ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

Second Cycle Degree in Computer Science and Engineering

Online Robot Adaptation by Actor-Critic Reinforcement Learning

Master Thesis in:
Intelligent Robotic Systems

Supervisor
Prof. Andrea Roli

Candidate
Pablo Sebastian Vargas Grateron

IV Session
Academic Year 2024-2025

Abstract

Robots in real-world environments must adapt their strategies as conditions change. This thesis investigates online strategy adaptation using an Actor-Critic Reinforcement Learning framework optimized by a Genetic Algorithm (GA). By evolving meta parameters and initial network weights over 50 generations, the agent achieves an optimal balance between exploration and exploitation. Using the ARGoS3 simulator, the research evaluates two scenarios: an energy-driven survival task and a multi-headed architecture for contextual rewards based on visual stimuli. Results show that combining learning and evolution enables robots to autonomously adapt to non-stationary environments.

"It's not magic that makes me strong.
It's the fact that I never stop trying."

— Marisa Kirisame, *Touhou Project*

Contents

List of Figures

List of Tables

Introduction

Modern robotic systems operating in everyday, real-world environments must possess an inherent ability to adapt their strategies to shifting external conditions. Conventional control methodologies and offline training approaches often rely on pre-trained models optimized for static, highly controlled settings. While effective in predictable scenarios, these static policies are inherently brittle when deployed in non-stationary environments, where rules, resource availability, and environmental constraints change unexpectedly. To overcome these limitations and manage persistent uncertainty, robots require a capacity for *online adaptation*, the ability to autonomously learn and modify their behaviour in real-time based on continuous interaction with their surroundings.

This thesis investigates online strategy adaptation through a hybrid Evolutionary Reinforcement Learning framework. Specifically, it merges an Actor-Critic Reinforcement Learning (RL) paradigm with Evolutionary Computation. Within this framework, an actor neural network continuously generates actions, while a critic network evaluates the effectiveness of those choices via a Temporal Difference error signal. This dual structure enables the robot to learn optimal policies without the need for predefined environmental models. However, the efficiency of this online learning is critically dependent on meta-parameters, such as learning rates and exploration thresholds. To solve this bottleneck, a Genetic Algorithm (GA) is employed to optimize the ideal relationship between learning parameters, neural architectures, and robot performance, evolving initial configurations to ensure both structural flexibility and rapid behavioural convergence.

The validity of this approach is tested and analysed through two distinct non-stationary experimental scenarios conducted within a robotic simulator. The first experiment replicates a survival task where an agent must manage its internal

energy resources. The robot must autonomously shift its behavioural strategy between active foraging and conservative waiting based on fluctuating battery densities and movement costs, effectively managing the exploration-exploitation dilemma. The second experiment introduces an original multi-headed neural architecture designed to solve complex contextual reward problems involving coloured visual stimuli. In this scenario, the utility of a target changes radically depending on the environmental phase. This specifically tests the agent's cognitive flexibility and its ability to avoid *catastrophic forgetting* when rapidly "unlearning" obsolete rules to adapt to new reward contingencies.

Ultimately, this work aims to demonstrate how the combination of structural specialization and evolutionary meta-parameter optimization allows a robotic agent to autonomously navigate, survive, and successfully adapt to the profound complexities of non-stationary tasks.

Structure of the Thesis This thesis is organized into five chapters, providing a comprehensive overview of the transition from theoretical foundations to experimental validation and comparative analysis:

- **Chapter 1: Theoretical Background** provides the fundamental concepts required to understand the research. It covers the principles of RL, focusing on the Actor-Critic framework, the challenges of non-stationary environments, and the integration of evolutionary computation to optimize learning processes.

- **Chapter 2: Experimental Methodology** describes the simulation environment and the design of the non-stationary tasks used for evaluation. It details the training process, specifically the configuration of the GA used to evolve the controllers' parameters and structures.

- **Chapter 3: Proposed Architectures** presents the technical design of the control systems. It describes the implementation of standard adaptive models and the introduction of a specialized multi-headed structural approach for handling severe environmental shifts.

- **Chapter 4: Results and Discussion** provides a comprehensive analysis of the data collected during the simulations. It evaluates the evolutionary dynamics, the performance of the proposed models, and offers a comparative discussion on their ability to adapt online across unseen testing environments.
- **Chapter 5: Conclusions and Future Work** summarizes the main findings and contributions of the thesis. It reflects on the achievement of the research goals and suggests potential directions for further development in the field of adaptive autonomous robotics.

Chapter 1

Theoretical Background

The design of autonomous robotic systems has historically relied on explicit, rule-based programming or offline training phases conducted in highly controlled, static environments. While these conventional approaches yield high performance in predictable and stationary scenarios, they inherently struggle when deployed in the real world. Real-world environments are fundamentally non-stationary: external conditions change unexpectedly, resource availability fluctuates, and previously optimal behaviours can suddenly become obsolete or even detrimental to the agent's survival.

To operate effectively under such persistent uncertainty, modern robotic systems must move beyond rigid pre-programming and static models. They require an inherent capacity for online adaptation, the ability to continuously learn, evaluate, and autonomously modify their decision-making policies in real-time based solely on their ongoing interaction with the environment. Addressing this challenge necessitates a departure from classic control paradigms, leaning instead towards bio-inspired and learning-based methodologies.

This chapter establishes the theoretical foundations required to understand how such adaptive behaviour can be computationally synthesized. It begins by exploring the principles of Evolutionary Robotics (ER) [NF00], a bio-inspired methodology that automates the bottom-up design of neural controllers through natural selection. Subsequently, it introduces the paradigm of Reinforcement Learning (RL) [SB05], with a specific focus on the Actor-Critic architecture, which pro-

vides the mathematical framework for lifetime learning and policy optimization. Finally, it examines how the hybridization of these two fields provides a robust mechanism to manage the continuous exploration-exploitation dilemma, enabling autonomous agents to successfully navigate and survive the complexities of strictly non-stationary environments.

1.1 Evolutionary Robotics

The design of autonomous controllers has conventionally relied on explicit, top-down programming, where human developers manually define rules and behaviours for specific tasks. However, this approach is often constrained by human design biases and struggles to anticipate the infinite complexities of unstructured environments. Evolutionary Robotics (ER) proposes a radically different, bio-inspired methodology [NF00]. By applying the Darwinian principles of natural selection, ER automates the synthesis of robotic controllers—typically in the form of Artificial Neural Networks (ANNs)-based solely on the agent's embodied interaction with its environment.

In a standard ER framework, the optimization process is driven by a Genetic Algorithm (GA). Instead of programming a single deterministic solution, a population of pseudo-randomly initialized controllers is generated and evaluated iteratively. Each individual in the population is scored according to a fitness function, a high-level metric that rewards desirable autonomous behaviours (such as efficient energy gathering or target discrimination) without explicitly dictating how those behaviours should be mechanically achieved. The highest-performing controllers are then selected to produce the next generation of agents through bio-inspired operators such as synaptic mutation and structural crossover.

This bottom-up approach is highly effective at discovering robust, non-intuitive sensory-motor mappings. By exploring a vast, high-dimensional parameter space, evolution can identify optimal network topologies and baseline synaptic weights that are perfectly tailored to the robot's specific hardware constraints.

1.2 Reinforcement Learning

While evolutionary algorithms excel at discovering robust neural topologies and baseline behaviours offline, autonomous agents deployed in the real world must also possess the capacity to learn and adapt during their own lifetime. This requirement introduces the paradigm of *Reinforcement Learning* (RL).

As comprehensively formalized by Sutton and Barto in their seminal work "Reinforcement Learning: An Introduction" [SB05], RL allows an agent to learn an optimal mapping from situations to actions through direct, trial-and-error interaction with its environment. Formally, the RL problem is framed as a **Markov Decision Process** (MDP). At each discrete time step t, the agent observes a representation of the environment's state $S_t \in \mathcal{S}$ and selects an action $A_t \in \mathcal{A}(s)$ based on its internal policy $\pi(a|s)$, which defines the probability of selecting action a in state s. In response to this action, the environment transitions to a new state S_{t+1} and yields a numerical reward signal $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$.

Rather than being explicitly instructed on which actions to take, the agent's sole objective is to maximize the total amount of reward it receives over the long run. Mathematically, the agent seeks to maximize the **expected return** G_t, defined as a specific function of the reward sequence. For continuous or long-term tasks, this is typically formulated using the discounted return:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1.1}$$

where $\gamma \in [0, 1]$ is the *discount factor*, a parameter that determines the present value of future rewards [SB05]. To achieve this maximization, the agent must estimate the **value function** $V_\pi(s)$, which defines the expected return starting from state s and strictly following policy π thereafter:

$$V_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s \right] \tag{1.2}$$

A fundamental and persistent challenge within this framework is the **exploration-exploitation dilemma** [SB05]. To maximize its cumulative reward, an agent must exploit actions that it has previously evaluated and found to be effective.

However, to discover these optimal actions in the first place, it must concurrently explore the environment by selecting novel, untried behaviours.

In strictly non-stationary environments such as the robotic scenarios addressed in this thesis, where resource availability drops or contextual colour rules change unexpectedly, a static approach to this dilemma is inherently brittle. To survive, the agent must employ dynamic mechanisms to continuously balance the exploitation of known resources with the active exploration of new strategies, ensuring it can swiftly adapt whenever the environmental contingencies shift.

1.3 The Actor-Critic Architecture

Within the broad landscape of Reinforcement Learning algorithms, the Actor-Critic architecture stands out for its stability and efficiency, particularly when operating in the continuous and complex state spaces typical of robotics. As comprehensively analysed by Ivo Grondman et al. in their foundational paper *"A survey of actor-critic reinforcement learning: Standard and natural policy gradients"* [GBLB12], and mathematically formalized by Konda and Tsitsiklis in *"Actor-critic algorithms"* [KT02], this architecture explicitly separates the learning and decision-making processes into two distinct neural components:

- **The Actor:** This unit maintains and updates the robot's stochastic policy. Given a state vector s, the Actor generates preferences (logits) for each available action P_a. These preferences are then converted into actual selection probabilities using a Soft-max function, modulated by the inverse temperature parameter β:

$$P(a|s) = \frac{e^{\beta \cdot P_a}}{\sum_j e^{\beta \cdot P_j}} \tag{1.3}$$

  This stochastic mechanism is essential for managing the exploration-exploitation dilemma: a high value of β leads to deterministic exploitation, whereas a low value encourages the exploration of new actions.

- **The Critic:** The Critic does not select actions; rather, it acts as an evaluator of the current policy by estimating the value function $V(s)$. After the Actor

executes an action, the Critic evaluates the quality of the newly reached state by calculating the prediction error, known as the **Temporal Difference (TD) error**, denoted by δ.

In real-world robotic systems, different actions (such as moving towards a target or waiting) may require varying execution times. To maintain learning consistency, the canonical TD-error formulation is extended by incorporating the actual temporal duration of the action Δt:

$$\delta = r + \gamma^{\Delta t} \cdot V(s_{next}) - V(s) \tag{1.4}$$

In this equation, r represents the immediate reward (or penalty), and $\gamma \in [0, 1]$ is the temporal discount factor.

The δ signal generated by the Critic acts as the core driver for online adaptation, guiding the simultaneous update of both networks. For instance, the synaptic weights w_i of the Critic are updated proportionally to the learning rate α to minimize the value estimation error:

$$w_i \leftarrow w_i + \alpha \cdot \delta \cdot s_i \tag{1.5}$$

Concurrently, the Actor utilizes the same δ signal to update its action preferences: if $\delta > 0$, the action proved better than expected, and its selection probability is reinforced; conversely, if $\delta < 0$, the action is discouraged.

Despite the immense adaptive potential of the Actor-Critic framework, its convergence efficiency critically depends on the proper balancing of its meta-parameters, such as the learning rates (α) and the discount factor (γ). Manually optimizing these values for non-stationary environments is a prohibitive and often suboptimal task, thereby necessitating the integration of global evolutionary optimization techniques.

1.4 Evolutionary Reinforcement Learning

While the Actor-Critic architecture provides a mathematically sound framework for lifetime adaptation, its practical success is heavily bottlenecked by the configu-

ration of its learning meta-parameters. Parameters such as the learning rates (α), the temporal discount factor (γ), and the exploration-exploitation temperatures (β) strictly dictate the speed, stability, and foresight of the learning process. In complex robotic applications, manual tuning of these hyper-parameters is often a prohibitive and largely sub-optimal task, as the ideal configuration heavily depends on the specific hardware dynamics and environmental constraints.

To overcome this fundamental limitation, Evolutionary Computation can be deeply integrated with RL, creating a hybrid paradigm often referred to as Evolutionary Reinforcement Learning (ERL). As demonstrated by Capi and Doya in their pivotal study *"Application of Evolutionary Computation For Efficient Reinforcement Learning"* [CD06], GA can be effectively employed to search the high-dimensional, continuous space of RL parameters.

Instead of relying on human intuition or exhaustive grid searches, the evolutionary process treats the RL meta-parameters, and crucially, the initial synaptic weights and network topologies, as a genetic sequence (genotype). A population of robotic controllers is instantiated with pseudo-random parameter configurations and evaluated over a simulated lifetime. The fitness of each individual is determined by its overall RL performance, such as its ability to efficiently gather rewards or maintain energy reserves. Through successive generations of selection, crossover, and mutation, the GA prunes failing strategies and converges upon the optimal learning configuration tailored to the specific task.

This hybridization effectively bridges the gap between phylogenetic (cross-generational) and ontogenetic (lifetime) adaptation. Evolution endows the agent with a highly optimized "innate" predisposition, fine-tuned learning rates and favourable initial weights, while Reinforcement Learning leverages this starting point to achieve rapid and highly efficient policy acquisition during the agent's life. Ultimately, synthesizing these meta-parameters via evolutionary optimization is not merely an enhancement, but a critical prerequisite for developing agents capable of surviving the unpredictable and rapid shifts inherent to non-stationary environments.

1.5 Adaptation in Non-Stationary Environments and Catastrophic Forgetting

While standard Reinforcement Learning algorithms are highly effective in static scenarios, robots operating in everyday, real-world applications often face non-stationary environments, where rules, constraints, or resource locations change unexpectedly. Surviving such volatility requires the agent to carefully manage the exploration-exploitation dilemma. A static exploration strategy is inherently brittle under dynamic conditions.

To address this, evolutionary processes can be used to synthesize an optimal, context-dependent relationship between a robot's internal state and its learning strategy. As explored by Genci Capi in *"Online Robot Strategy Adaptation by Learning and Evolution"* [Cap10], agents can dynamically alter their behaviour based on their internal energy reserves. In this thesis, this concept is implemented through a dynamic Soft-max action selection mechanism. By actively modulating the exploration temperature (β) based on the robot's real-time energy levels, the agent can seamlessly transition from deterministic exploitation of known resources to stochastic exploration when survival is threatened by environmental changes.

However, adapting to sudden contextual shifts introduces a profound vulnerability in standard deep neural networks, a phenomenon known as *Catastrophic Forgetting*. When a neural network must learn a new task or adapt to a newly introduced environmental rule, the synaptic weights required for previously learned tasks are often drastically overwritten, resulting in the sudden erasure of prior knowledge. In architectures utilizing shared hidden layers, adjusting the latent representations to accommodate a new context inherently corrupts the existing mappings.

Understanding this phenomenon is crucial, as it provides the theoretical justification for the topological choices observed in the advanced scenarios of this study. When faced with discrete contextual reward shifts (such as alternating target colours), the evolutionary optimization process naturally favours flattened, linear architectures. By removing the shared hidden layers and adopting a multi-headed structure, the network minimizes cross-contamination between distinct environmental rules, allowing the agent to rapidly "unlearn" obsolete policies and

adapt to new contingencies without destroying its fundamental motor capabilities.

Chapter 2

Experimental Methodology and Training Process

This chapter outlines the experimental methodology and the training framework established to evaluate the robot's online adaptation capabilities. The primary objective is to investigate how a hybrid strategy, merging online Actor-Critic RL with Evolutionary Computation, enables an agent to navigate and survive despite drastic, non-stationary environmental shifts.

All simulations are conducted within the ARGoS3 environment [PTO+12], a multi-robot simulator chosen for its high-performance physics engine and its ability to handle complex interactions across multiple generations. The methodology is validated through two distinct experimental tracks:

- **Scenario A** focuses on an energy-driven survival task, replicating the foundational work of [Cap10] where the robot must adapt its foraging strategy based on battery density and movement costs.
- **Scenario B** introduces an original contextual task involving ColorBots (Yellow, Red, and Green targets), where the robot must identify the reward-giving colour as contingencies change throughout the simulation.

To optimize these controllers, a GA is employed over 50 generations. This evolutionary process is responsible for sculpting the initial weight configurations, network structures, and the meta parameters that govern the balance between

exploration and exploitation, ensuring the robot is prepared for rapid online adaptation from the moment it is placed in the environment.

The following sections will detail the specific problem definitions, the configuration of the simulation environment, the mechanics of the evolutionary training, and the design of the fitness functions used to guide the agents' development.

2.1 Problem Definitions

This section defines the two experimental scenarios designed to evaluate the online adaptation capabilities of the proposed Actor-Critic architectures. Both problems are framed as non-stationary tasks where the reward structure or environmental dynamics change over time, forcing the agent to shift its behavioural strategy to remain optimal.

2.1.1 Scenario A: Energy-Based Strategy Adaptation

The first scenario is a survival task inspired by the biologically-motivated "Cyber Rodent" experiments [Cap10]. The robot's primary objective is to maintain its energy level by foraging for battery packs in a 2.5m×3.5m arena. The difficulty of the task lies in the non-stationary nature of the environment, which alternates between two distinct configurations, Environment 1 (`ENV1`) and Environment 2 (`ENV2`), for a total of four phases (`ENV1 > ENV2 > ENV1 > ENV2`).

