest setting  $\beta = 0.2$ , which makes forward loss more significant to the gradient updates

### 3.1.2. Strengths and limitations

The key innovation of ICM is the use of the inverse dynamics model. By predicting the action between states, the encoder is encouraged to extract features that are controllable by the agent. This helps overcome the “noisy TV problem,” where an agent would otherwise be attracted to inherently unpredictable but uncontrollable stimuli.

Despite its advantages, ICM has several limitations:

- It requires access to the true environment states to train the inverse model, which may not always be available.
- The learned representation might not capture all the useful information, since some details that matter later may not seem relevant for action at the current state.
- Subsequent large-scale studies (Burda, Edwards, Pathak, et al., 2018) showed that ICM can sometimes underperform in some environments compared to simpler methods like Random Network Distillation (RND).
- The same study showcased that noisy TV still leads to significant training slow-down for both ICM and RND.

The relative performance of ICM and RND is therefore an open empirical question. This issue will be revisited in the experimental part of this thesis, where both methods are compared directly under controlled conditions.

## 3.2. Random Network Distillation

Random Network Distillation is another kind of world models introduced in Burda, Edwards, Storkey, and Klimov, 2018 useful for providing exploration bonus based on random features encoding. Idea behind the model is simple yet it leads to surprisingly good results. At the time of publishing it achieved state of the art results in “Montezuma’s Revenge” which is very hard Atari game in terms of exploration required (see Figure 3.2).

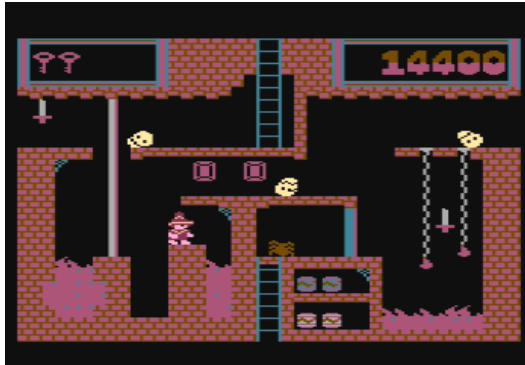


Figure 3.2: Screenshot from *Montezuma’s Revenge* (Atari 8-bit, PAL version). Source: Wikipedia/Wikimedia Commons.

### 3.3. Architecture

RND has two components, randomly initialized state encoder with frozen weights called target network, and predictor network that given the same input should predict output of the target network.

More formally, target network is a random function encoding states  $s_t$  into  $f(s_t)$ . Predictor with parameters  $\theta$  minimizes loss given by:

$$\mathcal{L}_\theta = \frac{1}{2} \mathbb{E}_{s_t \sim \mathcal{D}} [\|f(s_t) - \hat{f}(s_t; \theta)\|^2] \quad (3.3)$$

Where  $\mathcal{D}$  is distribution from which the states are drawn.

Intrinsic reward is generated with:

$$r_t^{int} = \frac{1}{2} \|f(s_t) - \hat{f}(s_t; \theta)\|^2 \quad (3.4)$$

It is designed to be added on top of extrinsic rewards so that the final reward expression equals:

$$r_t = r_t^{ext} + \eta r_t^{int}, \quad (3.5)$$

With  $\eta$  controls magnitude of intrinsic rewards.

### 3.4. Strengths and limitations

The main strength of RND lies in its conceptual simplicity and robustness. Unlike methods such as ICM, it does not require learning a forward dynamics model of the environment or an explicit inverse model. Instead, the target network is randomly initialized and fixed, which eliminates co-adaptation problems between model components. This simplicity makes RND easy to implement and train, while still producing strong exploration behavior in sparse-reward domains such as Atari (Burda, Edwards, Storkey, and Klimov, 2018). Another practical advantage is that the target network is deterministic and within the function class of the predictor, so prediction errors cannot arise from environmental stochasticity alone. This helps reduce, but not fully eliminate, the tendency to assign high intrinsic rewards to uncontrollable noise sources.

However, RND is not immune to the “noisy TV problem.” Large-scale studies have shown that stochastic inputs such as random pixels can still disrupt learning, sometimes even more severely than with ICM (Burda, Edwards, Pathak, et al., 2018). In such cases, the predictor network continues to assign novelty bonuses to inherently unpredictable signals, distracting the agent from meaningful exploration. In addition, the intrinsic reward signal in RND gradually diminishes as the predictor converges on the target network, which can reduce the agent’s exploratory drive over long training horizons. Finally, RND provides no model of environment dynamics or semantics of the state space; it is purely a mechanism for novelty detection, which limits its use to guiding exploration rather than enabling planning or representation learning.

Despite these limitations, RND remains one of the most effective curiosity-driven methods due to its stability, low computational overhead, and strong empirical performance in challenging sparse-reward environments.

### 3.5. Dreamer

Dreamer (Hafner et al., 2020) is an architecture related to the learning-from-imagination area of world models research. Achieving state of the art results on multiple difficult tasks, it

became one of the most influential model-based architectures in RL. The following section aims to explain unique use of world models in Dreamer architecture while skipping details of implementations, which are a significant part of Dreamer’s success with versions 2 and 3, advancing the architecture with numerous improvements of varying sizes and importance (Hafner et al., 2022; Hafner et al., 2024). Additionally, with an example of InDRiVE architecture it shows possibility of implementing RL system that combines both intrinsic rewards and learning from imagination in order to overcome Dreamer architecture limitation of weak exploration capabilities.

### 3.5.1. Architecture

Similarly to Dyna architecture (Sutton, 1991), Dreamer learns world model to enable exploring and learning from imagined trajectories (Hafner et al., 2020). What makes dreamer unique is capability to fully learn policy in latent space. The system learns to encode observations into the latent space and then apply actions sampled from actor network in order to advance to the next latent state through latent dynamics model, for each state in latent space a reward is generated using designated reward model.

World model is a recurrent state-space model (RSSM) (Hafner et al., 2019) learning to encode observation into the latent space, learning the transition function by optimizing reconstruction loss of a next state, and learning to predict rewards for a given state using reward model. Policy is trained only using states provided by the world model.

As in other actor-critic based approaches, Dreamer simultaneously learns to predict the value of all states using critic network, and then chooses actions that maximize estimate of values with an actor network. What makes dreamer unique is ability to execute those steps in latent space. Critic network minimizes error of its prediction and rewards generated by reward component of the world model. On the other hand, actor network gradient requires overcoming the issue of sampling actions during trajectory generation. This is made possible using reparameterization trick (Kingma and Welling, 2022) for continuous actions and straight-through gradients (Bengio et al., 2013) for discrete actions. In this way gradient information is not lost during state transition and gradients can be backpropagated.