- **Environment 1 (High Resource Density):** This setting features eight active battery packs with long reactivation times. Because the energy cost for movement is low, the optimal strategy is exploratory, requiring the robot to actively search for and capture any visible battery pack.

- **Environment 2 (Low Resource Density):** This setting contains only two battery packs with very short reactivation times. However, the energy cost for movement is significantly increased. In this context, an active search becomes inefficient; the optimal strategy is conservative, where the robot learns to wait near a recently captured battery until it reactivates rather than wasting energy searching for others.

The challenge for the RL agent is to "unlearn" the exploratory behaviour when the environment switches to `ENV2` and vice versa, using the prediction error from the Critic to drive this online strategy shift.

2.1.2 Scenario B: Multi-colour Contextual Task

The second scenario is designed to test the robot's ability to handle discrete contextual reward shifts. In this setup, the arena is populated by three types of "ColorBots", which are stationary robots serving as targets distinguishable by their LED colours: Yellow, Red, and Green. The difficulty of the task lies in the non-stationary nature of the reward schedule, which cycles through three distinct environmental configurations, `TYPE1`, `TYPE2`, and `TYPE3`, for a total of three phases (`TYPE1 > TYPE2 > TYPE3`).

- **Environment TYPE1 (Yellow Reward):** In this setting, only capturing **Yellow ColorBots** provides an energy reward. The optimal strategy requires the robot to identify and selectively pursue yellow targets while ignoring Red and Green ColorBots to avoid energy penalties associated with incorrect captures.

- **Environment TYPE2 (Red Reward):** The reward contingency shifts so that only **Red ColorBots** are active for rewards. To maintain energy efficiency, the robot must suppress its previously learned preference for yellow and adapt its targeting behaviour toward the newly rewarded red stimuli.

- **Environment TYPE3 (Green Reward):** In the final phase, the reward is exclusively provided by **Green ColorBots**. The robot must again modify its strategy to identify the new rewarded target type, ensuring its capture actions remain aligned with the current environmental context.

The challenge for the RL agent is to autonomously identify, through online interaction, which colour is currently rewarded. The agent must successfully modify its pursuit and capture strategy each time the environmental context switches to minimize battery penalties and maximize survival.

2.2 Simulation Environment

This section provides the technical specifications of the simulation framework and the specialized controllers developed for both the learning agent and the interactive target entities.

2.2.1 Robots Specification and Controllers

The implementation utilizes distinct control logics for the learning agent and the environmental targets, tailored to the specific requirements of each experimental scenario.

- **Scenario A Learning Agent**: The agent is a Foot-bot (fig. 2.1) equipped with an internal battery capacity of 21,600 units. Given the simulation frequency of 2 ticks per second, this energy level provides the robot with a maximum lifespan of 3 hours of simulated time in the absence of recharge. The agent is governed by an Actor-Critic architecture that processes environmental inputs, specifically the normalized distance and availability of batteries within its field of view—to select between three discrete actions:

  - `SEARCH`: The robot performs a rotational scan of approximately 10° over 2 simulation ticks (1 second). This action consumes 0.75 units per step, resulting in a total cost of 1.5 battery units.
  - `WAIT`: The robot remains stationary for a maximum duration of 20 steps (10 seconds) to conserve energy. The action terminates early if a battery is detected within the visual range. Each step spent waiting consumes 0.5 units, totalling 1 unit for every second of inactivity.
  - `CAPTURE`: If a target is present, the robot executes a straight-line approach using differential steering. The energy cost is calculated based on the distance travelled: 4.0 units per meter in `ENV1` and 15.0 units per meter in `ENV2`. If this action is selected when no battery is visible, the agent receives a fixed penalty of 2.5 battery units.

Algorithm 1 Actor-Critic Asynchronous Control Loop (Scenario A)

Input: *stepCounter*, *batteryLevel*
hasBattery, *distance*, *angle* ← ReadSensors
if *request_new_action* is **True then** ▷ Phase 1: Action Selection
    action ← SelectActionFromActorCritic(*hasBattery*, *distance*)
    start_step ← *stepCounter*
    request_new_action ← **False**
end if
if not *request_new_action* **and not** *action_ended* **then** ▷ Phase 2: Execution (Spans multiple ticks)
    if *action* is CAPTURE **then**
        MoveTowards(*angle*) **and** ApplyMovementCosts
        action_ended ← CheckCaptureSuccessOrTimeout(*hasBattery*)
    else if *action* is SEARCH **then**
        Rotate **and** ApplySearchCost
        action_ended ← CheckSearchTimeout(*start_step*)
    else if *action* is WAIT **then**
        StopMotors **and** ApplyWaitCost
        action_ended ← CheckTimeoutOrBatterySpotted(*hasBattery*)
    end if
end if
if *action_ended* is **True then** ▷ Phase 3: RL Update
    StopMotors
    duration ← (*stepCounter* − *start_step*)/$TICKS\_PER\_SEC$
    RegisterRewardAndUpdateAgent(*duration*, *batteryLevel*)
    request_new_action ← **True**
    action_ended ← **False**
end if
if *batteryLevel* ≤ 0 **or** *stepCounter* ≥ MAX_STEPS **then**
    TerminateSimulation
end if

- **Scenario B Learning Agent**: The agent utilized in the second scenario is a Foot-bot (fig. 2.1) initialized with a battery capacity of 100 units. Unlike the survival-oriented agent, this controller is equipped with enhanced sensory capabilities, allowing it to perceive both the distance to the nearest target and its specific colour. In this contextual foraging task, the agent operates through a dual-decision stream designed to independently manage behavioural execution and target prioritization:

  - **Action Line**: This stream governs the robot's physical behaviour, choosing between `SEARCH` and `CAPTURE`. In a departure from Scenario A, the `SEARCH` action facilitates a continuous rotation that persists until a target of the prioritized colour is identified within the agent visual cone.
  - **Target Line**: A specialized output head determines the specific target colour the robot will pursue: **Yellow**, **Red**, or **Green**.

  The performance of the agent is regulated by a specific energy-cost system. Every selection of the `SEARCH` action results in a 0.5-unit energy deduction from the battery. For the `CAPTURE` action, the reward is context-dependent: capturing the correct color for the current environmental phase grants a reward of 3 battery units, whereas capturing an incorrect colour results in a penalty of 0.4 units.

Finally, visual perception is simulated through the Range and Bearing (RAB) communication system. To emulate the constraints of a physical camera, the robot is configured to process only those signals received within a 30-degree cone centred on its forward-facing axis ($\pm 15°$). Messages originating from sensors outside this range are systematically ignored, ensuring that target detection and capture actions are based solely on objects within the agent's simulated field of view.

2.2.2 Simulation and Arena

The temporal scale of the simulation is standardized across both scenarios to allow for long-term behavioural analysis. The total experiment duration is set to 43,200 steps, which corresponds to exactly 6 hours of simulated time. This value is

Algorithm 2 Multi-Head Actor-Critic Control Loop (Scenario B)

Input: $stepCounter$, $batteryLevel$, $currentEnvironment$
$hasBot, dist_Y, dist_R, dist_G, angle, detectedColor \leftarrow$ ReadVisionSensors
if $requires\_new\_action$ is **True then** ▷ Phase 1: Multi-Head Action Selection
    $actionType \leftarrow$ SampleActionHead ▷ SEARCH or CAPTURE
    $targetColor \leftarrow$ SampleTargetHead ▷ YELLOW, RED, or GREEN
    $start\_step \leftarrow stepCounter$
    $requires\_new\_action \leftarrow$ **False**
end if
if not $requires\_new\_action$ **and not** $action\_ended$ **then** ▷ Phase 2: Physical Execution
    if $actionType$ is SEARCH **then**
        RotateRobot **and** ApplySearchCost
        $action\_ended \leftarrow$ CheckSearchTimeout($start\_step$)
    else if $actionType$ is CAPTURE **then**
        if not $hasBot$ **or** $detectedColor \neq targetColor$ **then**
            ApplyPenaltyCost ▷ Wrong target or no target
            $action\_ended \leftarrow$ **True**
        else
            MoveTowards($angle$) **and** ApplyMovementCost
            $action\_ended \leftarrow$ CheckCaptureSuccessOrTimeout($hasBot$)
        end if
    end if
end if
if $action\_ended$ is **True then** ▷ Phase 3: RL Update
    StopMotors
    $duration \leftarrow (stepCounter - start\_step)/TICKS\_PER\_SEC$
    $reward \leftarrow$ CalculateReward($currentEnvironment, targetColor$)
    UpdateCritic($reward, duration$)
    UpdateActorHeads($actionType, targetColor$) ▷ Updates both branches
    $requires\_new\_action \leftarrow$ **True**
    $action\_ended \leftarrow$ **False**
end if
if $batteryLevel \leq 0$ **or** $stepCounter \geq MAX\_STEPS$ **then**
    TerminateSimulation
end if

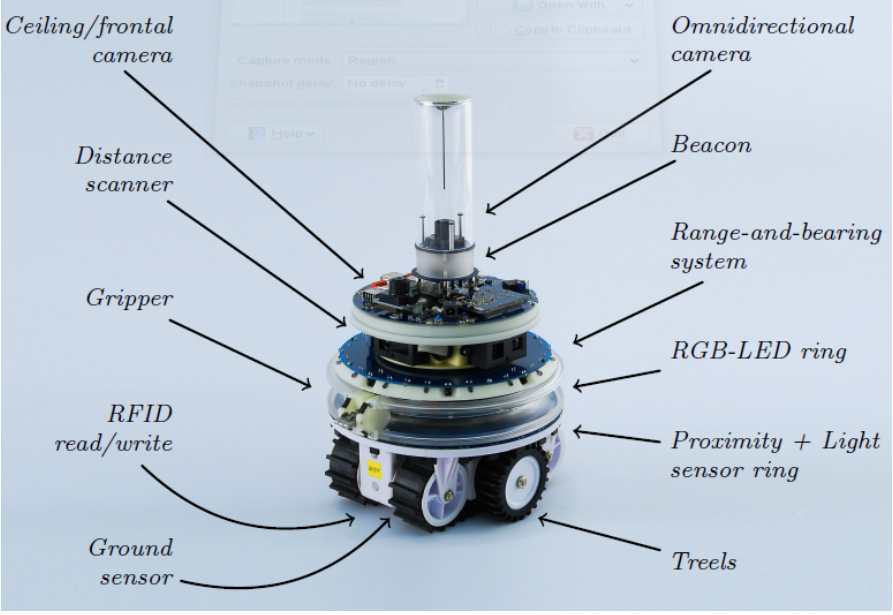

Figure 2.1: The Foot-bot robotic platform: overall structure and schematic of the onboard sensors utilized for online adaptation tasks.

derived from the simulation frequency of 2 ticks per second: 6 hours consist of 360 minutes, totalizing 21,600 seconds, which results in 43,200 ticks. This extended time horizon ensures the agent experiences multiple environmental shifts, providing sufficient data to evaluate its ability to "unlearn" obsolete strategies and re-adapt to new reward contingencies.

The simulation environment is divided into two primary setups corresponding to the experimental tracks:

- **Scenario A Arena:** This setup utilizes a rectangular arena with dimensions of 2.5m×3.5m. The distribution of the learning agent and the ten battery packs follows a specific spatial layout designed to test foraging efficiency under varying movement costs (fig. 2.2).

  The resource dynamics in this arena are highly dependent on the current environmental phase, which swaps every 10,800 ticks:

  - **Environment 1 (`ENV1`):** When a battery is captured, it becomes inactive and remains disabled for 100 seconds (200 simulation ticks) before recharging.

– **Environment 2 (`ENV2`):** To compensate for the lower density of resources, captured batteries in this phase have a significantly shorter cool-down of 10 seconds (20 simulation ticks).

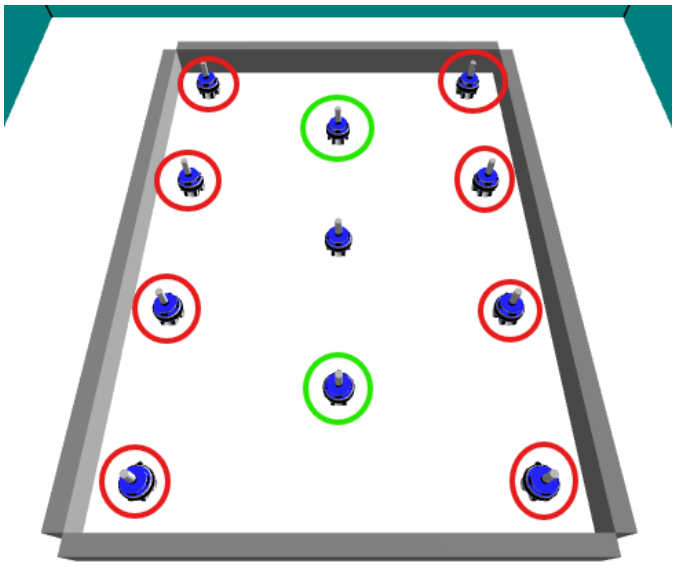

Figure 2.2: Schematic of the arena for Scenario A, illustrating the spatial distribution of battery targets. Red markers represent the eight battery packs active during the high-density phase (`ENV1`), while green markers indicate the two packs available during the low-density phase (`ENV2`).

- **Scenario B Arena:** For the multi-color contextual foraging task, the arena is configured as a 4.0m×4.0m square area (fig. 2.3). In this setup, the learning agent and the three types of ColorBots are distributed alternately throughout the space, requiring the robot to navigate between different colour stimuli to identify the currently rewarded target.

  In this scenario, the environmental context swaps every 14,400 ticks. The resource dynamics are uniform across all targets:

  – All ColorBots, regardless of their color (Yellow, Red, or Green) or the current environment type (`TYPE1`, `TYPE2`, or `TYPE3`), feature a fixed reactivation delay.

– Once captured, a ColorBot remains inactive for 50 seconds (100 simulation ticks) before re-spawning and becoming available for a new capture attempt.

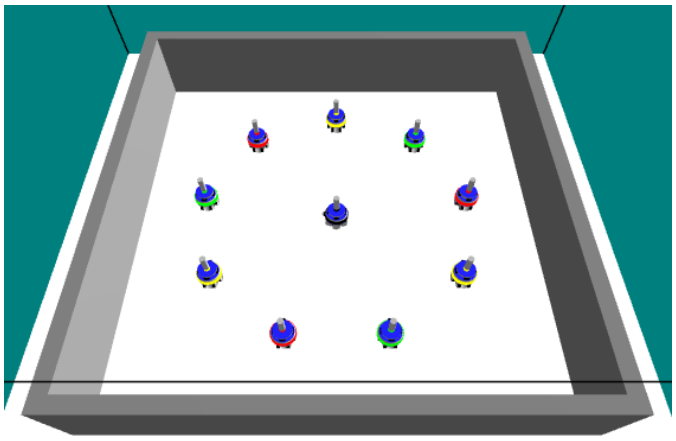

Figure 2.3: Schematic of the Scenario B arena highlighting the multi-color target distribution.

2.3 Evolutionary Training Process

The optimization of the robotic controllers is performed through an evolutionary process that integrates Reinforcement Learning (RL) with a **Genetic Algorithm (GA)**. This hybrid approach, categorized as Evolutionary Reinforcement Learning, is utilized to discover the optimal set of meta parameters, initial weights, and neural architectures that enable effective online adaptation. This GA framework is applied identically to both experimental scenarios to maintain a consistent training methodology.

2.3.1 Population Structure and Distribution

The evolutionary process is conducted over **50 generations**, with each generation consisting of a population of **100 individuals**. To prevent premature convergence and maintain a balance between the exploitation of successful traits and the ex-

ploration of the parameter space, the population is divided into three distinct groups:

- **Elite (10%)**: The top 10 individuals from the previous generation are preserved as exact copies to ensure that the highest-performing solutions are never lost.

- **Mutated (70%)**: New individuals are generated by selecting parents from a "Parent Pool" consisting of the top 40% of the population. These offspring undergo **uniform crossover** and subsequent mutation .

- **Random Immigrants (20%)**: To prevent genetic stagnation, 20 individuals in each generation are created with entirely random parameters, injecting fresh diversity into the gene pool.

2.3.2 Optimization Space

The GA explores a multi-dimensional space to optimize the following parameters for the Actor-Critic controllers:

- **Learning Rates**: Alpha values for the input-to-hidden and hidden-to-output layers for both the Actor and the Critic (α_1, α_2 for the Actor; p_1, p_2 for the Critic).

- **Architectural Complexity**: The number of hidden neurons for both modules is evolved, ranging from 0 (linear model) to 5 neurons.

- **Behavioural Meta parameters**: The energy thresholds (e_1, e_2) that trigger shifts in the exploration-exploitation balance, and the corresponding softmax temperatures $(\beta_1, \beta_2, \beta_3)$ used to modulate action selection.

- **Initial Neural Weights**: Starting weights for all layers are evolved within a range of $(-0.5, 0.5)$ to provide the agent with a beneficial "innate" starting point for its lifetime learning.

2.3.3 Fitness Function

To guide the GA toward optimal adaptive behaviours, a specialized fitness function is designed for each scenario. These functions are responsible for evaluating the performance of each individual at the end of its 6-hour simulated lifespan (43,200 ticks), providing a scalar value that determines its reproductive success in the next generation.

The fitness evaluation is bifurcated based on the agent's survival. If the robot's battery reaches zero before the end of the simulation, it is considered a "failure", and its fitness is calculated based on its longevity. If the robot survives the entire duration, it is considered a "success", and its fitness is rewarded based on efficiency and task performance.

Fitness A: Survival and Energy Efficiency In the energy-management task, the primary objective is survival through efficient foraging. The fitness function for Scenario A prioritizes longevity and the maintenance of high energy reserves:

1. **Successful Survival** (Ticks = 43,200):

$$Fit_A = 1.0 + \frac{\text{Average Battery Level}}{\text{Maximum Battery Capacity}} \tag{2.1}$$

   This results in a score between **1.0 and 2.0**, where higher values represent agents that managed to forage effectively while minimizing movement costs.

2. **Premature Failure** (Ticks < 43,200):

$$Fit_A = (0.5 \times T_{norm}) + (0.49 \times E_{avg\_norm}) \tag{2.2}$$

   Where T_{norm} is the normalized survival time and E_{avg_norm} is the average energy level during its lifetime. This ensures that even failing agents are ranked by their ability to delay battery depletion.

Fitness B: Contextual Accuracy and Energy In the multi-colour contextual task, the fitness function is more complex as it must account for the **correctness of captures** in addition to survival and energy. The "Score" represents the difference

between correct captures (e.g., Yellow in TYPE1) and incorrect ones (e.g., Red or Green in TYPE1).

1. **Successful Survival** (Ticks = 43,200):

$$Fit_B = 3.0 + E_{avg_norm} + (S_{total_norm} \text{ if } S_{total_norm} \geq 0 \text{ else } -0.1) \tag{2.3}$$

   Where E_{avg_norm} is the average energy across the three environmental phases and S_{total_norm} is the normalized total score. This formula, yielding a value between **2.9 and 5.0**, strongly rewards agents that not only survive but also correctly identify the rewarded colour for each phase.

2. **Premature Failure** (Ticks < 43,200):

$$\begin{aligned} Fit_B = (0.5 \times T_{norm}) + (0.5 \times E_{avg\_norm}) \\ + (S_{total\_norm} + 1.0 \text{ if } S_{total\_norm} \geq 0 \text{ else } 0) \end{aligned} \tag{2.4}$$

   This multi-component approach ensures that even if a robot dies, its evolutionary rank is determined by how much energy it gathered and how accurately it targeted the correct colors before failure.

Comparison of Objectives While both scenarios utilize energy as a baseline metric, Scenario A focuses on **behavioural strategy adaptation** (Exploratory vs. Conservative) , whereas Scenario B focuses on **contextual discrimination** (Target Selection). By evolving the meta parameters (learning rates α,p and softmax temperatures β) under these fitness pressures, the GA identifies configurations that allow the Actor-Critic networks to "unlearn" old strategies and adapt to new ones with minimal delay.

2.3.4 Mutation and Reproduction Logic

Reproduction utilizes **Uniform Crossover**, where each parameter has a 50% probability of being inherited from either parent. Following crossover, continuous parameters undergo **Additive Gaussian Mutation**. The mutated value x' is derived from the original parameter x using the following formula:

$$x' = \text{clamp}(x + \epsilon, x_{min}, x_{max}), \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{2.5}$$

where the standard deviation σ of the Gaussian noise is scaled according to the specific range of the parameter:

$$\sigma = s \cdot (x_{max} - x_{min}) \tag{2.6}$$

In these equations, s represents the mutation scale, while x_{min} and x_{max} define the boundaries of the search space for that parameter. The `clamp` function ensures that the resulting value remains within the physically or mathematically valid bounds defined for the controller.

The scale of the mutation is tailored to the parameter type to balance fine-tuning with global exploration. For instance, the discount factor `critic_gamma` utilizes a conservative 5% scale, while learning rates α and soft-max temperatures β employ larger scales 20% to prevent the population from becoming trapped in local optima.

Chapter 3

Proposed Actor-Critic Architectures

This chapter details the design and implementation of the neural architectures developed to address the non-stationary challenges described in previous sections. Both architectures are based on an Actor-Critic Reinforcement Learning framework, optimized via the Genetic Algorithm to enable rapid and efficient online adaptation.

3.1 Model A: Baseline Actor-Critic

The neural architecture for Scenario A is based on a Reinforcement Learning Actor-Critic model, significantly inspired by the "Cyber Rodent" experiments conducted by Capi et al. [Cap10]. The system is designed to handle survival tasks in non-stationary environments where the agent must adapt its decision-making policy based on internal energy states and external resource availability.

State Representation (s) The model observes a state vector s composed of 3 normalized values ($N\_STATE = 3$):

- **Bias (s_0):** A constant term set to 1.0 to allow for the translation of activation functions.

- **Battery Charge (s_1):** The robot's current energy level, calculated as the ratio between the current battery and its maximum capacity: $\frac{m\_batteryLevel}{MAX\_BATTERY}$.

- **Battery Distance (s_2):** The normalized distance to the nearest target detected within the 30° visual cone. If no battery is detected, this value defaults to 1.0, representing the maximum sensor range.

Neural Network Architecture The model is flexible and can be configured in two variations for both the Actor and the Critic, depending on the parameters optimized by the Genetic Algorithm:

- **Linear Model (No Hidden Layer):** Input values are mapped directly to output preferences through a linear combination of weights w.

- **Hidden Layer Model:** Includes a hidden layer consisting of 1 to 5 neurons. The activation function used for the hidden layer is the hyperbolic tangent (tanh).

The Actor (fig. 3.1) output consists of logits (action preferences), while the Critic (fig. 3.2) produces a single scalar value $V(s)$ representing the value estimate of the current state.

3.1.1 Learning Dynamics and Mathematical Foundations

Temporal Difference Error (TD-Error) The Critic evaluates the current state and calculates the prediction error δ (TD-error) using the discount factor γ and the received reward r:

$$\delta = r + \gamma^{\Delta t} \cdot V(s_{next}) - V(s)$$

In this equation, Δt represents the duration of the action in seconds. This allows for a consistent update even when actions, such as `CAPTURE` or `WAIT`, have variable execution times.

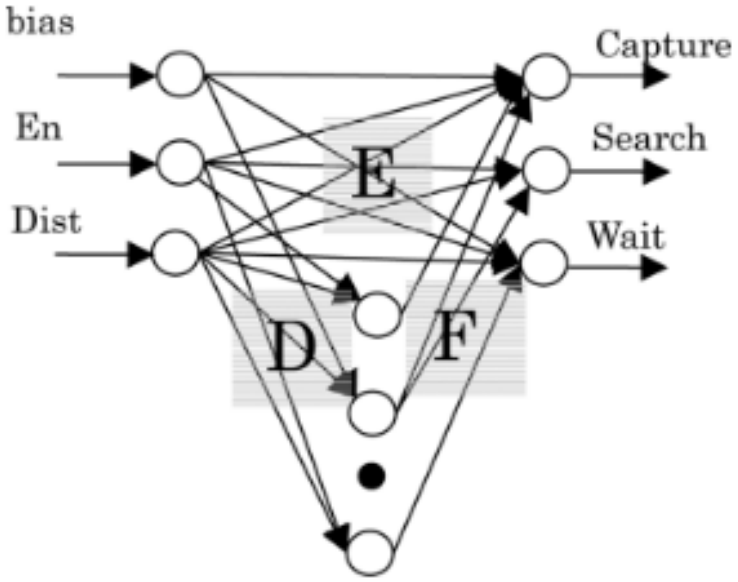

Figure 3.1: Architecture of the actor neural network (adapted from [CD06]).

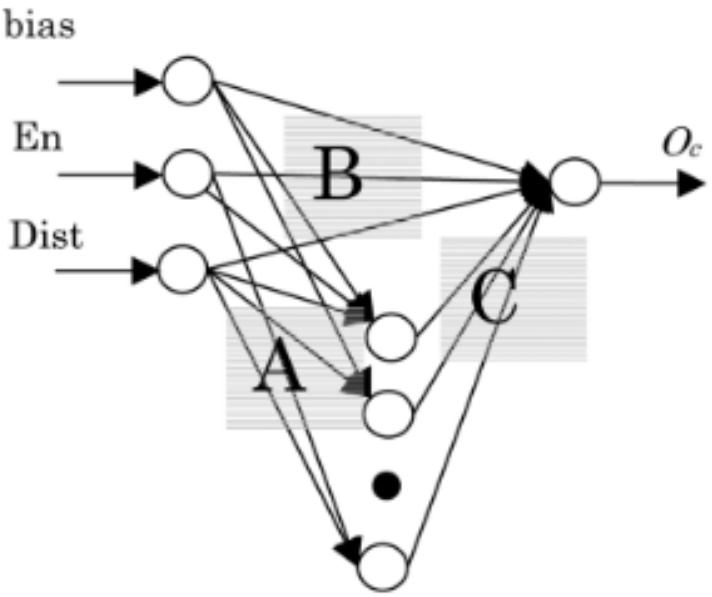

Figure 3.2: Architecture of the critic neural network (adapted from [CD06]).

Action Selection: Soft-max and Beta (β) The preferences calculated by the Actor (P_a) are converted into action probabilities using a Soft-max function modulated by the inverse temperature parameter β:

$$P(a|s) = \frac{e^{\beta \cdot P_a}}{\sum_j e^{\beta \cdot P_j}} \tag{3.1}$$

The β parameter is not static; it is dynamically modulated based on the normalized energy level ($energy_{norm}$) to balance exploration and exploitation:

- If $energy_{norm} \leq en_1$: β linearly interpolates between β_1 and β_2.
- If $en_1 < energy_{norm} \leq en_2$: β interpolates between β_2 and β_3.
- If $energy_{norm} > en_2$: β is fixed at β_3.

Weight Updates Weights are updated online using the δ signal. For the Actor's linear model, the update rule utilizes the advantage of the chosen action:

$$w_{a,i} \leftarrow w_{a,i} + \alpha \cdot \delta \cdot (\mathbb{I}(a = action) - P(a|s)) \cdot s_i$$

Where $\mathbb{I}$ is the indicator function for the action actually chosen and α is the learning rate. The Critic updates its weights to minimize the value estimation error:

$$w_i \leftarrow w_i + \alpha \cdot \delta \cdot s_i$$

In architectures with hidden layers, the error is back-propagated through the tanh derivative $(1 - h^2)$ to update the weights between the input and the hidden layer.

3.2 Model B: Multi-headed Actor-Critic Architecture

To address the complexity of contextual target selection in Scenario B, the baseline model is extended into a multi-headed Actor-Critic architecture. This design

enables the agent to decouple fundamental motor behaviours from contextual target prioritization, allowing for more efficient "unlearning" and re-adaptation as reward contingencies shift.

State Representation (s) In this scenario, the state vector s is expanded to 5 normalized values ($N\_STATE = 5$) to provide the network with sufficient information to discriminate between different colour stimuli:

- **Bias (s_0):** Constant term set to 1.0.
- **Average Reward (s_1):** A running average of the rewards obtained, representing a performance-based metric of the current policy.
- **Distance to Yellow (s_2):** Normalized distance to the nearest Yellow ColorBot in the visual cone.
- **Distance to Red (s_3):** Normalized distance to the nearest Red ColorBot in the visual cone.
- **Distance to Green (s_4):** Normalized distance to the nearest Green ColorBot in the visual cone.

Distances are normalized such that 1.0 represents immediate proximity and 0.0 indicates that the specific colour is not detected within the 30° frontal cone.

Multi-headed Neural Structure The Actor network employs a dual-branch (multi-head) structure fig. 3.3. Like Scenario A, it can be implemented as a linear model or with a hidden layer of 1 to 5 neurons using tanh activation.

- **Behavioural Head:** Produces logits for the primary movement actions: `SEARCH` (0) or `CAPTURE` (1).
- **Contextual Head:** Produces logits to select the target priority: `YELLOW` (0), `RED` (1), or `GREEN` (2).

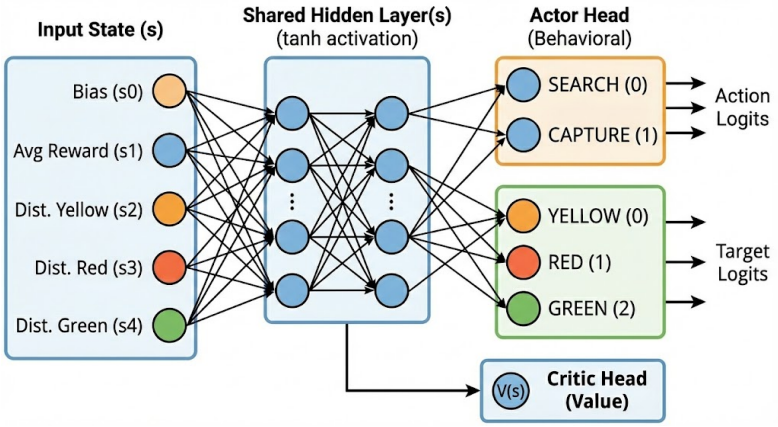

Figure 3.3: Architecture of the actor multi-head neural network.

Performance-Based Beta Modulation Unlike Scenario A, where β is modulated directly by the battery level, Scenario B utilizes a **Performance Metric** derived from the average reward ($m\_avgReward$) to control the exploration-exploitation trade-off.

The average reward is clamped to the range $[-0.5, 2.5]$ and normalized to $[0, 1]$. This value is then mapped to a performance metric $P \in [0.01, 10]$:

$$P = 0.01 + \left(\frac{\text{clamped\_reward} + 0.5}{3.0} \right) \cdot 9.99 \tag{3.2}$$

The inverse temperature β is then computed through piecewise linear interpolation using P as input:

- If $P \leq en_1$: β interpolates between β_1 and β_2.
- If $en_1 < P \leq en_2$: β interpolates between β_2 and β_3.
- If $P > en_2$: $\beta = \beta_3$.

This mechanism ensures that when the robot experiences a drop in performance (e.g., due to an environmental shift), β adjusts to encourage the exploration of new (Action, Target) combinations.

Conditional Learning Update The multi-headed structure introduces a conditional weight update logic to ensure that learning is targeted and does not de-

grade irrelevant skills.

- **Behavioural Branch:** The weights for the action head (w_{act}) are updated at every step based on the TD-error δ, regardless of the action chosen.
- **Contextual Branch:** The weights for the target head (w_{tar}) are updated **only if** the selected action was `CAPTURE`:

$$\text{If action} = \texttt{CAPTURE} \implies w_{tar,i} \leftarrow w_{tar,i} + \alpha \cdot \delta \cdot (\mathbb{I}(t = target) - P(t|s)) \cdot s_i \tag{3.3}$$

In the hidden layer model, the total gradient is the sum of the gradients from both heads. If the agent is in `SEARCH` mode, the gradient contribution from the contextual head is set to zero, focusing the hidden layer's adaptation solely on the search behaviour.

Chapter 4

Results

This chapter constitutes the empirical heart of the research, where the theoretical frameworks and neural architectures designed in previous sections are subjected to rigorous experimental validation. Moving from the conceptual design of the Actor-Critic models to their practical application in simulated environments, we provide a comprehensive analysis of how the evolutionary process shapes the robot's ability to navigate complexity and non-stationarity.

The results presented herein are the culmination of an extensive computational campaign conducted over two distinct experimental tracks: Scenario A, focused on energy-management strategies, and Scenario B, focused on multi-colour contextual discrimination. By utilizing a GA to optimize the meta parameters and initial configurations of the neural networks, we aim to observe the transition from initial random exploration to specialized adaptive behaviours.

Performance is evaluated through a multi-faceted approach that considers three primary dimensions of the agent's "life":

- **Evolutionary Progress:** The capacity of the GA to increase the overall fitness of the population over 50 generations, identifying clusters of parameters that promote survival in shifting environments.

- **Behavioural Efficiency:** The robot's ability to manage its internal energy reserves and optimize movement costs in response to varying resource densities.

- **Cognitive Flexibility:** The success of the multi-headed architecture in "unlearning" obsolete target preferences and rapidly adapting to new reward contingencies without losing fundamental motor skills.

The structure of this chapter follows the logical progression of the experiment. We begin by examining the macro-evolutionary trends, documenting the convergence of the population's fitness and the stabilization of evolved parameters such as learning rates and soft-max temperatures. Subsequently, the analysis shifts to a micro-behavioural perspective, offering a detailed look at individual simulation runs. Through these high-resolution observations, we demonstrate the role of the Critic's error signal as a catalyst for strategy switching, effectively bridging the gap between internal neural dynamics and external behavioural adaptation.

Ultimately, the data presented in this chapter seeks to confirm that the integration of Evolutionary Computation and Reinforcement Learning provides a robust pathway for developing autonomous agents capable of thriving in environments where the only constant is change.

Statistical Robustness To ensure the statistical significance and reliability of the results, the training phase was evaluated across 10 independent evolutionary trials. This multi-run approach accounts for the stochastic nature of the GA and mitigates the risk of capturing outliers. Each trial encompasses 50 generations with a population of 100 individuals, resulting in 5,000 simulated lifetimes per experiment. Cumulatively, the study relies on a robust dataset of 50,000 distinct evaluations, providing a stable foundation for the analysis of convergence and parameter evolution.

4.1 Evolutionary Dynamics

The primary objective of this experimental phase is to validate the effectiveness of the Evolutionary Reinforcement Learning framework in synthesizing adaptive robotic controllers. This section delves into the evolutionary dynamics observed across the 50 generations of the Genetic Algorithm (GA). By tracking the historical progression of the population, we can ascertain whether the algorithmic

design successfully guided the search process toward robust, non-stationary solutions rather than trapping the agents in brittle local optima. This section will unpack these dynamics by analysing the two most critical indicators of evolutionary progress: the convergence of the fitness function and the generational survival rate. Together, these metrics illuminate the transition from early populations characterized by random, inefficient exploration to mature populations capable of sophisticated strategy switching and contextual discrimination.