More formally, the Recurrent State-Space Model (RSSM) used in Dreamer maintains two types of information at each step  $t$ : a deterministic hidden state  $h_t$  and a stochastic latent variable  $z_t$ . The inclusion of both deterministic and stochastic components addresses complementary needs:  $h_t$  enables stable propagation of information across long horizons, while  $z_t$  captures uncertainty in the dynamics. Together, they define the latent state  $s_t = (h_t, z_t)$ . The deterministic state is updated from the previous step as

$$h_t = f_{\text{det}}(h_{t-1}, z_{t-1}, a_{t-1}). \quad (3.6)$$

On top of  $h_t$ , the model defines two distributions for the stochastic component  $z_t$ :

- the **posterior**  $q_\phi(z_t | h_t, o_t)$ , which incorporates the current observation  $o_t$  and is used during training,
- the **prior**  $p_\theta(z_t | h_t)$ , which ignores  $o_t$  and is used during imagination rollouts.

During training,  $z_t$  is sampled from the posterior, while a KL divergence term

$$\text{KL}(q_\phi(z_t | h_t, o_t) \| p_\theta(z_t | h_t))$$

is minimized to align the prior with the posterior. This ensures that the prior becomes a good predictor of  $z_t$  without access to  $o_t$ , making imagined trajectories realistic while grounding training in real observations.

The world model is optimized by maximizing a variational evidence lower bound (ELBO), which combines three objectives:

1. **Reconstruction loss:** encourages latent states  $s_t$  to contain information sufficient to predict observations  $o_t$ ,
2. **Reward prediction loss:** ensures  $s_t$  encodes features relevant for control by predicting rewards  $r_t$ ,
3. **KL regularization:** aligns the posterior  $q_\theta$  with the prior  $p_\theta$ , preventing the stochastic component from drifting arbitrarily.

Formally, the model loss is written as

$$\mathcal{L}_{\text{model}} = -\mathbb{E} \left[ \sum_t \log q_\theta(o_t | s_t) + \log q_\theta(r_t | s_t) \right] + \beta \mathbb{E} \left[ \sum_t \text{KL}(q_\phi(z_t | h_t, o_t) \| p_\theta(z_t | h_t)) \right], \quad (3.7)$$

where  $\beta$  is a weight on the KL term. Observation and reward loss terms are marked with  $q$  to signify that during model training predictions are based on posterior state distribution.

For policy and value learning, Dreamer uses imagined trajectories  $s_{t:t+H}$  generated by rolling out the RSSM in latent space. The actor maximizes the expected value of imagined returns:

$$J_{\text{actor}} = -\mathbb{E} \left[ \sum_t \gamma^t \hat{V}(s_t) \right], \quad (3.8)$$

while the critic learns to approximate those returns:

$$J_{\text{critic}} = \mathbb{E} \left[ \sum_t (V_\psi(s_t) - \hat{V}(s_t))^2 \right]. \quad (3.9)$$

Here,  $\hat{V}(s_t)$  is a multi-step return estimate from rewards predicted by the model, and  $V_\psi(s_t)$  is the critic’s prediction of that return.

Crucially, both actor and critic are trained by backpropagation through the imagined trajectory. This allows the agent to improve its behavior using only latent imagination, reducing dependence on expensive environment interaction.

An important property of this formulation is that the world model objective can be readily swapped or extended without altering the overall training pipeline. In practice, the choice of reconstruction, reward, and KL regularization terms reflects assumptions about which aspects of the environment should be captured. For example, in visually rich environments, reconstruction terms may ensure perceptual fidelity. Alternative objectives, such as contrastive representation learning, auxiliary predictive tasks, or disentangled latent regularization, can replace or complement the ELBO terms. Since the actor and critic consume only latent states, they remain compatible with any world model objective, provided it yields a coherent and generalizable latent dynamics. This modularity makes Dreamer a flexible framework that can integrate advances in representation learning while retaining its latent imagination mechanism (Hafner et al., 2020).

### 3.5.2. Dreamer-V2 and V3

Dreamer-V2 and Dreamer-V3 extend the original Dreamer framework by refining the world model and improving its generalization (Hafner et al., 2022; Hafner et al., 2024). Both versions build on the idea of learning a latent dynamics model through a Recurrent State-Space Model (RSSM), but introduce changes that increase representational capacity and stability.

Dreamer-V2 (Hafner et al., 2022) reinterprets the world model as a form of sequential variational autoencoder (VAE). The model jointly learns a prior  $p_\theta(z_t | h_t)$  and a posterior  $q_\phi(z_t | h_t, o_t)$  over stochastic latent variables, while a deterministic recurrent state  $h_t$  ensures temporal consistency. Unlike the original Dreamer, Dreamer-V2 employs discrete latent variables, which improve expressiveness and reduce the tendency of stochastic variables to collapse. The world model is trained by minimizing a free-energy (variational ELBO) objective that balances three terms: observation reconstruction, reward prediction, and a KL divergence aligning prior and posterior distributions. This formulation yields a stronger and more stable latent representation, which allows the policy and value functions to be learned entirely from imagined trajectories in latent space. Interestingly, this formulation resonates with the Free Energy Principle proposed in cognitive science (Friston, 2010), which describes perception and action as processes of minimizing variational free energy. While Dreamer-V2 employs free-energy minimization in the narrow technical sense of optimizing a variational evidence lower bound (ELBO), the conceptual overlap highlights a shared theme: both artificial agents and biological systems can be understood as reducing prediction error by aligning internal models with external observations.

These improvements enabled Dreamer-V2 to achieve human-level performance on the Atari benchmark, matching or surpassing leading model-free methods within the same sample budget. The approach demonstrated that imagination-based training can rival purely reactive reinforcement learning even in visually complex domains.

Dreamer-V3 (Hafner et al., 2024) preserved the sequential VAE formulation but introduced a suite of normalization, balancing, and regularization techniques to make the method robust across tasks without hyperparameter tuning. With a single fixed set of hyperparameters, Dreamer-V3 was shown to outperform specialized algorithms on more than 150 diverse environments, including Atari, continuous control, and robotic tasks. Most notably, it became the first algorithm to solve the Minecraft “diamond” challenge, a long-horizon, sparse-reward problem requiring thousands of steps of planning and exploration. This milestone highlighted the scalability of Dreamer’s latent imagination to open-ended and complex environments.

### 3.5.3. Limitations

Despite their empirical success, Dreamer-V2 and V3 share an important limitation: weak exploration in sparse-reward settings. Neither variant integrates intrinsic motivation mechanisms such as curiosity or novelty bonuses. As a result, Dreamer agents rely solely on extrinsic rewards, which can lead to insufficient exploration when feedback is delayed or deceptive. This resonates with Schmidhuber’s critique of the Free Energy Principle (FEP) (Schmidhuber, 2010), where he argues that minimizing free energy leads agents to prefer predictability, not novelty. Dreamer can be seen as embodying a similar limitation: its training objective emphasizes stable prediction in latent space, which ensures efficient planning but does not inherently incentivize exploration. As a result, Dreamer agents, much like Friston’s active inference agents, are excellent at “explaining away” prediction errors but lack an intrinsic drive to seek out novel, learnable patterns. Furthermore, it can be hypothesized that regularization of latent posterior distribution toward the prior, although weakened by introduction of

KL-balancing, may lead to diminished exploratory behavior, as lower exploration may allow the model to minimize KL-divergence more easily.