4.1.1 Fitness and Success Rate Evolution

Scenario A

As defined in the methodology, the fitness function yields a score between 0.0 and 0.99 for premature failures (battery depletion before 43,200 ticks), and a score between 1.0 and 2.0 for agents that successfully survive the full 6-hour duration. For survivors, the fractional component directly reflects the average energy reserves maintained.

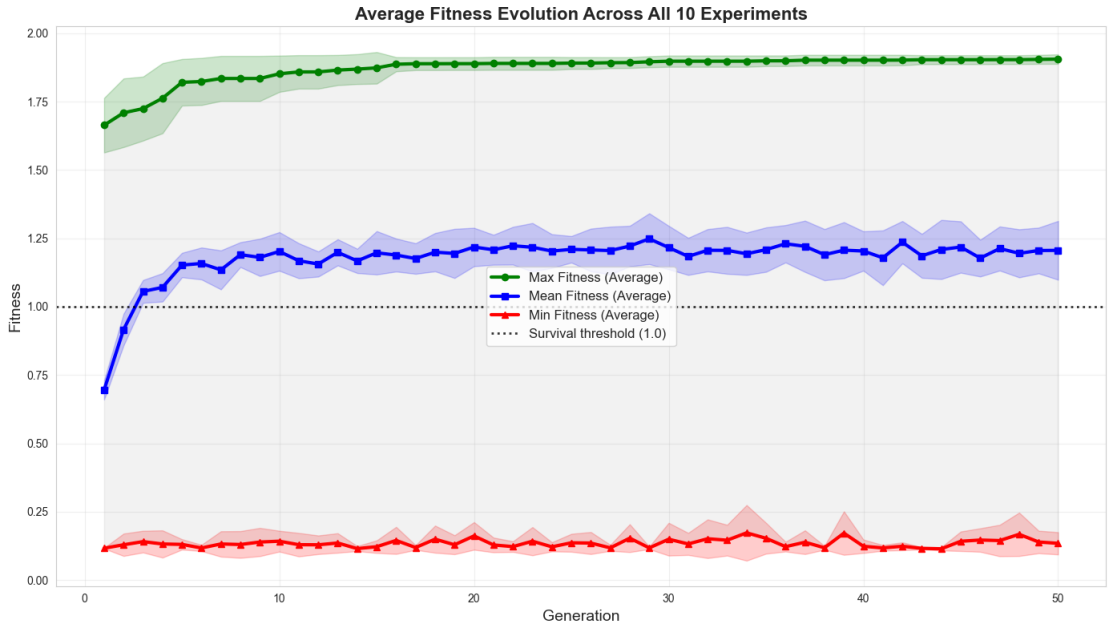

Figure 4.1: Scenario A fitness evolution across all 10 experiments.

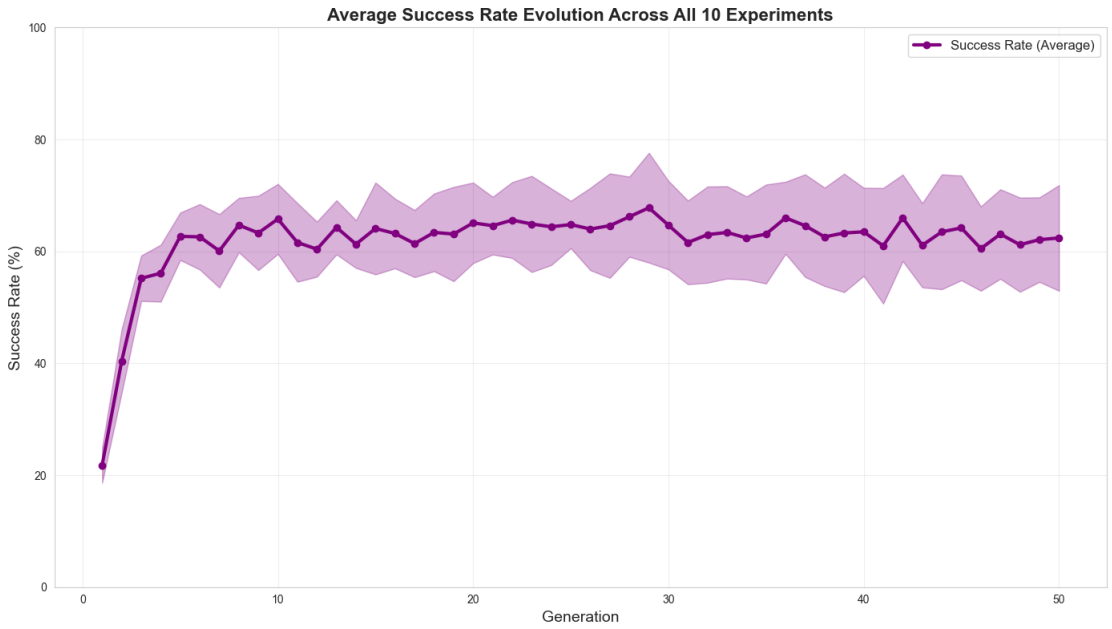

Figure 4.2: Scenario A success evolution across all 10 experiments.

To quantify the magnitude of the evolutionary progress, table 4.1 presents a direct comparison of the overall population metrics between the initial instantiation (Generation 1) and the final optimized state (Generation 50), averaged across all 10 independent experiments.

Generation	Max Fitness	Mean Fitness	Min Fitness	Success Rate (%)
Gen 1 (Initial)	1.6637	0.6956	0.1167	21.8%
Gen 50 (Final)	1.9053	1.2059	0.1345	62.4%
Absolute Growth	**+ 0.2416**	**+ 0.5103**	**+ 0.0178**	**+ 40.6%**

Table 4.1: Overall evolutionary progress: Comparison of average fitness metrics and success rates between Generation 1 and Generation 50 across 10 independent GA runs for Scenario A.

As illustrated in the data and the accompanying convergence plots (fig. 4.1, fig. 4.2), the evolutionary trajectory of the population can be distinctly categorized into three main phases:

- **Initial Exploration (Generations 1 to 3):** At the onset of the GA, the

	Max Fitness		Mean Fitness		Min Fitness		Success Rate (%)	
Experiment	**Gen 1**	**Gen 50**	**Gen 1**	**Gen 50**	**Gen 1**	**Gen 50**	**Gen 1**	**Gen 50**
actorcritic_RL_1	1.5972	1.9086	0.7195	1.1245	0.1163	0.1286	23.0	58.0
actorcritic_RL_2	1.6810	1.9116	0.7503	1.2611	0.1159	0.1054	24.0	69.0
actorcritic_RL_3	1.8361	1.9124	0.6747	1.3021	0.1168	0.2510	23.0	72.0
actorcritic_RL_4	1.6141	1.9278	0.7232	1.1548	0.1172	0.1134	24.0	57.0
actorcritic_RL_5	1.5785	1.8773	0.6595	1.0795	0.1159	0.1459	20.0	49.0
actorcritic_RL_6	1.5884	1.9245	0.7193	1.1165	0.1176	0.1160	26.0	57.0
actorcritic_RL_7	1.5693	1.9121	0.7128	1.0675	0.1203	0.1160	21.0	49.0
actorcritic_RL_8	1.7950	1.9139	0.6561	1.2520	0.1156	0.1403	18.0	64.0
actorcritic_RL_9	1.5823	1.8766	0.6332	1.2910	0.1159	0.1110	15.0	71.0
actorcritic_RL_10	1.7948	1.8879	0.7073	1.4099	0.1156	0.1175	24.0	78.0
Overall Averages	**1.6637**	**1.9053**	**0.6956**	**1.2059**	**0.1167**	**0.1345**	**21.8**	**62.4**

Table 4.2: Detailed comparison of fitness metrics and success rates between Generation 1 and Generation 50 for each independent GA run (Scenario A).

population exhibits notably poor performance. As shown in table 4.1, the average Mean Fitness at Generation 1 is only 0.6956, sitting well below the 1.0 threshold required for survival. The average Success Rate is a mere 21.8%, meaning nearly 80% of the randomly instantiated agents fail to manage their energy efficiently, leading to premature battery depletion. The *Min Fitness* recorded at 0.1167 serves as a baseline for these rapidly failing configurations.

- **Rapid Adaptation (Generations 4 to 16):** Following the initial phase, the evolutionary learning curve experiences a steep acceleration. The selection mechanisms effectively identify the most promising configurations from the rare initial survivors (the 21.8% from Gen 1), allowing the GA to quickly discard failing strategies. During this phase, the Mean Fitness successfully crosses the 1.0 survival threshold, indicating that the majority of the population has adopted a viable strategy to balance exploratory foraging with energy conservation across environmental shifts. Concurrently, the deviation of the Max Fitness starts to stabilize, suggesting a transition from broad exploration to focused fine-tuning of the Actor-Critic meta parameters.

- **Stabilization and Mature Convergence (Generations 17 to 50):** In the final phase, the GA enters a stable plateau, settling upon robust and near-optimal configurations. The progress is striking: by Generation 50, the

average Success Rate nearly triples, reaching 62.4%, with top-performing runs (such as `actorcritic_RL_10`) achieving up to a 78.0% survival rate (table 4.2). The *Max Fitness* converges to an impressive 1.9053, demonstrating that elite agents learn to survive while maintaining average battery levels around 90%. The *Mean Fitness* stabilizes at 1.2059, confirming the overall robustness of the population.

It is theoretically significant to note the behaviour of the *Min Fitness*, which barely increases from 0.1167 (Gen 1) to 0.1345 (Gen 50). While the elite and average individuals undergo massive improvements, the minimum fitness exhibits sporadic oscillations and remains anchored near 0.1. Far from being a flaw, this variance is a direct and intended consequence of the continuous injection of random immigrants (20% of the population) at each generation. This mechanism successfully preserves necessary genetic diversity, ensuring that the elite solutions are constantly challenged by novel configurations and preventing the overall population from stagnating in brittle local optima.

Scenario B

Unlike Scenario A, the fitness landscape in Scenario B is significantly more complex. The fitness function must account not only for survival time and energy efficiency but also for the agent's contextual accuracy—its ability to correctly identify and capture the specific target colour (Yellow, Red, or Green) assigned to the current environmental phase, while actively avoiding the others. Consequently, the maximum achievable fitness score in this scenario approaches 5.0.

To quantitatively assess the GA's ability to navigate this complex, multi-objective search space, table 4.3 compares the overall population metrics between Generation 1 and Generation 50.

As illustrated in the data and the accompanying convergence plots (fig. 4.3, fig. 4.4), the evolutionary trajectory for Scenario B reveals the stark difficulty of the contextual adaptation task, marked by three distinct phases of learning:

- **Initial Exploration (Generations 1 to 10):** The onset of the evolutionary process is characterized by near-total failure. As detailed in table 4.3,

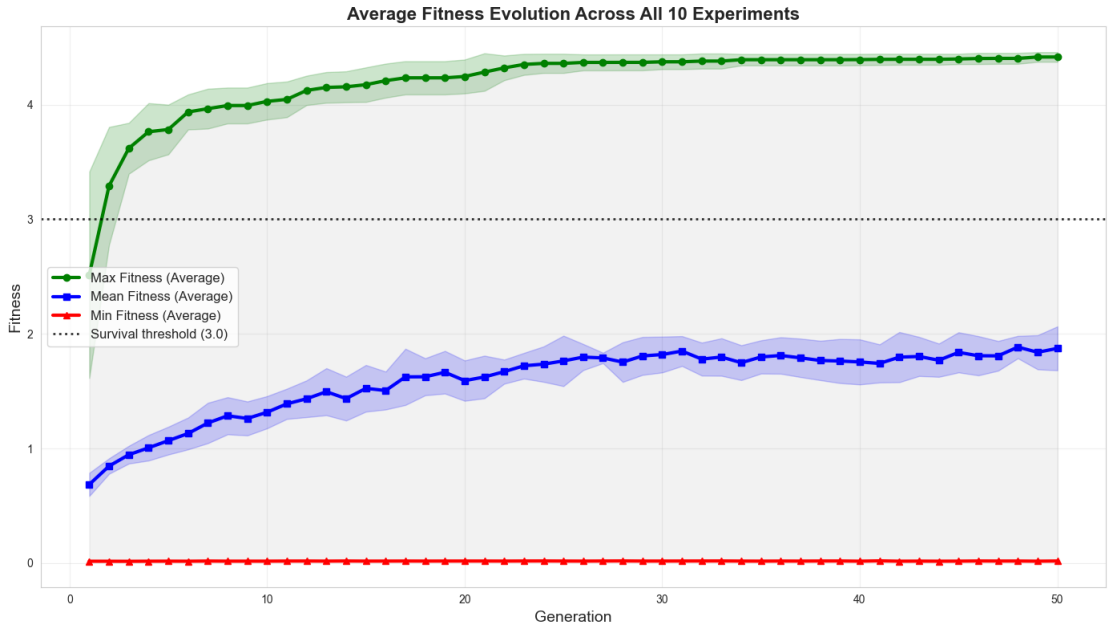

Figure 4.3: Scenario B fitness evolution across all 10 experiments.

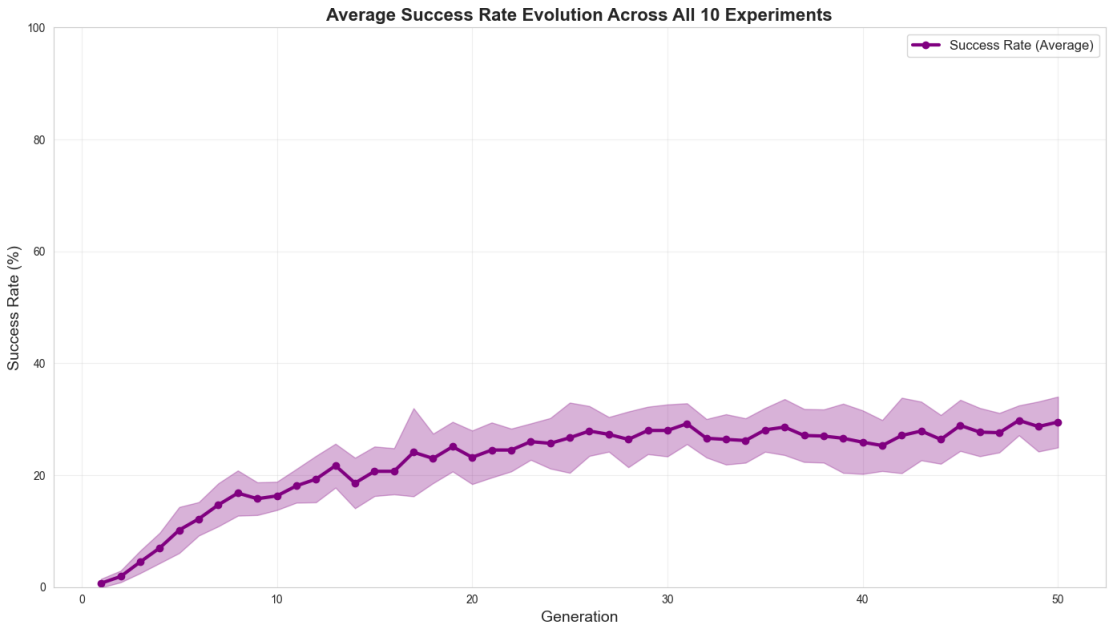

Figure 4.4: Scenario B success evolution across all 10 experiments.

Generation	Max Fitness	Mean Fitness	Min Fitness	Success Rate (%)
Gen 1 (Initial)	2.5150	0.6864	0.0167	0.7%
Gen 50 (Final)	4.4154	1.8735	0.0186	29.5%
Absolute Growth	**+ 1.9004**	**+ 1.1871**	**+ 0.0019**	**+ 28.8%**

Table 4.3: Overall evolutionary progress: Comparison of average fitness metrics and success rates between Generation 1 and Generation 50 across 10 independent GA runs for Scenario B.

	Max Fitness		Mean Fitness		Min Fitness		Success Rate (%)	
Experiment	**Gen 1**	**Gen 50**	**Gen 1**	**Gen 50**	**Gen 1**	**Gen 50**	**Gen 1**	**Gen 50**
actorcritic_RL_1	1.5274	4.4624	0.5151	1.7557	0.0160	0.0160	0.0	26.0
actorcritic_RL_2	3.6463	4.4335	0.8664	1.9550	0.0179	0.0181	2.0	30.0
actorcritic_RL_3	3.3713	4.4506	0.6194	1.6339	0.0161	0.0162	1.0	25.0
actorcritic_RL_4	1.6258	4.3899	0.7440	1.6569	0.0159	0.0185	0.0	27.0
actorcritic_RL_5	1.7822	4.3726	0.6052	2.1977	0.0158	0.0186	0.0	38.0
actorcritic_RL_6	1.5750	4.4951	0.7663	2.0749	0.0177	0.0198	0.0	29.0
actorcritic_RL_7	3.1668	4.3713	0.6033	1.9471	0.0161	0.0202	1.0	32.0
actorcritic_RL_8	3.5846	4.4271	0.7969	1.8600	0.0174	0.0192	1.0	31.0
actorcritic_RL_9	1.6215	4.3842	0.6770	2.0344	0.0182	0.0208	0.0	35.0
actorcritic_RL_10	3.2493	4.3678	0.6705	1.6191	0.0154	0.0189	2.0	22.0
Overall Averages	**2.5150**	**4.4154**	**0.6864**	**1.8735**	**0.0167**	**0.0186**	**0.7**	**29.5**

Table 4.4: Detailed comparison of fitness metrics and success rates between Generation 1 and Generation 50 for each independent GA run (Scenario B).

the *Success Rate* at Generation 1 is a mere 0.7%, meaning less than 1 in 100 agents manages to survive the 6-hour simulation. The average *Mean Fitness* sits at 0.6864, with the *Min Fitness* drastically low at 0.0167. This catastrophic early performance highlights the challenge of the task: agents with randomly initialized contextual heads frequently target the wrong colours, incurring heavy energy penalties that lead to immediate and rapid battery depletion.