Moreover, while the sequential VAE formulation stabilizes training, world model learning remains fragile in highly stochastic environments, where prediction errors reflect noise rather than learnable structure (Barsainyan et al., 2025).

### 3.5.4. InDRiVE: Combining Intrinsic Rewards with Imagination

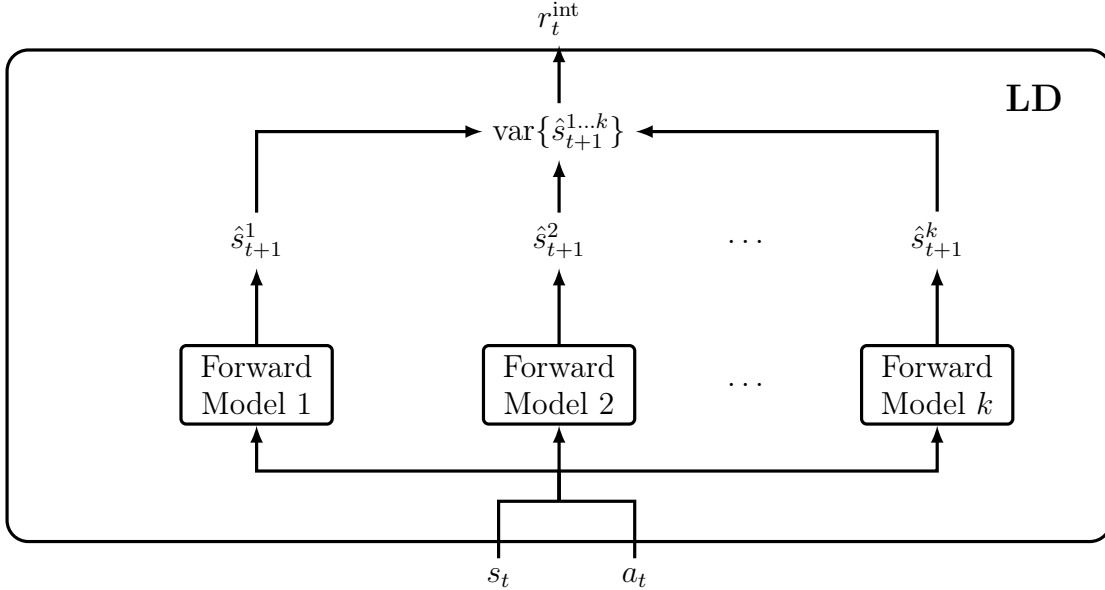


Figure 3.3: Latent Disagreement (LD) Reward adapted from Khanzada and Kwon, 2025. Intrinsic reward  $r_t^{\text{int}}$  is calculated based on variance between next state predictions  $\hat{s}_{t+1}$  generated by  $k$  independent world models.

To address Dreamer’s exploration limitations, recent work has explored augmenting latent imagination with intrinsic curiosity signals. The InDRiVE architecture (Khanzada and Kwon, 2025) builds directly on the Dreamer-V3 framework but introduces an initial exploration-driven phase. Here, extrinsic rewards are withheld, and the agent is trained solely on intrinsic signals derived from world model uncertainty. An ensemble of RSSMs predicts the next latent state, and the variance among their predictions is used as an intrinsic reward. This “latent disagreement” encourages the agent to explore uncertain and potentially informative regions of the environment.

After this intrinsic pre-training phase, extrinsic rewards are introduced, and the policy is fine-tuned to optimize task-specific objectives. This two-phase training procedure mitigates Dreamer’s weakness in sparse-reward environments and has been shown to yield more robust exploration and better sample efficiency. In experiments, InDRiVE achieved higher success rates in autonomous driving tasks compared to Dreamer-V2 and V3, while requiring significantly fewer training steps. Notably, it is the first study to train an ego-vehicle exclusively with intrinsic signals, demonstrating that uncertainty-driven exploration alone can produce a versatile, task-agnostic world model. The ease of fine-tuning suggests that InDRiVE may produce somewhat more general and robust representations, possibly aided by serendipitous discovery effects as discussed by Kumar et al., 2025. However, this connection is speculative, as the authors did not evaluate it directly, and further research would be needed to confirm

such dynamics.



## Chapter 4

# Experiments

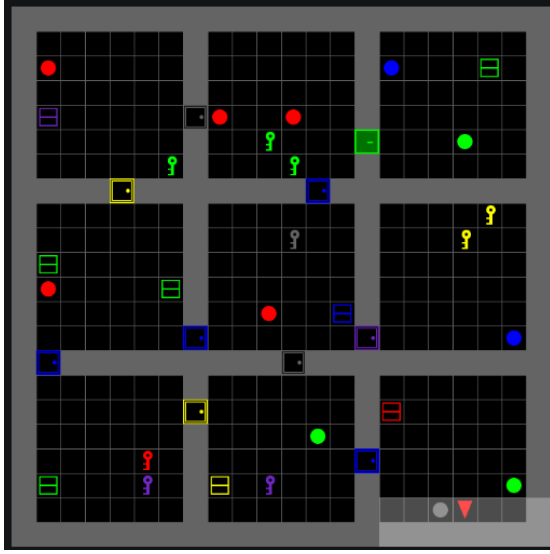
### 4.1. Method

#### 4.1.1. Training environment

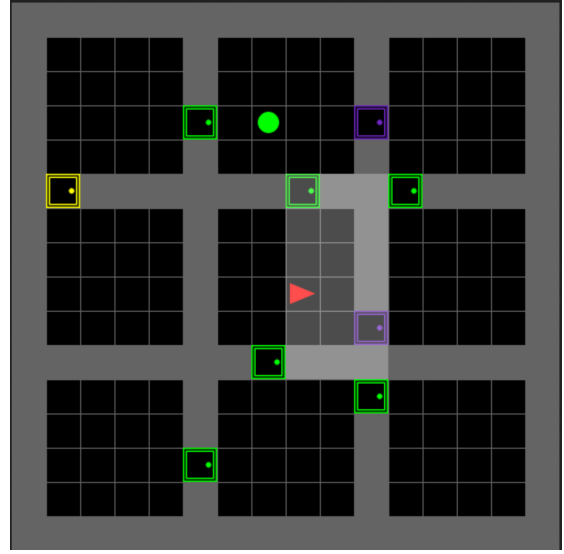
The experiments in this work were conducted using the MiniGrid library (Chevalier-Boisvert et al., 2023), a lightweight collection of reinforcement learning environments implemented within the OpenAI Gym interface (Towers et al., 2024). MiniGrid provides grid-world tasks with a discrete, low-dimensional observation space and a configurable degree of partial observability, making it suitable for testing exploration and sample-efficiency of RL algorithms. The library includes a range of benchmark tasks, from simple navigation problems to more complex multi-room mazes, which are made available through a set of default registered environments. These environments are widely used in the literature as standardized benchmarks for evaluating exploration strategies due to their controlled complexity and sparse reward structure suitable for research of exploration. BabyAI environments are a special subset of MiniGrid environments that were originally designed for research of human-in-the-loop training regimen utilizing basic language comprehension (Chevalier-Boisvert et al., 2019). Apart from the default 7x7 RGB view of the grid and information about the agents’ view direction, BabyAI environments contain instructions that define the goal of the task at hand. As my work is not focused on instructions comprehension and all agents were trained to perform in single environment, the textual data was ignored during the training phase. In addition, for simplicity of feature extraction, the directional information was also stripped as it was not crucial for navigating the static environment.