- **Rapid Cognitive Adaptation (Generations 10 to 30):** Despite the harsh initial conditions, the GA drives a massive evolutionary leap. The *Max Fitness* experiences a sharp upward trajectory as the elite agents begin to effectively utilize the Critic's TD-error to update their contextual preferences. During this phase, the multi-headed architecture proves its worth: agents learn to decouple their motor skills (the "capture" action) from their target selection, allowing them to rapidly "unlearn" a colour preference when the environment shifts, minimizing penalties and gradually increasing their lifespan.

- **Stabilization and Mature Convergence (Generations 30 to 50):** By the final generations, the algorithm reaches a stable plateau. The *Final Max Fitness* converges to an outstanding 4.4154, representing highly sophisticated agents capable of surviving the environmental swaps with near-perfect target discrimination. The *Final Mean Fitness* climbs by +1.1871 to reach 1.8735, proving the general capability of the population.

It is crucial to contextualize the final *Success Rate* of 29.5%. While this figure may appear lower than the 62.4% observed in Scenario A, it actually represents a monumental relative increase of over $4,100\%$ from the initial 0.7%. The lower absolute survival rate reflects the inherent brutality of Scenario B's penalty system: a single failure to quickly switch contextual targets upon an environmental change can cascade into unrecoverable energy loss. Thus, a nearly 30% success rate across the entire population indicates that the GA successfully engineered a robust strategy-switching logic capable of mitigating profound non-stationary disruptions.

4.1.2 Analysis of Parameters Evolution

The convergence of fitness and survival rates discussed in the previous section is fundamentally driven by the GA's ability to navigate a high-dimensional continuous and discrete parameter space. Before analysing the specific configurations that the evolutionary process selected as optimal, it is essential to define the boundaries of this search space.

The GA was tasked with optimizing a comprehensive set of hyper parameters for the Actor-Critic controllers, which govern both the neural network learning dynamics and the agent's behavioural logic. table 4.5 summarizes the specific parameters, their symbols, and the continuous ranges or discrete options available during the initialization and mutation phases for both Scenario A and Scenario B.

Category	Parameter Description	Symbol	Search Range / Options
Learning Rates	Actor learning rate (Hidden Layer)	α_1	$[0.0, 1.0]$
	Actor learning rate (Output Heads)	α_2	$[0.0, 1.0]$
	Critic learning rate (Hidden Layer)	p_1	$[0.0, 1.0]$
	Critic learning rate (Output Head)	p_2	$[0.0, 1.0]$
Temporal Discount	Critic discount factor	γ	$[0.0, 1.0]$
Energy Thresholds	Low energy boundary	en_1	$[0.1, 0.6]$
	High energy boundary	en_2	$[0.6, 0.9]$
Soft-max Temperatures	Temperature anchor at minimum energy ($energy_{norm} = 0$)	β_1	$[1.0, 10.0]$
	Temperature anchor at first threshold ($energy_{norm} = en_1$)	β_2	$[1.0, 10.0]$
	Temperature anchor for safe energy ($energy_{norm} \geq en_2$)	β_3	$[1.0, 15.0]$
Neural Architecture	Actor hidden layer neurons	N_{actor}	$\{0, 1, 2, 3, 4, 5\}$
	Critic hidden layer neurons	N_{critic}	$\{0, 1, 2, 3, 4, 5\}$
	Initial synaptic weights	w_{init}	$[-0.5, 0.5]$

Table 4.5: Parameter search space for the Genetic Algorithm optimization, applicable to both Scenario A and Scenario B.

The search space includes critical structural choices, such as whether to utilize a purely linear mapping (0 hidden neurons) or a more complex non-linear representation (up to 5 hidden neurons with tanh activation) for both the Actor and the Critic. Furthermore, the temporal discount factor (γ) and the learning rates dictate how quickly the networks adapt to the Temporal Difference (TD) error, while the energy thresholds (en_1, en_2) and temperatures ($\beta_1, \beta_2, \beta_3$) strictly regulate the agent's exploration-exploitation trade-off in response to its internal state or current performance metric.

In the following subsections, we will dissect how these parameters evolved across the 50 generations, identifying which specific configurations were selected

by the elite individuals to solve the survival and contextual discrimination tasks.

Scenario A

The data extracted from the evolutionary logs reveals how the Genetic Algorithm (GA) sculpted the meta parameters to master the energy-management task. By comparing the initial random distributions at Generation 1 with the converged values at Generation 50, we can identify not only the specific survival strategies adopted by the elite populations but also the degree of evolutionary consensus, measured via the standard deviation, across the 10 independent runs.

To provide a comprehensive overview of this evolutionary process, table 4.6 summarizes the average parameter changes and their final standard deviations.

Category	Parameter	Gen 1 Mean	Gen 50 Mean	Gen 50 Std	Change (%)
Learning Rates	α_1 (Actor Hidden)	0.4923	0.3172	±0.1049	**-35.57%**
	α_2 (Actor Output)	0.4972	0.5045	±0.1601	+1.47%
	p_1 (Critic Hidden)	0.4962	0.5119	±0.1603	+3.16%
	p_2 (Critic Output)	0.5130	0.5369	±0.1468	+4.66%
Temporal Discount	γ (Critic)	0.4995	0.4023	±0.2176	**-19.46%**
Energy Thresholds	en_1 (Low Energy)	0.3500	0.3632	±0.0688	+3.77%
	en_2 (High Energy)	0.7497	0.7212	±0.0378	-3.79%
Soft-max Temperatures	β_1 (Critical Zone)	5.4617	4.6764	±1.1311	**-14.38%**
	β_2 (Transition)	5.5164	4.4046	±1.3483	**-20.16%**
	β_3 (Safe Zone)	7.8761	6.1664	±2.9788	**-21.71%**
Neural Architecture	Actor Hidden Neurons	2.4600	2.6350	±0.1908	+7.11%
	Critic Hidden Neurons	2.5250	2.4640	±0.1651	-2.42%

Table 4.6: Parameter evolution: Comparison of Generation 1 and Generation 50 averages (aggregated across 10 independent runs) for Scenario A, including the final Standard Deviation to highlight evolutionary consensus.

Learning Rates The evolution of the learning rates highlights a sophisticated decoupling between feature extraction and action selection. The Actor's hidden layer learning rate (α_1) experienced the most drastic reduction across all parameters, dropping to 0.3172 ± 0.1049. This decrease suggests that stabilizing the internal representations of the state space early on prevents catastrophic interference. Conversely, the learning rates for the output heads (α_2, p_1, and p_2) remained relatively high, stabilizing around 0.50 to 0.53. The moderate standard deviations (around ±0.15) for these output layers indicate that while a highly plastic output

is generally preferred to handle sudden environmental shifts, different evolutionary runs found slightly varying localized optima for this plasticity.

Temporal Discount The Critic's temporal discount factor (γ) underwent a notable reduction, decreasing from 0.4995 to 0.4023. However, **the standard deviation for γ is exceptionally high ($\pm$0.2176)**, with final values spanning from 0.099 to 0.781 across different runs. This high variance is a significant finding: it demonstrates that there is no single optimal time horizon for this highly non-stationary task. Depending on the initial random seeds and the specific evolutionary path, some populations evolved a hyper-myopic Critic ($\gamma \approx 0.10$) that prioritizes immediate survival, while others successfully maintained a more far-sighted approach ($\gamma \approx 0.78$). Both strategies ultimately yielded viable survivors, highlighting multiple pathways to robust energy management.

Energy Thresholds In stark contrast to the temporal discount, the energy boundaries that dictate behavioural switching demonstrated an extremely strong evolutionary consensus. The lower threshold settled at $en_1 = 0.3632$ with a remarkably low standard deviation of $\pm$0.0688, and the upper threshold settled at $en_2 = 0.7212$ with an **exceptionally tight standard deviation of** $\pm$0.0378. This lack of variance proves that these specific energy breakpoints are universally optimal. A critical urgency threshold near 36% provides just enough time buffer to locate resources, while a safe zone beginning at 72% reliably guarantees that the agent can exploit its environment without risking sudden starvation, regardless of the neural network's specific internal weights.

Soft-max Temperatures The Soft-max inverse temperatures ($\beta_1, \beta_2, \beta_3$), governing the exploration-exploitation trade-off, experienced significant overall reductions. However, analysing the variance reveals distinct behaviours depending on the energy state. When the robot is in a critical or transition state, the temperatures (β_1 and β_2) converge with moderate deviations ($\pm$1.13 and $\pm$1.34), enforcing a consistent level of exploration necessary to escape starvation. Conversely, **the Safe Zone temperature (β_3) exhibits an unusually high standard deviation ($\pm$2.9788)**, with elite configurations ranging from highly stochastic (2.40)

to highly deterministic (11.10). This high variance suggests a relaxation of evolutionary pressure: when the battery level is comfortably above en_2, the immediate penalty for making a suboptimal "exploratory" move is minimal. Consequently, the GA allows for a much wider diversity of behavioural strategies as long as the agent remains in the safe zone.

Neural Architecture The GA was given the freedom to choose the structural complexity of the Actor and Critic networks, ranging from 0 to 5 hidden neurons. By Generation 50, the mean number of hidden neurons for both the Actor (2.635) and the Critic (2.464) converged tightly around the middle of the available spectrum. The remarkably narrow standard deviations (± 0.1908 and ± 0.1651, respectively) act as definitive proof of structural consensus. Across all 10 independent experiments, the evolution unequivocally rejected both the overly simplistic linear models (which lack the capacity to map complex states) and the overly complex 4-5 neuron models (which are prone to over-fitting and computational inefficiency), zeroing in on 2 to 3 neurons as the perfect architectural balance.

Scenario B

The parameter evolution for the multi-colour contextual task (Scenario B) unveils a radically different set of survival strategies compared to the energy-management task. The introduction of the multi-headed architecture and the harsh penalties for incorrect contextual captures forced the Genetic Algorithm (GA) to find a highly specialized configuration. By analysing the shift from the first generation to the last, and critically examining the final standard deviations, we can observe an extraordinary level of evolutionary consensus.

The table 4.7 details the progression of the meta parameters across the 10 independent runs for Scenario B.

Learning Rates In stark contrast to Scenario A, where output learning rates remained high, Scenario B exhibits a universal and significant suppression of learning rates across all layers. Both the Actor's hidden (α_1) and output (α_2) learning rates plummeted by over 35%, stabilizing around 0.31. The remarkably narrow

Category	Parameter	First Gen Mean	Last Gen Mean	Last Gen Std	Change (%)
Learning Rates	α_1 (Actor Hidden)	0.4886	0.3140	±0.0764	**-35.73%**
	α_2 (Actor Output)	0.4989	0.3168	±0.0698	**-36.50%**
	p_1 (Critic Hidden)	0.4932	0.3922	±0.0638	-20.48%
	p_2 (Critic Output)	0.5063	0.4522	±0.0431	-10.68%
Temporal Discount	γ (Critic)	0.5021	0.6847	±0.0227	**+36.37%**
Energy Thresholds	en_1 (Low Energy)	0.3570	0.4392	±0.0369	**+23.00%**
	en_2 (High Energy)	0.7501	0.7369	±0.0555	-1.76%
Soft-max Temperatures	β_1 (Critical Zone)	5.4980	3.0468	±0.6646	**-44.58%**
	β_2 (Transition)	5.4818	2.2941	±0.2258	**-58.15%**
	β_3 (Safe Zone)	8.0477	12.1128	±0.5522	**+50.51%**
Neural Architecture	Actor Hidden Neurons	2.5310	2.7510	±0.1762	+8.69%
	Critic Hidden Neurons	2.4390	2.8980	±0.2870	+18.82%

Table 4.7: Parameter evolution: Comparison of First Generation and Last Generation averages (aggregated across 10 independent runs) for Scenario B, including the final Standard Deviation to highlight evolutionary consensus.

standard deviations (±0.0764 and ±0.0698) indicate a strong, universal evolutionary pressure. In a multi-headed architecture, high plasticity in the output layers is actively detrimental: if the network overwrites its synaptic weights too rapidly when adapting to the "Yellow" context, it will undergo catastrophic forgetting of the "Red" and "Green" contextual mappings. The GA thus evolved universally slow learning rates, forcing the neural network to build stable, generalized representations that preserve multi-colour knowledge across environmental swaps.

Temporal Discount The Critic's temporal discount factor (γ) evolved in the exact opposite direction compared to Scenario A, surging by 36.37% to reach a mean of 0.6847. Crucially, the standard deviation is incredibly tight (±0.0227), meaning every single successful population converged on this specific time horizon. While the purely spatial foraging of Scenario A favoured myopia, the contextual discrimination of Scenario B requires far-sightedness. Capturing the correct colour yields an immediate reward, but capturing the wrong colour triggers an immediate penalty that cascades into severe energy depletion. A higher γ enables the Critic to effectively link current target-selection actions to a longer sequence of future state evaluations, universally proving to be the optimal predictive strategy.

Energy Thresholds The evolution of the internal energy boundaries reflects an adaptation to the brutal penalty mechanics of the multi-colour task. The lower

threshold (en_1) increased significantly by 23.00%, moving from 0.3570 to 0.4392 with a low variance ($\pm$0.0369). This means the evolved agents enter a state of "critical urgency" much earlier—when their battery is at $\approx$ 44%, rather than 36%. Because capturing the wrong context accelerates energy drain exponentially, the robot requires a much larger time and energy buffer to safely "unlearn" the obsolete colour and discover the new active reward without dying in the process.

Soft-max Temperatures The Soft-max inverse temperatures exhibit the most extreme and fascinating evolutionary divergence in this study. The Safe Zone temperature (β_3) spiked massively to 12.1128, while the Transition (β_2) and Critical (β_1) temperatures crashed to 2.2941 and 3.0468, respectively. The exceedingly narrow standard deviations for β_2 ($\pm$0.2258) and β_3 ($\pm$0.5522) confirm that this radical bifurcation is the absolute definitive solution to the task. This creates an aggressive "strategy-switching" mechanism. When the agent is safely collecting the correct colour, energy is high ($> en_2$), and $\beta_3 \approx 12.11$ dictates an almost perfectly deterministic, greedy exploitation of the known policy. However, when the environment silently swaps the rewarding colour, the agent's greedy captures suddenly yield penalties. The battery level drops, crossing en_2 and entering the transition zone. Instantly, the temperature plunges toward $\beta_2 \approx 2.29$, triggering an explosive surge of behavioural stochasticity. This massive injection of exploration is precisely what allows the agent to break free from its obsolete policy and quickly sample the environment to discover the newly rewarded colour.

Neural Architecture The structural complexity required to solve the multi-colour task is slightly higher than that of the simple foraging scenario. Both networks expanded, with the Actor converging around 2.75 hidden neurons ($\pm$0.1762) and the Critic around 2.90 hidden neurons ($\pm$0.2870). The multi-headed architecture inherently demands a larger representational capacity in the shared hidden layer, as it must concurrently encode the spatial coordinates of multiple distinct target colours before routing them to the specialized behavioural heads. Once again, the tight variance demonstrates that linear mappings were entirely insufficient, but overly deep networks were systematically pruned by the GA.

4.2 Testing and Results

This chapter presents the crucial phase of experimental validation, where the controllers optimized in the previous stages are extracted from the evolutionary process and subjected to rigorous testing in environments not seen during training. The primary objective of this analysis is to verify the robustness and generalization capabilities of the developed Actor-Critic architectures, ensuring that the learned survival strategies and contextual discrimination are not tied to specific random configurations of the arena, but rather represent universal adaptive solutions for the proposed problems.

The transition from the optimization (training) phase to the verification (testing) phase allows for the quantification of the real value of the evolved meta parameters, distinguishing between simple numerical convergence and effective behavioural intelligence. Through an intensive testing campaign, which produced a total of 600 simulations across both scenarios, we will analyse how "best-in-class" agents perform when faced with unprecedented random seeds, evaluating whether battery stability and capture precision remains constant even under conditions of spatial uncertainty.

4.2.1 Testing Method

To rigorously evaluate the performance of the synthesized controllers, we implemented a post-evolutionary testing protocol aimed at verifying the robustness of the learned strategies in scenarios never encountered during optimization. While the evolutionary process identifies high-performing individuals within a population, it is essential to confirm that such results are statistically significant and not influenced by environmental "noise".

The test protocol was structured as follows:

- **Selection**: From each of the 10 independent experiments conducted for both scenarios (A and B), the best individual (Elite) from the final generation (Generation 50) was selected. To establish a comparative baseline, the best individuals from the first generation (Generation 1) were also extracted. This resulted in a set of 20 "best-in-class" controllers per scenario.

- **Cross-Validation**: Each controller was subjected to 30 independent test simulations. To ensure the impartiality of the results, each simulation was initialized with a unique random seed (in the range 1000–1029) that was not utilized during the training phase.

- **Configuration**: The testing environment maintained the same non-stationary dynamics as the training phase (e.g., resource swaps every 10,800 ticks).

This approach generated a dataset of 300 results for the final generation per scenario (600 simulations total when including the first generation baseline), allowing for an accurate statistical analysis of the *Success Rate* and the *Mean Fitness*. By comparing these 300 "`last_gen`" runs against the 300 "`first_gen`" runs, we can objectively measure the improvement gained through the evolutionary synthesis process.