MiniGrid environments are varied in exploration difficulty and in terms of planning horizons required to successfully accomplish the tasks. As an example "BabyAI-Unlock-V0" requires visiting multiple separate rooms and opening specific doors in correct order. In preliminary experiments, 10 default environments were evaluated on the difficulty of exploration. Out of them, 2 ("BabyAI-KeyCorridorS3R1-v0", "BabyAI-FindObjS7-v0") were selected for further analysis as their difficulty allowed for short training times using limited computational resources. For all environments a single random instance controlled with random seed was used for all trainings in order to assure that all agents perform exactly the same task.

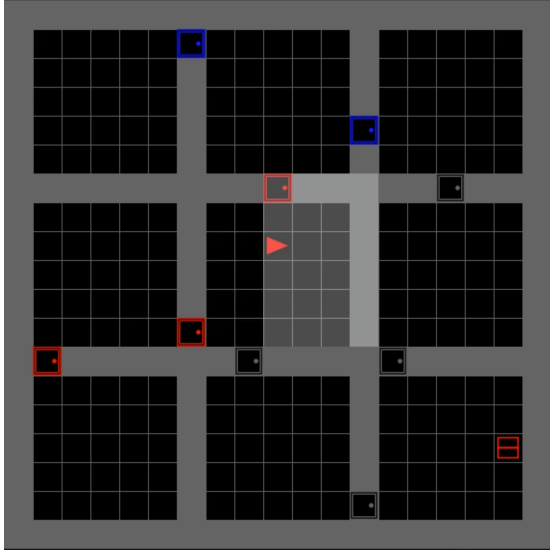
Exploration performance was measured based on minimum number of training steps required to reach optimal policy for a given task. Policy was evaluated based on average number of steps required to successfully perform the task in a single episode, the threshold for optimal policy depends on the task. If a model is unable to reach optimal policy in less than 1,000,000 steps, then its results are clipped to 1,000,000.



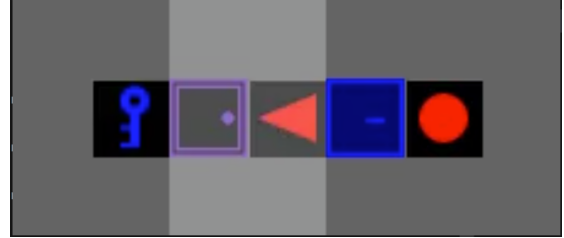
(a) Unlock



(b) Pick above



(c) Find Object



(d) Key Corridor

Figure 4.1: Example MiniGrid environments. "Key Corridor" was used in weight update skip experiment, "Find Object" was used in PPO/PPO+ICM/PPO+RND experiment.

#### 4.1.2. Model architecture

All agents use a Proximal Policy Optimization (PPO) backbone with separate actor and critic heads on a shared visual encoder (Schulman et al., 2017). The Intrinsic Curiosity Module (ICM) (Pathak et al., 2017) and Random Network Distillation (RND) (Burda, Edwards, Storkey, and Klimov, 2018) are integrated as optional, parallel modules that emit intrinsic rewards; PPO remains the sole policy optimization algorithm (ICM is used with PPO here instead of its original A3C pairing).

Image observations from MiniGrid/BabyAI are processed by a lightweight convolutional

encoder:

$$\text{Conv}(C \rightarrow 16, 2 \times 2) \rightarrow \text{ReLU} \rightarrow \text{Conv}(16 \rightarrow 32, 2 \times 2) \rightarrow \text{ReLU} \rightarrow \\ \text{Conv}(32 \rightarrow 64, 2 \times 2) \rightarrow \text{ReLU} \rightarrow \text{Flatten} \rightarrow \text{Linear}(\cdot \rightarrow 64) \rightarrow \text{ReLU}.$$

The resulting 64-dimensional feature vector feeds two linear heads: a categorical policy head (discrete actions) and a value head.

**ICM (world-model curiosity).** ICM has its own copy of the visual encoder (parameters not shared with PPO). It comprises of an inverse dynamics model and a forward model both implemented as two-layer MLPs with ReLU and width 64. Training minimizes ICM loss and provides intrinsic reward as described in previous chapter.

RND also uses an encoder separate from PPO. A frozen random target network  $T$  and a trainable predictor  $P$  share the same architecture (encoder  $\rightarrow$  MLP  $64 \rightarrow 64 \rightarrow 64$  with ReLU).

#### 4.1.3. Experimental setup

In preliminary experiments PPO agents were compared to PPO + ICM agents.

In the second experiment, a training procedure was designed to measure the impact of world model accuracy on exploration performance. ICM weight updates were systematically skipped according to the parameter  $skip\_prob \in [0, 1]$ . With values closer to 1, weight updates were skipped more frequently, leading to slower adaptation of the world model, whereas values closer to 0 resulted in nearly continuous updating. This manipulation created a controlled gradient of model accuracies, enabling an investigation of how different levels of predictive power influence the policy’s ability to explore and solve tasks efficiently. ”BabyAI-KeyCorridorS3R1-v0” environment was utilized for this experiment as it is simple enough to allow fast training times, yet difficult enough to require extensive exploration capabilities. In order to successfully perform designated task agents must open the violet door, pick up the key, open the locked blue door, drop the key, and lastly pick up the red ball. As corridor has limited space and agents cannot pass through objects, a perfect sequence of actions intertwined with careful movement must be performed to succeed.

#### 4.1.4. Training parameters and hardware

Agents were trained for 1,000,000 environment steps with PPO, PPO+ICM, or PPO+RND using the hyperparameters in Table 4.1. PPO used a small CNN feature extractor with linear actor and critic heads. PPO+ICM added a separate CNN encoder and two MLPs for the inverse and forward models, while PPO+RND used a predictor network trained against a fixed random target network.

All environments had maximum episode length 400 steps, discrete actions, and a fixed random seed of 42. BabyAI instructions and directional input were ignored.

Experiments were run on a workstation with an AMD Ryzen 7 5800H CPU and an NVIDIA GeForce RTX 3070 GPU. Eight agents were trained in parallel, with training times ranging from 15 minutes to 2 hours depending on task and convergence speed.

Table 4.1: Training and hyperparameter settings for PPO, PPO+ICM, and PPO+RND.

| <b>General PPO Settings</b>             |                    |
|-----------------------------------------|--------------------|
| Environment steps                       | 1,000,000          |
| Rollout steps                           | 1,024              |
| Epochs                                  | 100                |
| Minibatch size                          | 1,024              |
| Discount factor $\gamma$                | 0.99               |
| Clipping parameter $\epsilon$           | 0.2                |
| Entropy coefficient                     | 0.01               |
| Value loss coefficient                  | 0.5                |
| Optimizer                               | Adam               |
| Actor learning rate                     | $3 \times 10^{-4}$ |
| Critic learning rate                    | $10^{-3}$          |
| Feature extractor learning rate         | $10^{-3}$          |
| <b>PPO+ICM Specifics</b>                |                    |
| ICM learning rate                       | $10^{-3}$          |
| Forward model loss weight $\beta$       | 0.2                |
| Inverse model loss weight $(1 - \beta)$ | 0.8                |
| Intrinsic reward scale $\eta$           | 0.002              |
| <b>PPO+RND Specifics</b>                |                    |
| Predictor learning rate                 | $10^{-3}$          |
| Intrinsic reward scale $\eta$           | 0.2                |

## 4.2. Results

### 4.2.1. Preliminary experiment

Before running the main experiments, a set of ten MiniGrid and BabyAI environments was evaluated in order to identify tasks of appropriate difficulty for studying exploration. The selection included a spectrum from very simple navigation problems (MiniGrid-Empty-16x16-v0, MiniGrid-LavaCrossingS9N1-v0, BabyAI-PickupAbove-v0) to more complex multi-room tasks requiring sequential reasoning and long-horizon planning (BabyAI-Unlock, BabyAI-UnlockLocalDist, and BabyAI-KeyCorridor-S3R2/S3R3/S6R3).

Results showed that:

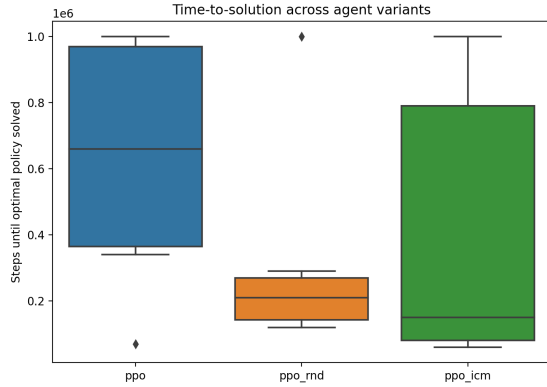
- In the simplest environments, exploration modules (ICM, RND) were unnecessary. PPO alone reliably solved the tasks faster, since intrinsic bonuses distracted from exploiting already dense reward signals.
- In the hardest environments, none of the agents (PPO, PPO+ICM, PPO+RND) were able to solve the tasks within one million environment steps. These tasks required exploration horizons that were beyond the feasible training budget.
- Two environments, BabyAI-FindObjS7-v0 and BabyAI-KeyCorridorS3R1-v0, struck a useful middle ground: sufficiently challenging to require systematic exploration, yet solvable within the allocated computational resources. These were selected for the main experiments reported in the following sections.

In addition, the preliminary runs were used to tune the intrinsic reward coefficient  $\eta$ , which scales the contribution of intrinsic bonuses relative to extrinsic rewards. Through trial-and-

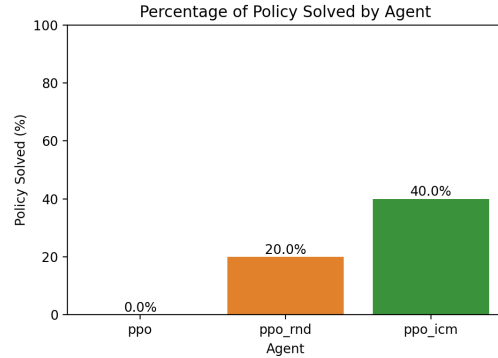
error, a value of  $\eta = 0.002$  was found to yield stable learning for ICM with large magnitude errors and  $\eta = 0.2$  for RND. This weighting makes intrinsic rewards five times less influential than extrinsic ones, ensuring that curiosity signals guide exploration without overwhelming the task objective.

#### 4.2.2. Plain PPO, RND and ICM comparison experiment

The second experiment compared the performance of plain PPO agents with two curiosity-augmented variants: PPO+RND and PPO+ICM. Two MiniGrid environments were selected for evaluation: "BabyAI-KeyCorridorS3R1-v0" and "BabyAI-FindObjS7-v0". These environments differ substantially in exploration difficulty. In Key Corridor, the task is solved by retrieving a ball placed behind two doors, requiring the agent to collect, use, and drop a key in the correct order. In contrast, Find Object requires navigating multiple rooms and picking up a specific object, posing a significantly harder challenge due to long horizons and lack of intermediate rewards.



(a) *Key Corridor*: time-to-solution across agents.



(b) *Find Object*: percentage of runs solving the task.

Figure 4.2: Comparison of PPO, PPO+RND, and PPO+ICM agents across two environments. In *Key Corridor*, most runs solved the task, with differences in sample efficiency. In *Find Object*, overall success rates were very low.

As shown in Figure 4.2, curiosity mechanisms significantly improved exploration in *Key Corridor*. Both PPO+ICM and PPO+RND agents converged to optimal policies much faster than plain PPO, with RND showing the most consistent sample efficiency. In *Find Object*, however, the results reveal the limitations of these methods. Plain PPO never solved the task across runs, PPO+RND succeeded only once out of five trials, and PPO+ICM twice. This low success rate reflects the difficulty of exploration in environments that require searching through multiple rooms without any shaping signals.

Interestingly, RND’s effectiveness diminished in *Find Object* because its intrinsic rewards quickly lost magnitude after the predictor network adapted to the simple statistics of the environment. In contrast, ICM maintained a stronger exploratory drive, but even here, the absence of dense feedback and the extended planning horizon made solving the task rare. These results highlight that while curiosity modules improve sample efficiency in moderately difficult environments, they remain insufficient for harder exploration problems requiring long-term memory and structured exploration.

### 4.2.3. Weight update skip experiment

Figure 4.3 illustrates three representative learning trajectories for "Key-Corridor-S3R1-v0" task. The green trace shows a failure case in which the agent does not discover a successful policy and episode lengths remain near the upper bound for most of training. The blue trace exemplifies a local minimum: after partial improvement, performance plateaus at  $\approx 30$  steps and fails to cross the optimal-policy threshold (22 steps) for roughly  $8.5 \times 10^5$  simulation steps. By contrast, the orange trace demonstrates rapid convergence: once informative exploratory behavior is discovered, mean episode length drops steeply and stabilizes at the optimal level. These examples clarify the qualitative modes underlying the aggregate analyses (regressions and distributions): non-convergence, suboptimal plateaus, and efficient convergence. Notably, two runs remained at local minimum for  $8 \times 10^5$  to  $9 \times 10^5$  steps. Most of the runs followed the trajectory of quick convergence occurring at different stages of training with quickest runs solving the optimal policy with 60 000 simulation steps.