4.2.2 Results

Scenario A

Best Controllers For the validation and testing phase, we selected the **elite individual** (the agent with the highest training fitness) from the final generation (Generation 50) of each of the 10 independent evolutionary runs. These ten controllers, identified as the "best-in-class" for their respective evolutionary trajectories, serve as the primary subjects for evaluating the robustness and generalization of the synthesized behaviours.

As detailed in table 4.8 and table 4.9, the selected set exhibits a high degree of **structural and behavioural heterogeneity**, confirming the Genetic Algorithm's ability to discover multiple distinct optima within the high-dimensional continuous search space.

Architecturally, the controllers range from purely linear mappings with zero hidden neurons (e.g., GA 7 and GA 9) to more complex, deep configurations featuring up to five hidden neurons in the Actor and Critic networks (e.g., GA 2 and GA 4). The **learning rates** (table 4.8) also display massive variance. Some elites evolved extremely low learning rates for feature extraction ($\alpha_1 = 0.06$ in

GA ID	Fitness	Hidden Neurons	Actor Learning Rates		Critic Learning Rates	
		(Actor / Critic)	α_1 (Hidden)	α_2 (Output)	p_1 (Hidden)	p_2 (Output)
GA 1	1.9086	1 / 1	0.3136	0.3994	0.9805	0.6981
GA 2	1.9116	5 / 0	0.2458	0.8158	0.9404	0.9216
GA 3	1.9124	5 / 3	0.1040	0.5086	0.6769	0.2021
GA 4	1.9278	4 / 5	0.1959	0.0755	0.2832	0.8530
GA 5	1.8773	3 / 0	0.1705	0.3228	0.3718	0.8823
GA 6	1.9245	3 / 0	0.9629	0.5360	0.0570	0.1542
GA 7	1.9121	0 / 0	0.0634	0.8743	0.8740	0.0630
GA 8	1.9139	3 / 3	0.0675	0.2836	0.7934	0.8507
GA 9	1.8766	0 / 4	0.4767	0.4928	0.3276	0.7174
GA 10	1.8879	4 / 4	0.5839	0.0040	0.2369	0.2312

Table 4.8: Architectural configurations and Learning Rates of the 10 selected elite controllers from Generation 50 (Scenario A).

GA 7 and GA 8), suggesting a highly stable internal representation, while others maintained high plasticity across all layers (e.g., GA 1 and GA 2).

GA ID	Critic Discount	Energy Thresholds		Soft-max Temperatures		
	γ	en_1 (Low)	en_2 (High)	β_1 (Critical)	β_2 (Transition)	β_3 (Safe)
GA 1	0.5523	0.4710	0.6916	1.0065	1.8136	2.3140
GA 2	0.5544	0.5142	0.7272	1.4689	5.7524	3.6954
GA 3	0.3202	0.6000	0.6368	1.2118	6.8075	3.8163
GA 4	0.8943	0.3843	0.8001	6.2579	1.0000	11.8065
GA 5	0.8881	0.4522	0.8235	6.1592	2.7866	13.0292
GA 6	0.8597	0.3872	0.7339	7.4376	8.1727	3.0391
GA 7	0.6568	0.4922	0.8379	6.1438	6.1205	12.2401
GA 8	0.5216	0.5467	0.7202	4.3093	5.9567	8.5155
GA 9	0.0000	0.2877	0.6278	2.6503	3.0479	1.0000
GA 10	0.0000	0.3363	0.7851	1.6085	1.0000	1.0000

Table 4.9: Behavioural meta parameters (Temporal Discount, Energy Thresholds, and Soft-max Temperatures) for the 10 selected elite controllers.

This diversity is equally evident in the evolved behavioural meta parameters (table 4.9). The temporal discount factor (γ) varies from purely reactive values ($\gamma = 0.0$ in GA 9 and GA 10), where the Critic evaluates states based exclusively on immediate energy rewards, to far-sighted configurations ($\gamma > 0.85$ in GA 4, 5, and 6), where the agent attempts to optimize long-term energy stability. Furthermore, the dynamic exploration-exploitation strategies, strictly governed by the β anchors and the energy boundaries (en_1, en_2), show unique calibrations. Despite these

profound differences in "internal" configuration and spatial representations, all ten selected controllers achieved high fitness scores (> 1.87), indicating multiple viable strategies to master the non-stationary survival constraints.

Fitness and Success Rate The detailed analysis of the test simulations reveals a stark bifurcation in the generalization capabilities of the evolved controllers. As summarized in table 4.10 and visually supported by the fitness distributions in fig. 4.5, the testing phase successfully identified which evolutionary trajectories produced universally robust behaviours and which fell victim to environmental over-fitting.

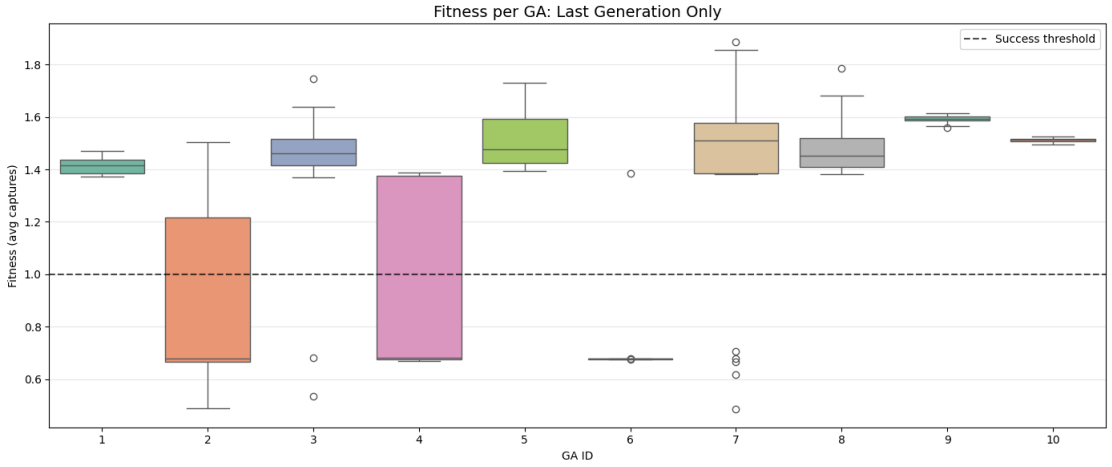

Figure 4.5: Distribution of fitness results across 30 independent seeds, totalling 300 simulations. The box plots highlight the variance and interquartile ranges of the elite controllers.

A remarkable 50% of the tested populations (GAs 1, 5, 8, 9, and 10) synthesized what can be defined as "perfect" generalizing controllers. These agents achieved a flawless **100% Success Rate** across all 30 unseen test seeds. As visually evident in the compressed box plots of fig. 4.5, top-performing elites like GA 9 and GA 10 not only survived every simulation but did so with an exceptionally tight fitness standard deviation (± 0.01). This lack of variance demonstrates that their energy-management strategies are virtually immune to the initial spatial distribution of resources.

Controller	Mean Fitness	Fitness Std Deviation	Success Rate (%)	Success Std Deviation
GA 1	1.41	± 0.03	**100.00%**	± 0.00
GA 2	0.59	± 0.44	16.67%	± 0.38
GA 3	1.42	± 0.24	93.33%	± 0.25
GA 4	0.98	± 0.35	43.33%	± 0.50
GA 5	1.51	± 0.10	**100.00%**	± 0.00
GA 6	0.70	± 0.13	3.33%	± 0.18
GA 7	1.40	± 0.38	83.33%	± 0.38
GA 8	1.49	± 0.11	**100.00%**	± 0.00
GA 9	1.59	± 0.01	**100.00%**	± 0.00
GA 10	1.51	± 0.01	**100.00%**	± 0.00

Table 4.10: Testing phase results: Mean Fitness and Success Rate across 30 random seeds for the elite controllers of the final generation (Scenario A).

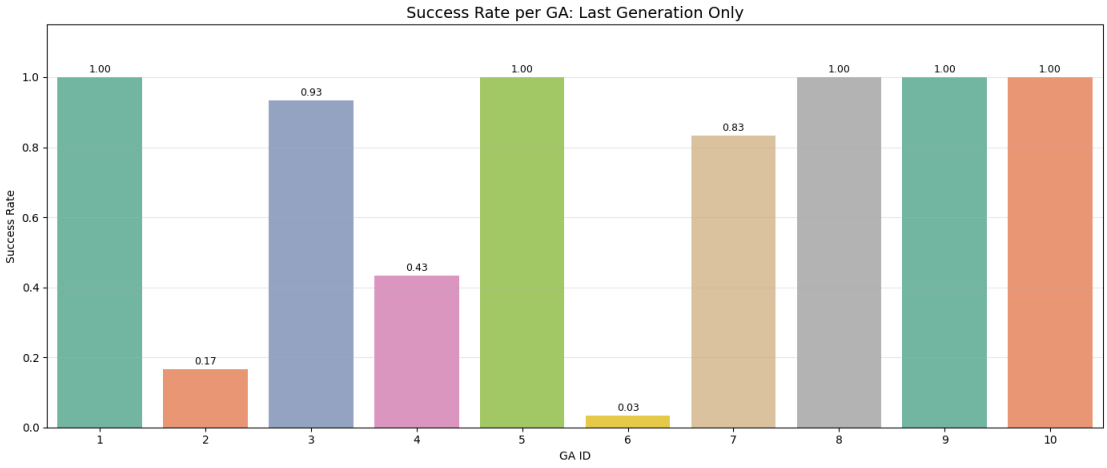

Figure 4.6: Distribution of success rate results across 30 independent seeds, totalling 300 simulations, illustrating the stark contrast between generalizing and over-fitted controllers.

The disparities in survival are immediately apparent in the success rate chart (fig. 4.6). The data clearly highlights that, although all elite controllers achieved high fitness scores (> 1.87) during the training phase, their actual generalization capabilities on novel environmental seeds vary drastically. Analysing these failure cases is crucial for understanding the inherent limitations of evolutionary optimization.

Looking at the **GA 2** controller, we observe a steep collapse in performance: the *Mean Fitness* drops to 0.59, associated with a *Success Rate* of merely 16.67%. Since in our evaluation framework a fitness score below 1.0 mathematically equates to premature death via battery depletion, this indicates that in over 80% of the test simulations, the agent failed to survive the full duration. The strategy learned by GA 2 proved to be highly fragile: while highly performant in the specific spatial configurations encountered during training, it was incapable of sustaining the robot when the resource distribution and spawn points were newly randomized.

However, the most emblematic case of systematic failure is represented by the **GA 6** controller. With a near-zero *Success Rate* (3.33%) and a high standard deviation during failures, this individual serves as a textbook example of environmental **over-fitting**. Over the 50 generations of evolution, the neural networks of GA 6 (which, as previously noted, utilized a highly far-sighted discount factor of $\gamma = 0.8597$) evidently memorized specific trajectories or exploited statistical anomalies tied exclusively to the training seeds. The algorithm optimized its synaptic weights for a specific map layout, failing to generalize the fundamental cognitive concepts of exploration and energy conservation. Consequently, when faced with new spatial topologies, the robot was completely "disoriented" and rendered entirely ineffective.

These failure cases, while seemingly negative, strongly confirm the absolute necessity and validity of the *cross-validation* protocol adopted in this study. The GA is a blind optimizer that will exploit any characteristic of the simulated environment to maximize its objective function. Testing the elite individuals against a broad battery of 30 unseen random seeds is the only rigorous methodological tool that allows us to filter out brittle, over-fitted solutions (like GA 2 and GA 6) and reliably validate the genuinely intelligent and universally robust architectures (like GA 9 and GA 10).

Comparative Analysis To accurately quantify the true algorithmic gain achieved by the Genetic Algorithm, a direct comparative analysis was conducted between the best individuals of the initial, randomly instantiated population (First Generation) and the fully optimized elites (Last Generation). Evaluating both cohorts across the same 30 unseen testing seeds provides a definitive measure of how much generalizable intelligence was actually synthesized during the 50 generations of training. The visual distributions of these shifts are captured in fig. 4.7 for the Mean Fitness and fig. 4.8 for the Success Rate.

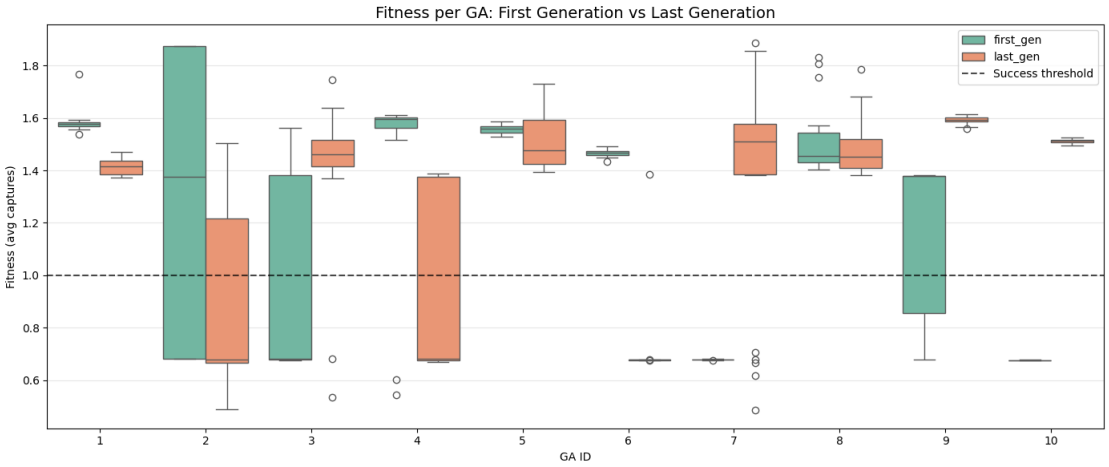

Figure 4.7: Comparative analysis of fitness distribution between the initial and final generations. The plot illustrates the overall fitness improvement across 30 independent runs.

The statistical data reveals a heterogeneous distribution of results, characterized by improvements, stable refinements, and explicit cases of algorithmic regression. We can categorize the evolutionary trajectories into three distinct behavioural phenomena:

1. Transformative Learning and Massive Improvement: The most compelling evidence of the GA efficacy is observed in runs where the first generation was completely incapable of surviving, yet the final generation achieved near-perfect mastery. **GA 10** serves as the prime example: its initial elite scored a mere 0.68 in mean fitness (indicating consistent, premature battery depletion), but the 50th-generation controller surged to 1.51, marking a massive +0.84 improve-

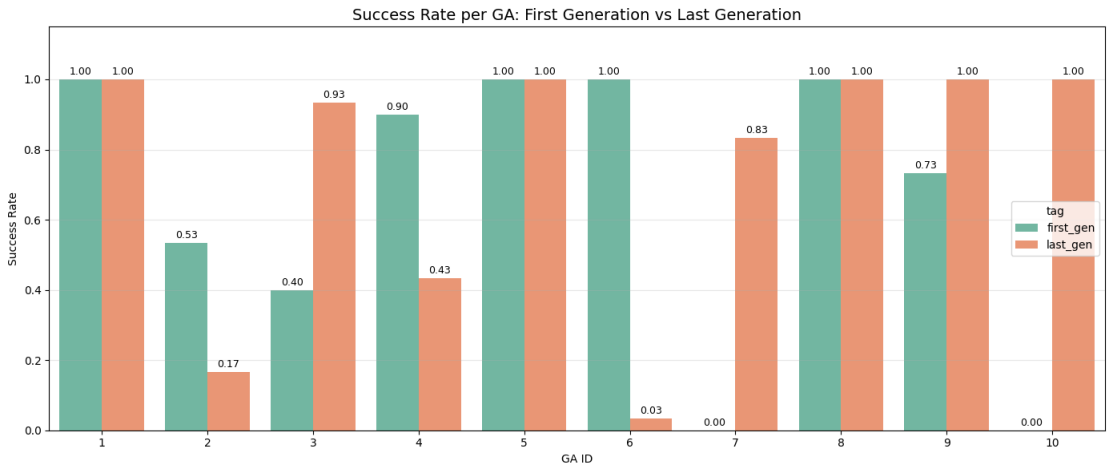

Figure 4.8: Comparative analysis of success rate distribution between the initial and final generations. The plot illustrates the overall success rate improvement across 30 independent runs.

ment. Similarly, **GA 7** climbed from 0.68 to 1.40 (+0.72), and **GA 3** improved from 0.97 to 1.42 (+0.45). In these experiments, the evolutionary pressure successfully broke the agents out of failing paradigms, discovering and refining robust energy-management strategies that generalized exceptionally well to novel spatial distributions.

2. Fortuitous Initialization and Stable Retention: Interestingly, some runs (such as **GA 1**, **GA 5**, and **GA 8**) began with extraordinarily high fitness scores in the first generation (1.58, 1.56, and 1.51, respectively). This phenomenon occurs when the random weight initialization fortuitously yields a "good enough" baseline policy for survival. In these cases, the absolute improvement metric appears negligible or slightly negative (e.g., GA 8 showed a -0.02 change). However, rather than signifying a failure, this indicates that the GA successfully protected these optimal topological mappings from destructive mutations, fine-tuning their stability and ensuring a perfect 100% success rate in the final testing phase.

3. Environmental Over-fitting and Evolutionary Regression: Perhaps the most analytically significant outcomes are the severe regressions observed in **GA 4** (-0.54) and, most drastically, **GA 6** (-0.77). In the case of GA 6, the randomly generated first-generation controller was surprisingly robust (fitness 1.47).

Yet, after 50 generations of intensive "optimization", the final controller plummeted to a fitness of 0.70 when evaluated on the test seeds. This provides a textbook demonstration of *environmental over-fitting*. The evolutionary process effectively dismantled a naturally generalizable baseline to aggressively exploit the specific spatial anomalies of the training maps. While this greedy optimization likely maximized the training fitness, it resulted in a brittle, hyper-specialized network that collapsed completely when the resource spawn locations were randomized during testing.