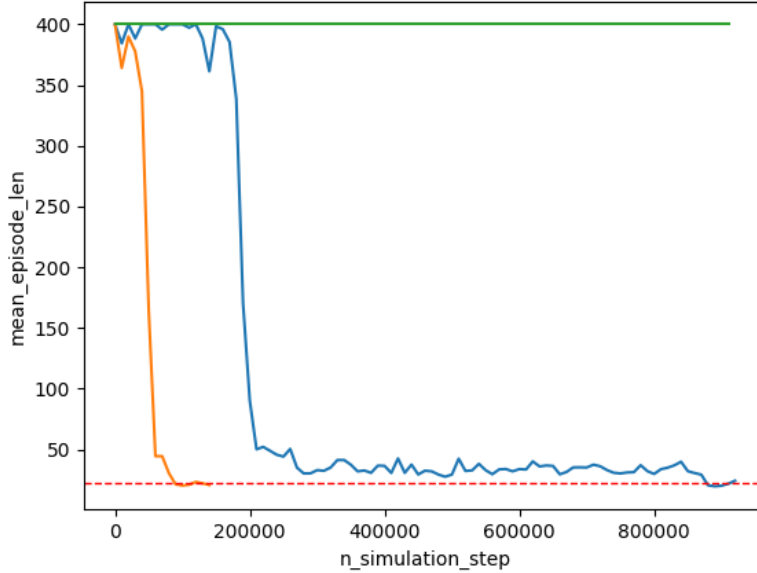


Figure 4.3: Three illustrative training runs in *Key Corridor*.  $y$ -axis: mean episode length (lower is better; max = 400).  $x$ -axis: simulation steps (not episodes). The horizontal red line marks the optimal-policy cutoff at 22 steps. Green: a run that fails to learn (episode length remains high). Blue: a run trapped in a local minimum around 30 steps for  $\sim 850,000$  steps. Orange: a run that converges quickly once exploratory reward is found.

To better understand the mechanism underlying the relationship between update probability and exploration, Figure 4.4 shows how ICM prediction error evolved during training. Agents with high update probability ( $p = 0.5, 0.8, 1.0$ ) rapidly decreased their prediction error, reflecting fast adaptation of the world model. By contrast, agents with lower update probability ( $p = 0.2, 0.3$ ) maintained elevated error levels over extended periods, indicating slower learning and weaker predictive capacity. The trajectory for  $p = 0.0$  illustrates the baseline, where prediction error remained consistently low due to the absence of updates, but at the cost of failing to capture novel structure in the environment. Together, these results confirm that update frequency directly shapes the quality of the world model and thereby the exploratory capacity of the agent. Notably, error seems to decrease with higher update

probability with lowest error achieved at  $p = 1$ . This however is not directly correlated with increased exploratory performance, that seems to reach optimal error at lower probabilities  $p = 0.5$  to  $0.9$

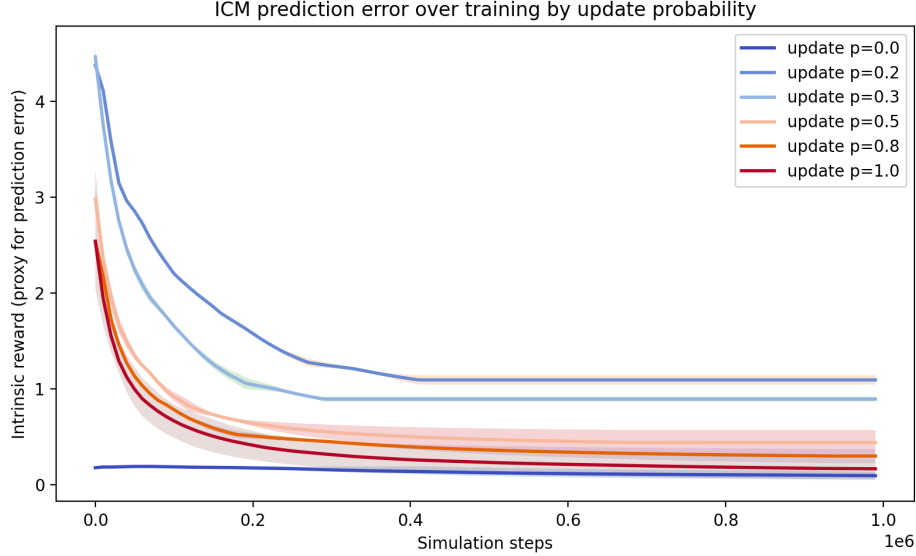


Figure 4.4: ICM prediction error (approximated by intrinsic reward) over training across world model update probabilities. Curves show the mean across seeds; shaded bands indicate interquartile range. Agents with higher update probability ( $p = 0.8, 1.0$ ) reduce prediction error more rapidly, while agents with lower update probability ( $p = 0.2, 0.5$ ) sustain higher error levels for longer, consistent with slower adaptation of the world model.

Two regression models were fitted to examine the effect of world model update probability on the number of steps required to reach an optimal policy. Quadratic terms were included to test hypothesis stating that improvements of world models will have diminishing effects. Diminishing returns would lead to U-shaped function that decreases number of states needed to learn policy until optimal level of world model learning is achieved, after that number of states needed would start to go up signifying diminishing effect of further world model learning.

First, using the untransformed dependent variable, both the linear ( $\beta = -9.90 \times 10^5$ ,  $p = .008$ ) and quadratic ( $\beta = 9.11 \times 10^5$ ,  $p = .014$ ) terms were statistically significant. This indicates that increasing the probability of updating world model weights accelerates policy learning up to a point, after which the benefit diminishes. The model explained approximately 12% of the variance ( $R^2 = .122$ ), and residuals showed marked positive skew.

Second, to reduce skewness, the dependent variable was log-transformed. In this specification, the linear effect remained significant ( $\beta = -2.47$ ,  $p = .017$ ), while the quadratic term showed only a marginal trend ( $\beta = 1.86$ ,  $p = .067$ ). This suggests that the primary effect of increased weight updating is to reduce solution time, with weaker evidence for diminishing returns. Model fit was slightly better ( $R^2 = .142$ ), and residual diagnostics indicated acceptable normality.

As shown in Figure 4.5, plotting the fitted quadratic curve against the observed data illustrates the general trend: policies are learned faster when the world model is updated more frequently, but the magnitude of improvement decreases at higher update probabilities.

Table 4.2: Regression of steps-to-solution on world model update probability.

| Predictor                              | $b$ (SE)           | 95% CI                 | $p$    |
|----------------------------------------|--------------------|------------------------|--------|
| <b>Model 1: Steps (raw)</b>            |                    |                        |        |
| Intercept                              | 464,500 (71,700)   | [321,000, 608,000]     | < .001 |
| Weight update probability              | -989,700 (358,000) | [-1,710,000, -272,000] | .008   |
| Weight update probability <sup>2</sup> | 910,500 (357,000)  | [195,000, 1,630,000]   | .014   |
| $R^2$                                  |                    | .122                   |        |
| Adj. $R^2$                             |                    | .090                   |        |
| $N$                                    |                    | 59                     |        |
| <b>Model 2: <math>\ln</math> Steps</b> |                    |                        |        |
| Intercept                              | 12.85 (0.20)       | [12.45, 13.25]         | < .001 |
| Weight update probability              | -2.47 (1.00)       | [-4.48, -0.46]         | .017   |
| Weight update probability <sup>2</sup> | 1.86 (1.00)        | [-0.14, 3.87]          | .067   |
| $R^2$                                  |                    | .142                   |        |
| Adj. $R^2$                             |                    | .111                   |        |
| $N$                                    |                    | 59                     |        |

Note. Unstandardized coefficients ( $b$ ). SE = standard error. 95% confidence intervals computed as  $b \pm 1.96(\text{SE})$ . Dependent variable: steps until optimal policy solved. Two-tailed  $p$  values.

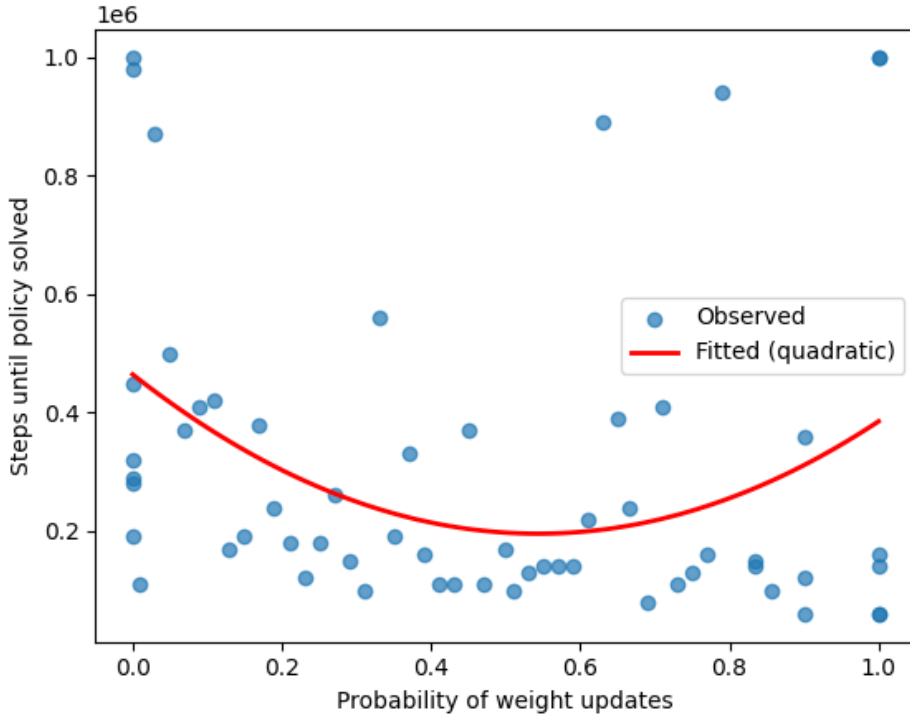


Figure 4.5: Quadratic regression of steps-to-solution on world model update probability. The red curve represents the fitted quadratic model; points indicate observed runs.

### 4.3. Discussion

The experiments reported in this thesis examined the relationship between the predictive accuracy of world models and exploration performance in reinforcement learning. Two main

observations emerge. First, curiosity-driven methods such as ICM and RND substantially improve exploration compared to plain PPO in moderately difficult environments (e.g., Key Corridor), but struggle in tasks that require extended horizons and sparse feedback (e.g., Find Object). Second, systematic manipulation of world model update frequency revealed that increasing predictive accuracy generally accelerates learning, but with diminishing returns. Regression analyses confirmed a significant linear benefit of more frequent updates, while the quadratic term indicated that improvements plateau at higher levels of accuracy.

One possible explanation for this diminishing effect is the relative simplicity of the chosen environment. In Key Corridor, the world model quickly achieves a sufficient level of predictive competence; beyond this point, additional accuracy yields little novel surprisal signal to guide exploration. As a result, the intrinsic reward diminishes and exploration stalls, even though the world model continues to improve. This dynamic reflects the theoretical concern raised by Schmidhuber: if predictions are too poor, everything appears novel and the intrinsic signal is uninformative; if predictions are too precise, no novelty is detected and exploration halts (Schmidhuber, 1991). Efficient exploration therefore requires a balance between predictive accuracy and surprisal, where prediction error remains meaningful without collapsing into either noise or triviality.

Taken together, the findings highlight the central role of predictive error not only as a technical signal for curiosity-driven exploration, but also as a conceptual bridge between reinforcement learning and cognitive science perspectives on adaptive behavior.

## 4.4. Future Research

This work highlights a narrow but consequential link between world-model accuracy and exploration performance. A broader research agenda—spanning three traditions of world-model use in RL—could yield larger gains:

The introductory theoretical analysis suggests that intrinsic rewards, learning from imagined trajectories and decision-time planning should be studied jointly rather than in isolation. Promising directions include: (a) intrinsic pretraining to shape world-model representations, followed by imagination-based policy learning (Dreamer-style) and (b) selectively invoking decision-time search when model uncertainty or disagreement crosses a threshold. A key open problem is how to regulate surprisal, and assure that novelty signals are sourced from meaningful representations. In general studies of representation learning should more closely combine their findings with research of novelty signals. Perhaps an undiscovered method for representation learning would allow continuous extraction, compression and learning of novel patterns.

Complementary to reward design is the quality of internal representations. The Fractured Entangled Representation (FER) hypothesis (Kumar et al., 2025) predicts that gradient-based optimization often yields entangled, brittle features; by contrast, Unified Factored Representations (UFR) would be sparse, modular, and symmetry-aware. Future work should couple curiosity-driven training with diagnostics for factorization and evaluate whether objective-less or weakly objective-guided exploration via intrinsic reward reliably pushes models toward UFR, compared to purely extrinsically motivated approaches. Possibly an experiment can be conducted whether world models or policy trained on purely intrinsic rewards allows more unified-factored representations compared to models based on extrinsic rewards. If intrinsic exploration allows serendipitous discovery, then hypothetically it could serve as a practical route to more generalizable world models. More efficient representations would also make learning novel patterns easier, combining this with a source of a good novelty signals might

bring us closer to life-long learning systems that allow learning novel patterns at varied levels of abstraction.

## Acknowledgements

The author acknowledges the use of AI-based tools for support with grammar, style, and clarity of expression, as well as assistance in producing selected figures. The core research, including literature review, theoretical framing, experimental design, implementation, analysis, and interpretation of results, was conducted comprehensively and independently by the author. The narrative structure and argumentation were also directed by the author, with AI tools serving only as supplementary aids in presentation.