In conclusion, this comparative analysis validates the overall architectural design while highlighting the unpredictable nature of meta-heuristic optimization. The data confirms that while a randomly initialized Actor-Critic network occasionally possesses baseline survival traits, the Genetic Algorithm is absolutely critical for systematically elevating failing models to a 100% survival rate (as seen in GA 9 and GA 10). Simultaneously, the regressions underscore the vital necessity of the cross-validation testing protocol to filter out runs that fall into deceptive, over-fitted local optima.

Overall Statistics and Significance Testing To evaluate the macro-level impact of the evolutionary process across the entire experimental suite, the test results were aggregated to compare the global performance of all First Generation elites ($n = 300$ simulations) against all Last Generation elites ($n = 300$ simulations). To rigorously determine whether the observed improvements were the result of the optimization process rather than random variance, a non-parametric Wilcoxon test was conducted with a significance level of $\alpha = 0.05$.

The most critical metric for evaluating the autonomous agents in this scenario is the *Success Rate*—the ability to continuously manage energy and survive the entire testing period without battery depletion. As illustrated in fig. 4.9, the global success rate improved from a mean of 65.67% in the initial generation to 74.00% in the final generation, representing a relative improvement of 12.7%. The Wilcoxon test confirmed that this increase is **statistically significant** ($U = 41250.0$, $p = 0.0263 < 0.05$). This definitively proves that the Genetic Algorithm successfully synthesized more robust, universally applicable survival traits across the population of controllers.

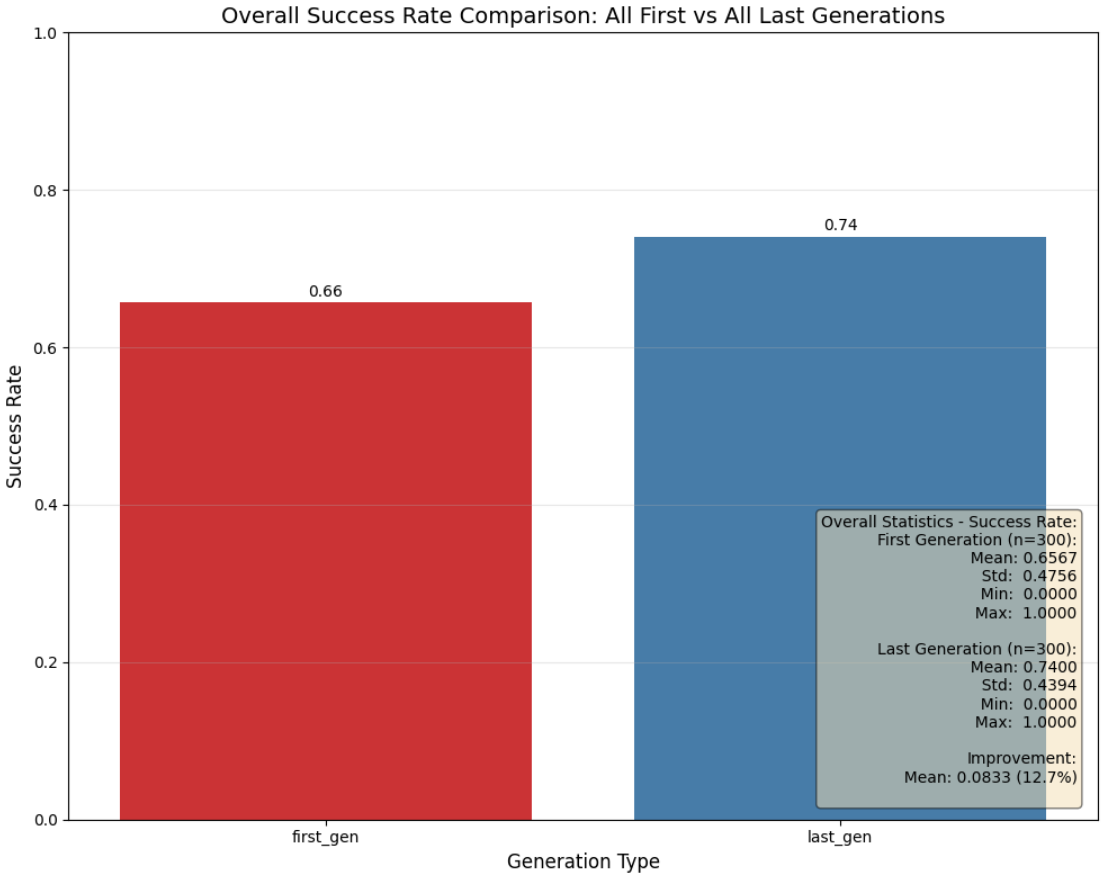

Figure 4.9: Bar graph showing the overall success rate difference between the first and last generations. The evolutionary process significantly increased the global survival probability.

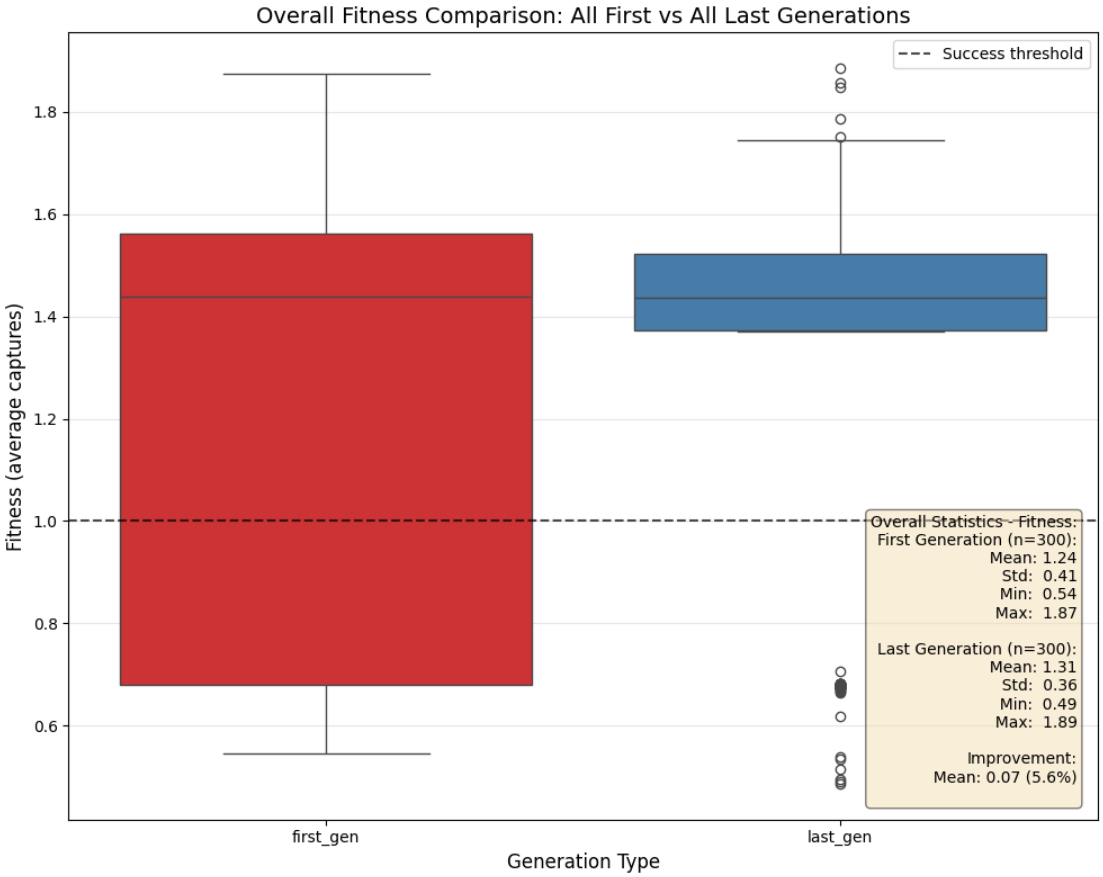

Figure 4.10: Box plot showing the overall fitness distribution between the first and last generations. The final generation exhibits a higher median and a more compressed interquartile range.

When examining the overall *Mean Fitness* (depicted in fig. 4.10), the global average increased from 1.24 ± 0.41 to 1.31 ± 0.36, representing a 5.6% objective improvement. While this percentage increase might initially appear modest, a closer inspection of the data reveals a clear and tangible upgrade in the agents' energy-management behaviours. As visually evident in the box plot, the median fitness of the Last Generation is noticeably higher, and the interquartile range is far more compressed. Furthermore, the reduction in standard deviation (from 0.41 to 0.36) indicates that the evolutionary process successfully pruned the most erratic, inefficient behaviours present in the initial random populations. The GA established a higher, more stable "performance floor", ensuring that even the lower-performing optimized agents maintained better energy reserves than their unoptimized counterparts.

Despite these observable, qualitative upgrades in behavioural consistency and energy accumulation, the statistical analysis via the Wilcoxon test revealed that this specific shift in the global fitness distribution is ultimately **not statistically significant** ($U = 42154.0$, $p = 0.6608 > 0.05$).

This divergence in statistical significance between survival (Success Rate) and energy accumulation (Fitness) offers a profound insight into the evolutionary dynamics of the task. It suggests that a fortuitously initialized random network (First Gen) can sometimes achieve peak energy levels comparable to an optimized one simply due to lucky spatial spawns. However, absolute peak energy is not the primary marker of intelligence in a non-stationary environment; *reliability* is. The statistically significant improvement in the success rate demonstrates that the true achievement of the GA was eliminating fatal cognitive flaws, ensuring that the agents could reliably survive environmental volatility regardless of where the resources appeared.

Behaviour Analysis Analysing the behavioural adaptation in the energy-based survival task (Scenario A) requires a nuanced approach to the data. Unlike standard classification tasks, the available actions in this scenario operate on fundamentally different time scales and execution frequencies. Specifically, `Capture` and `Wait` are durative actions that the agent can sustain over hundreds of consecutive control cycles, whereas `Search` is a discrete, terminating event that only

occurs upon successfully reaching a resource. Consequently, plotting raw execution counts on a single linear scale obscures the agent's strategy, as the sheer volume of navigation and resting cycles inherently dwarfs the capture events.

Therefore, the true indicator of online adaptation is observed by tracking how the statistical distribution of each individual action shifts in response to the changing environments, as summarized in Table 4.11.

Environment	**Action**	**Mean**	**Std Dev**	**Median**	**Max**
Environment 1	Wait	659.58	373.72	872.50	1372
	Search	958.92	937.15	361.50	2778
	Capture	278.11	185.35	188.50	572
Environment 2	Wait	686.68	411.05	962.00	1354
	Search	1076.81	1416.20	90.00	3559
	Capture	94.64	97.91	62.00	451
Environment 3	Wait	884.24	1455.72	949.50	10799
	Search	953.11	1148.63	149.50	3183
	Capture	125.81	101.79	90.50	494
Environment 4	Wait	654.69	433.29	958.50	1067
	Search	1187.94	1546.34	121.00	3565
	Capture	64.02	86.03	45.00	480

Table 4.11: Statistical distribution of execution counts for `Wait`, `Search`, and `Capture` actions across the four testing environments in Scenario A. Data collected from the final generation of evolved controllers.

In the baseline scenario, Environment 1, the agent operates in what can be characterized as a resource-rich state. The data reflects a highly active and successful foraging strategy: `Capture` events are at their absolute peak (mean = 278.11, median = 188.50). Correspondingly, the agent maintains a balanced energy expenditure, alternating between active `Search` (median = 361.50) and conservative `Wait` periods (median = 872.50).

However, as the agent is tested across Environments 2, 3, and 4—where resource density decreases or environmental constraints become harsher—a distinct

and vital strategic shift emerges. The frequency of `Capture` events drops precipitously across these testing phases, reaching a minimum median of just 45.00 in Environment 4. Faced with this severe scarcity, the evolved Actor-Critic controllers successfully exhibit the desired energy-preserving behaviour.

To survive the lack of resources, the agents significantly increase their reliance on the `Wait` action. The median waiting time rises systematically, peaking at 962.00 in Environment 2 and maintaining high levels (949.50 and 958.50) in Environments 3 and 4. Most notably, in Environment 3, the maximum `Wait` time spikes to an extreme 10799 cycles. This demonstrates that when energy drops and foraging becomes unprofitable, the agent intentionally shifts its policy towards extreme conservation, halting its motors to minimize battery consumption.

Furthermore, the `Search` action in the harsher environments exhibits a massive divergence between its mean and median values (e.g., in Environment 4, mean = 1187.94 vs. median = 121.00, with a standard deviation of 1546.34). This high variance perfectly encapsulates the dynamic Soft-max temperature mechanism triggered by the internal energy level. The agent does not simply roam aimlessly; it remains in a low-energy `Wait` state (hence the low median search times) until its battery crosses a critical depletion threshold. At that exact moment, the exploration rate β forces an explosive spike in stochastic behaviour, causing the sudden, desperate `Search` bursts (reflected by the high maximum values up to 3565) needed to find a battery and avoid immediate "death".

Ultimately, these statistics validate the hybrid evolutionary approach: the robots do not execute a static routine, but autonomously modulate their exploration-exploitation ratio based on their internal physiological state and the external environment's unforgiving dynamics.

Scenario B

Best Controllers For the experimental validation of Scenario B (the multi-colour contextual task), the best individuals (elites) were extracted from the final generation (Generation 50) of each of the 10 independent evolutionary runs. These ten controllers constitute the reference sample for evaluating the optimal configurations identified by the Genetic Algorithm (GA) in an environment characterized

by dynamic penalties and variable rules

The architectural and parametric data of these controllers, presented in table 4.12 and table 4.13, highlight a uniform structural convergence. Unlike Scenario A, where the algorithm produced heterogeneous topologies, all 10 experiments in Scenario B independently converged on a purely linear architecture (0 hidden neurons for both the Actor and the Critic).

This architectural consensus indicates that, within the specific multi-head configuration utilized, the presence of a shared hidden layer proves disadvantageous. The use of deep latent representations in tasks subject to rapid context shifts exposes the network to the risk of catastrophic interference (catastrophic forgetting): updating the shared weights required to adapt to a new chromatic rule would result in the degradation of the mappings learned for the other contexts. The reduction to a direct linear sensor-to-action mapping minimizes interference between contexts, delegating behavioural specialization exclusively to the separate output layers.

GA ID	**Fitness**	**Hidden Neurons**	**Actor Learning Rates**		**Critic Learning Rates**	
		(Actor / Critic)	α_1 **(Hidden)**	α_2 **(Output)**	p_1 **(Hidden)**	p_2 **(Output)**
GA 1	4.4624	0 / 0	0.3000	0.1813	0.6785	0.6749
GA 2	4.4335	0 / 0	0.0933	0.3414	0.2565	0.3050
GA 3	4.4506	0 / 0	0.3048	0.1739	0.1322	0.8018
GA 4	4.3899	0 / 0	0.2436	0.2701	0.2638	0.3834
GA 5	4.3726	0 / 0	0.1542	0.3012	0.1916	0.5206
GA 6	4.4951	0 / 0	0.2310	0.2188	0.3395	0.3761
GA 7	4.3713	0 / 0	0.4339	0.1098	0.5022	0.4270
GA 8	4.4271	0 / 0	0.2407	0.3138	0.3237	0.5263
GA 9	4.3842	0 / 0	0.4198	0.3804	0.3900	0.3383
GA 10	4.3678	0 / 0	0.2384	0.2684	0.2334	0.4083

Table 4.12: Architectural configurations and learning rates of the 10 selected elite controllers from Generation 50 for Scenario B. Note the unanimous convergence on zero hidden neurons.

This structural configuration is accompanied by stabilized learning rates. Specifically, the Actor's output learning rate (α_2) settles at moderate values (between 0.10 and 0.38). This limited plasticity acts as a stabilization mechanism, preventing the agent from prematurely overwriting the learned policy in response to isolated penalties received during the environmental transition phase.

GA ID	Critic Discount	Energy Thresholds		Soft-max Temperatures		
	γ	en_1 **(Low)**	en_2 **(High)**	β_1 **(Critical)**	β_2 **(Transition)**	β_3 **(Safe)**
GA 1	0.7559	0.5222	0.6342	1.0000	1.0000	12.9336
GA 2	0.7873	0.3923	0.6923	1.2082	1.2512	7.4356
GA 3	0.8382	0.2749	0.8404	6.1086	1.0000	13.3347
GA 4	0.7242	0.4442	0.6099	2.7527	1.2067	9.7509
GA 5	0.7452	0.6000	0.9000	1.1044	1.0000	15.0000
GA 6	0.6353	0.5219	0.7616	1.0000	1.1963	12.7769
GA 7	0.6711	0.4759	0.7206	4.3895	1.0000	15.0000
GA 8	0.7868	0.4732	0.6696	1.0000	1.0000	11.3963
GA 9	0.7383	0.4833	0.7428	1.4180	1.0000	15.0000
GA 10	0.6963	0.3270	0.7692	3.0003	1.0000	14.2442

Table 4.13: Behavioural meta parameters (temporal discount factor, energy thresholds, and Soft-max temperatures) for the 10 selected elite controllers in Scenario B.

The behavioural meta parameters (table 4.13) further corroborate a rigidly specialized strategy. In contrast to the myopic policies observed in Scenario A, the temporal discount factor (γ) remains consistently high across all controllers (ranging from 0.6353 to 0.8382). Given the severity of the penalty for capturing an incorrect colour, the agents are required to evaluate the long-term consequences of their actions to prevent critical energy depletion.