# Bibliography

- Barsainyan, A. A., Lim, J. Y., & Liu, D. (2025). Toward a unified benchmark and taxonomy of stochastic environments. <https://arxiv.org/abs/2509.01793>
- Bellman, R. (1957). A markovian decision process. *Indiana University Mathematics Journal*, 6(4), 679–684. <https://doi.org/10.1512/iumj.1957.6.56038>
- Bellman equation [Accessed from Wikipedia]. (2025). [https://en.wikipedia.org/w/index.php?title=Bellman\\_equation&oldid=1297732794](https://en.wikipedia.org/w/index.php?title=Bellman_equation&oldid=1297732794)
- Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. <https://arxiv.org/abs/1308.3432>
- Burda, Y., Edwards, H., Pathak, D., Storkey, A., Darrell, T., & Efros, A. A. (2018). Large-scale study of curiosity-driven learning. <https://doi.org/10.48550/arXiv.1808.04355>
- Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2018). Exploration by random network distillation. <https://arxiv.org/abs/1810.12894>
- Chevalier-Boisvert, M., Bahdanau, D., Lahlou, S., Willems, L., Saharia, C., Nguyen, T. H., & Bengio, Y. (2019). Babyai: A platform to study the sample efficiency of grounded language learning. <https://arxiv.org/abs/1810.08272>
- Chevalier-Boisvert, M., Dai, B., Towers, M., de Lazcano, R., Willems, L., Lahlou, S., Pal, S., Castro, P. S., & Terry, J. (2023). Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, abs/2306.13831.
- Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3), 181–204. <https://doi.org/10.1017/S0140525X12000477>
- Craik, K. J. W. (1943). Chapter 3. In *The nature of explanation*. Cambridge University Press.
- Francois-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *Foundations and Trends in Machine Learning*, 11(3–4), 219–354. <https://doi.org/10.1561/22000000071>
- Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.
- Ha, D., & Schmidhuber, J. (2018). World models. <https://doi.org/10.5281/ZENODO.1207631>
- Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2020). Dream to control: Learning behaviors by latent imagination. <https://doi.org/10.48550/arXiv.1912.01603>
- Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., & Davidson, J. (2019). Learning latent dynamics for planning from pixels. <https://arxiv.org/abs/1811.04551>
- Hafner, D., Lillicrap, T., Norouzi, M., & Ba, J. (2022). Mastering atari with discrete world models. <https://arxiv.org/abs/2010.02193>
- Hafner, D., Pasukonis, J., Ba, J., & Lillicrap, T. (2024). Mastering diverse domains through world models. <https://arxiv.org/abs/2301.04104>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>

- Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. <https://arxiv.org/abs/2006.11239>
- Hohwy, J., Roepstorff, A., & Friston, K. (2008). Predictive coding explains binocular rivalry: An epistemological review. *Cognition*, 108(3), 687–701. <https://doi.org/10.1016/j.cognition.2008.05.010>
- Hosoya, T., Baccus, S. A., & Meister, M. (2005). Dynamic predictive coding by the retina. *Nature*, 436(7047), 71–77. <https://doi.org/10.1038/nature03689>
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. <https://arxiv.org/abs/1502.03167>
- Jiang, M., Rocktäschel, T., & Grefenstette, E. (2022). General intelligence requires rethinking exploration. *arXiv preprint arXiv:2211.07819*. <https://arxiv.org/abs/2211.07819>
- Khanzada, F. K., & Kwon, J. (2025). Indrive: Intrinsic disagreement based reinforcement for vehicle exploration through curiosity driven generalized world model. <https://arxiv.org/abs/2503.05573>
- Kingma, D. P., & Welling, M. (2022). Auto-encoding variational bayes. <https://arxiv.org/abs/1312.6114>
- Konda, V., & Tsitsiklis, J. (1999). Actor-critic algorithms. In S. Solla, T. Leen, & K. Müller (Eds.), *Advances in neural information processing systems* (Vol. 12). MIT Press. [https://proceedings.neurips.cc/paper\\_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf)
- Kumar, A., Clune, J., Lehman, J., & Stanley, K. O. (2025). Questioning representational optimism in deep learning: The fractured entangled representation hypothesis [No. arXiv:2505.11581]. <https://doi.org/10.48550/arXiv.2505.11581>
- Latyshev, A., & Panov, A. I. (2023). Intrinsic motivation in model-based reinforcement learning: A brief review. <https://doi.org/10.48550/arXiv.2301.10067>
- Markov decision process [Accessed from Wikipedia]. (2025). [https://en.wikipedia.org/w/index.php?title=Markov\\_decision\\_process&oldid=1297575772](https://en.wikipedia.org/w/index.php?title=Markov_decision_process&oldid=1297575772)
- Milkowski, M. (2013). Explaining the computational mind. In *Explaining the computational mind* (pp. 85–131). MIT Press.
- Pathak, D., Agrawal, P., Efros, A. A., & Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction. <https://doi.org/10.48550/arXiv.1705.05363>
- Rensink, R. A., O'Regan, J. K., & Clark, J. J. (1997). To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8(5), 368–373.
- Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, 222–227.
- Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3), 230–247. <https://doi.org/10.1109/TAMD.2010.2056368>
- Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., & Silver, D. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839), 604–609. <https://doi.org/10.1038/s41586-020-03051-4>
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, July). Trust region policy optimization. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (pp. 1889–1897, Vol. 37). PMLR. <https://proceedings.mlr.press/v37/schulman15.html>
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. <https://arxiv.org/abs/1707.06347>

- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. <https://arxiv.org/abs/1712.01815>
- Simon, H. A. (1957). *Models of man; social and rational*. Wiley.
- Simons, D. J., & Chabris, C. F. (1999). Gorillas in our midst: Sustained inattention blindness for dynamic events. *Perception*, 28(9), 1059–1074. <https://doi.org/10.1068/p281059>
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2), 131–162. <https://doi.org/10.1007/s10710-007-9028-8>
- Stanley, K. O., & Lehman, J. (2015). *Why greatness cannot be planned: The myth of the objective*. Springer Publishing Company, Incorporated.
- Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2), 99–127. <https://doi.org/10.1162/106365602320169811>
- Sutton, R. S. (1991). Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bull.*, 2(4), 160–163. <https://doi.org/10.1145/122344.122377>
- Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 88(2), 135–170.
- Sutton, R. S., & Barto, A. G. (2020). *Reinforcement learning: An introduction* (Second). MIT Press.
- Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, & K. Müller (Eds.), *Advances in neural information processing systems* (Vol. 12). MIT Press. [https://proceedings.neurips.cc/paper\\_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf)
- Sutton, R. S., & Pinette, B. (1985). The learning of world models by connectionist networks. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society (CogSci)*, 54–64.
- Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., Cola, G. D., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J. J., Tan, H., & Younis, O. G. (2024). Gymnasium: A standard interface for reinforcement learning environments. <https://arxiv.org/abs/2407.17032>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. <https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- Whitney, W. F., & Fergus, R. (2019). Understanding the asymptotic performance of model-based rl methods. *arXiv preprint arXiv:1905.12310*.