Finally, the Softmax temperature anchors reveal a highly reactive exploration-exploitation switching mechanism. The Safe Zone temperature (β_3) is universally high (predominantly between 11.0 and 15.0), dictating a deterministic exploitation policy when the correct colour rule is established. However, the Transition temperature (β_2) drops to its mathematical lower bound (≈ 1.00) in nearly every instance. This demonstrates that the optimal survival strategy for this task necessitates an immediate transition from high certainty to substantial stochasticity (exploration) as soon as the energy level falls below en_2, compelling the agent to rapidly sample the environment to identify the newly rewarded context.

Fitness and Success Rate The detailed analysis of the test simulations for Scenario B reveals the inherent complexity of the multi-colour contextual task and highlights a pronounced variance in the generalization capabilities of the evolved controllers. As summarized in table 4.14 and visually supported by the distribu-

tions in fig. 4.11 and fig. 4.12, the evaluation against 30 novel environmental seeds successfully discriminated between robust, rule-abstracting behaviours and brittle, over-fitted strategies.

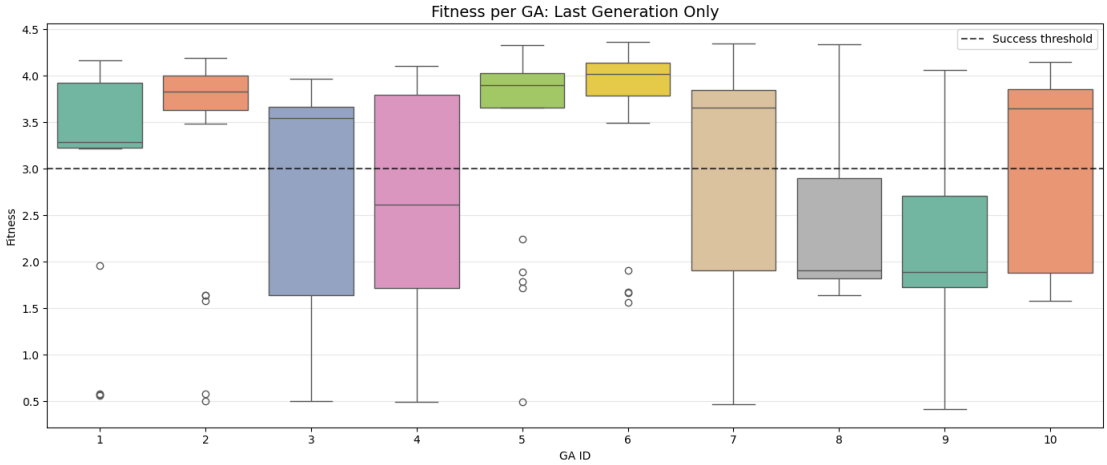

Figure 4.11: Distribution of fitness results across 30 independent seeds, totalling 300 simulations for Scenario B. The box plots highlight the variance and interquartile ranges of the elite controllers.

Controller	Mean Fitness	Fitness Std Deviation	Success Rate (%)	Success Std Deviation
GA 1	3.22	± 1.04	80.00%	± 0.41
GA 2	3.35	± 1.14	66.67%	± 0.48
GA 3	2.77	± 1.35	43.33%	± 0.50
GA 4	2.49	± 1.37	30.00%	± 0.47
GA 5	3.48	± 1.05	60.00%	± 0.50
GA 6	3.69	± 0.83	**83.33%**	± 0.38
GA 7	3.01	± 1.32	60.00%	± 0.50
GA 8	2.48	± 1.12	6.67%	± 0.25
GA 9	2.18	± 1.04	16.67%	± 0.38
GA 10	3.04	± 1.03	36.67%	± 0.49

Table 4.14: Testing phase results: Mean Fitness and Success Rate across 30 random seeds for the elite controllers of the final generation (Scenario B).

The introduction of dynamic, context-dependent penalties rendered Scenario B substantially more challenging than the pure spatial foraging of Scenario A. Consequently, no single controller achieved a flawless 100% survival rate across all testing environments. The most robust generalization was exhibited by the elite

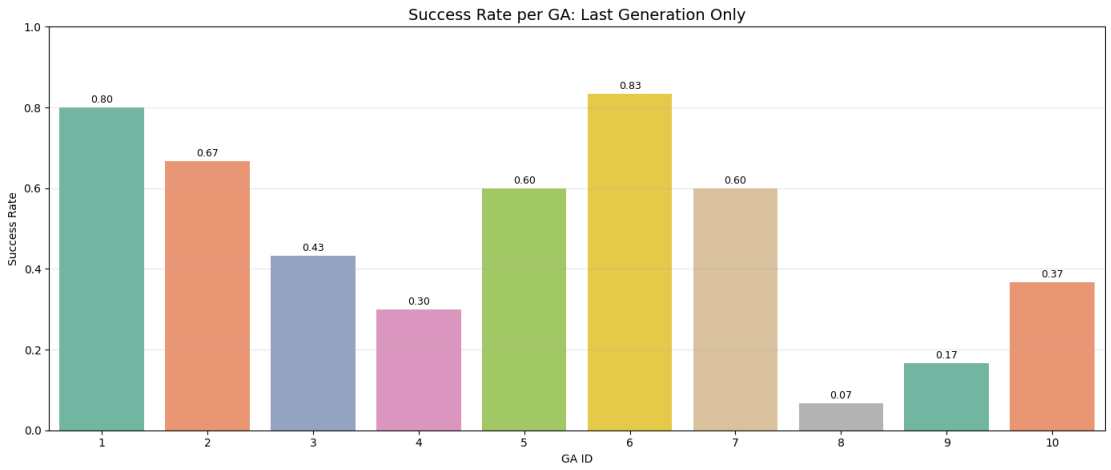

Figure 4.12: Distribution of success rate results across 30 independent seeds, totalling 300 simulations for Scenario B, illustrating the stark contrast between generalizing and over-fitted controllers.

controller from **GA 6**, which secured the highest Mean Fitness (3.69 ± 0.83) and the highest Success Rate (83.33%). The controller from **GA 1** also demonstrated strong adaptability, successfully managing energy transitions in 80.00% of the independent test runs.

Conversely, explicit cases of environmental over-fitting are evident within the dataset. Controllers such as **GA 8** and **GA 9**, despite achieving high fitness scores during the training generations (exceeding 4.38), suffered catastrophic performance degradation during testing. Their success rates plummeted to 6.67% and 16.67%, respectively. This severe drop indicates that their neural networks became hyper-specialized to the specific spatial layouts and sequential colour distributions of the training seeds. Lacking true behavioural abstraction, these agents failed to dynamically adjust their multi-headed policies when the environmental transition parameters were randomized.

Comparative Analysis To accurately quantify the learning efficacy of the Genetic Algorithm within the complex contextual demands of Scenario B, a direct comparison was conducted between the initially instantiated populations (First Generation) and the fully optimized elites (Last Generation). Evaluating both

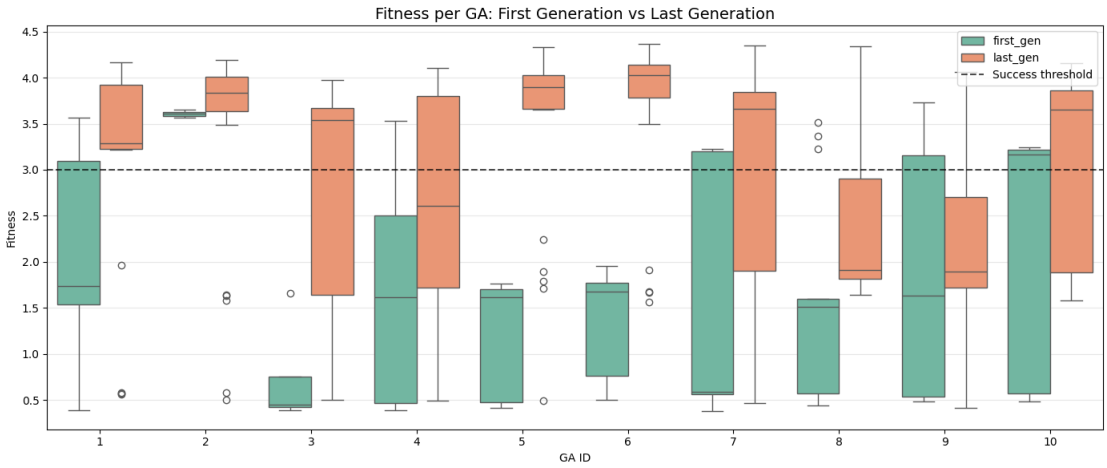

Figure 4.13: Comparative analysis of fitness distribution between the initial and final generations. The plot illustrates the overall fitness improvement across 30 independent runs.

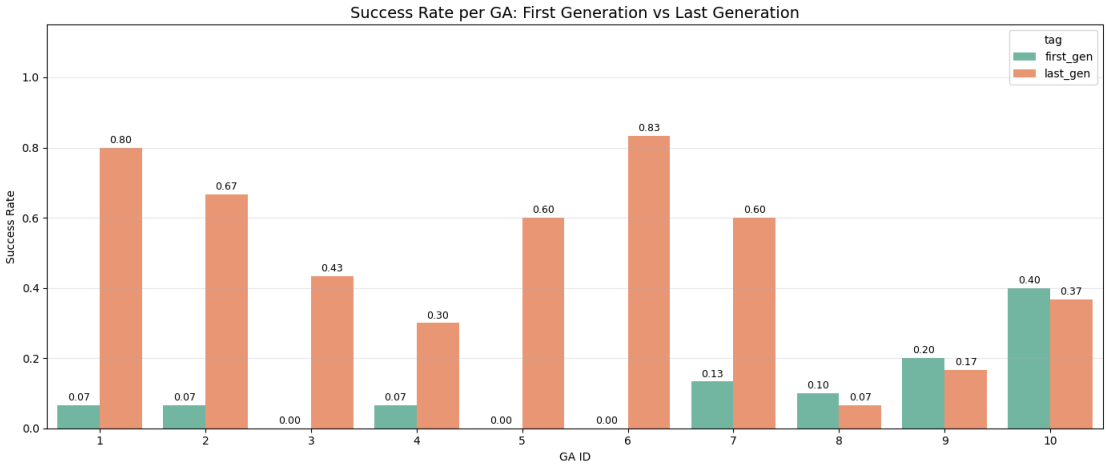

Figure 4.14: Comparative analysis of success rate distribution between the initial and final generations. The plot illustrates the overall success rate improvement across 30 independent runs.

cohorts across the same 30 unseen testing seeds provides a definitive measure of the generalizable intelligence synthesized during the evolutionary process. The distributions of these shifts are visually captured in fig. 4.13 and fig. 4.14.

The global metrics demonstrate a profound algorithmic impact. The overall mean success rate surged, rising from a baseline of 10.33% in the First Generation to 48.33% in the Last Generation. Correspondingly, the mean fitness improved by 85.9% (from 1.65 to 3.07). The per-GA statistical breakdown of these improvements reveals three distinct evolutionary phenomena that characterize the optimization landscape of this multi-colour task:

- **Transformative Optimization (Learning Contextual Rules):** The most compelling evidence of the algorithm's capability is observed in runs where the initial random networks were completely incapable of surviving, yet the final generation achieved strong contextual mastery. Controllers from GA 3, GA 5, and GA 6 recorded a First Generation success rate of exactly 0.00%. However, through 50 generations of selective pressure, the algorithm successfully synthesized the complex logic required for the task, elevating GA 6 to an 83.33% survival rate (+83.33% improvement), GA 5 to 60.00%, and GA 3 to 43.33%. Similarly, GA 1 and GA 2 achieved massive improvements in success rate (+73.33% and +60.00%, respectively), proving the algorithm's ability to construct complex, multi-headed switching strategies from scratch.

- **Partial Stagnation and Environmental Over-fitting:** Despite the widespread success, the data also exposes instances where the evolutionary process struggled to generalize. Controllers such as GA 8 and GA 9 started with low but non-zero survival probabilities (10.00% and 20.00%, respectively). However, after 50 generations of training, their testing success rates actually decreased slightly (−3.33%). While their training fitness likely improved by memorizing specific spatial layouts, they failed to learn the underlying colour-switching logic. Consequently, when evaluated on novel seeds, their hyper-specialized networks collapsed, revealing a severe case of environmental over-fitting. GA 10 exhibited a similar, though less drastic, stagnation, dropping from an initial 40.00% success rate to 36.67%.

- **The Fallacy of Fortuitous Initialization:** A highly unusual phenomenon occurred in GA 2. Its First Generation elite achieved the highest average fitness of any initial network (3.61) but possessed a catastrophic success rate of only 6.67%. This indicates a highly erratic policy: it occasionally gathered massive amounts of energy through sheer luck but died almost instantly in 93% of the testing environments. The Genetic Algorithm successfully dismantled this dangerous, high-variance policy. While the final fitness dropped slightly (-0.26 to 3.35), the success rate skyrocketed to 66.67% (+60.00%). The algorithm correctly prioritized consistent, cautious survival over reckless, lucky energy accumulation.

These contrasting outcomes validate the absolute necessity of the cross-validation testing protocol. While the global metrics unequivocally confirm the Genetic Algorithm's ability to synthesize advanced contextual intelligence, the per-GA breakdown highlights that high training fitness alone is an insufficient metric. Extensive testing on unseen environments is required to distinguish genuinely adaptive rules from localized spatial memorization.

Overall Statistics and Significance Testing To evaluate the macro-level impact of the evolutionary process across the entire experimental suite for the contextual task, the test results were aggregated to compare the global performance of all First Generation networks ($n = 300$ simulations) against all Last Generation elites ($n = 300$ simulations).

The overarching metric of viability in this non-stationary environment is the *Success Rate*. As illustrated in fig. 4.15, the global success rate exhibited a massive relative improvement of 367.7%. The initial random populations struggled severely with the dynamic penalties, achieving a mean survival rate of only $10.33\% \pm 0.30$. By the final generation, the global mean had risen to $48.33\% \pm 0.50$. While the absolute success rate remains lower than the simpler foraging task of Scenario A, the sheer magnitude of the relative increase highlights the algorithm's capacity to synthesize complex, multi-context logic from a near-zero baseline.

A parallel improvement is evident in the global *Mean Fitness* (fig. 4.16). The average energy accumulation score improved by 85.9%, climbing from 1.65 ± 1.17 in the first generation to 3.07 ± 1.19 in the last.

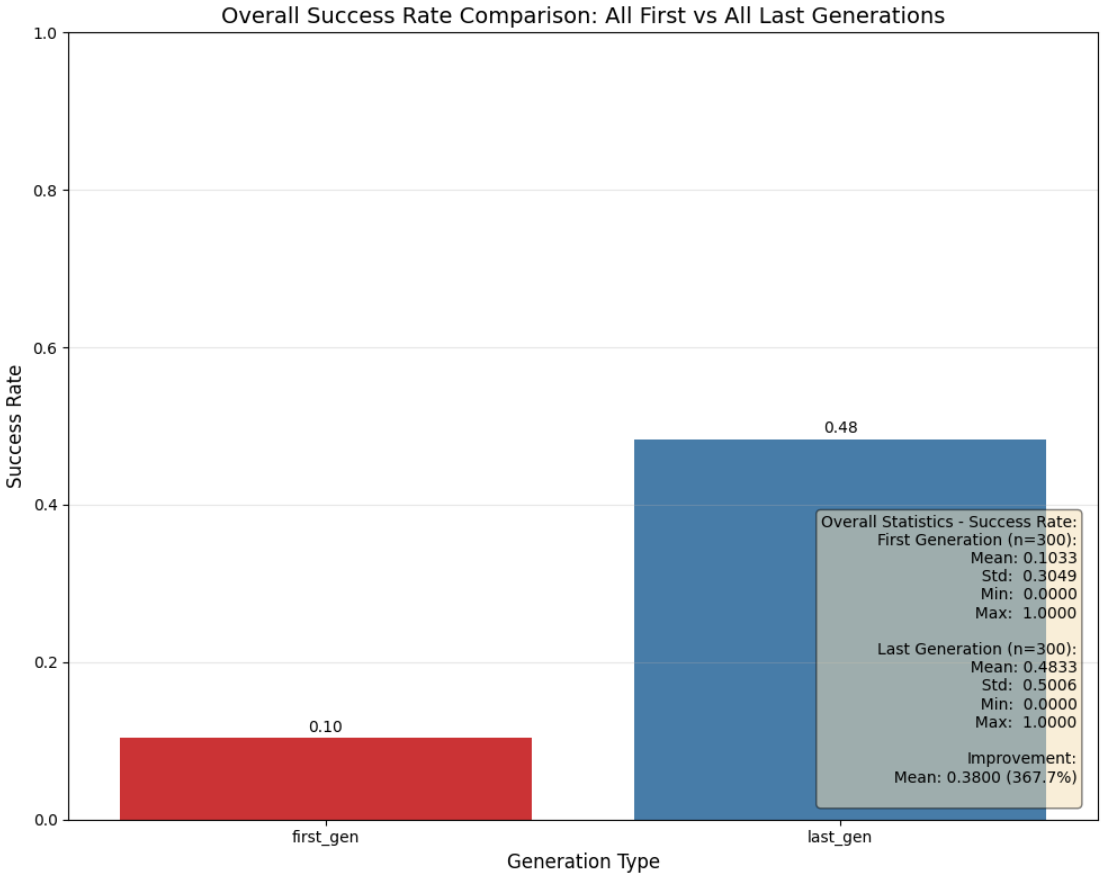

Figure 4.15: Bar graph showing the overall success rate difference between the first and last generations. The evolutionary process significantly increased the global survival probability.

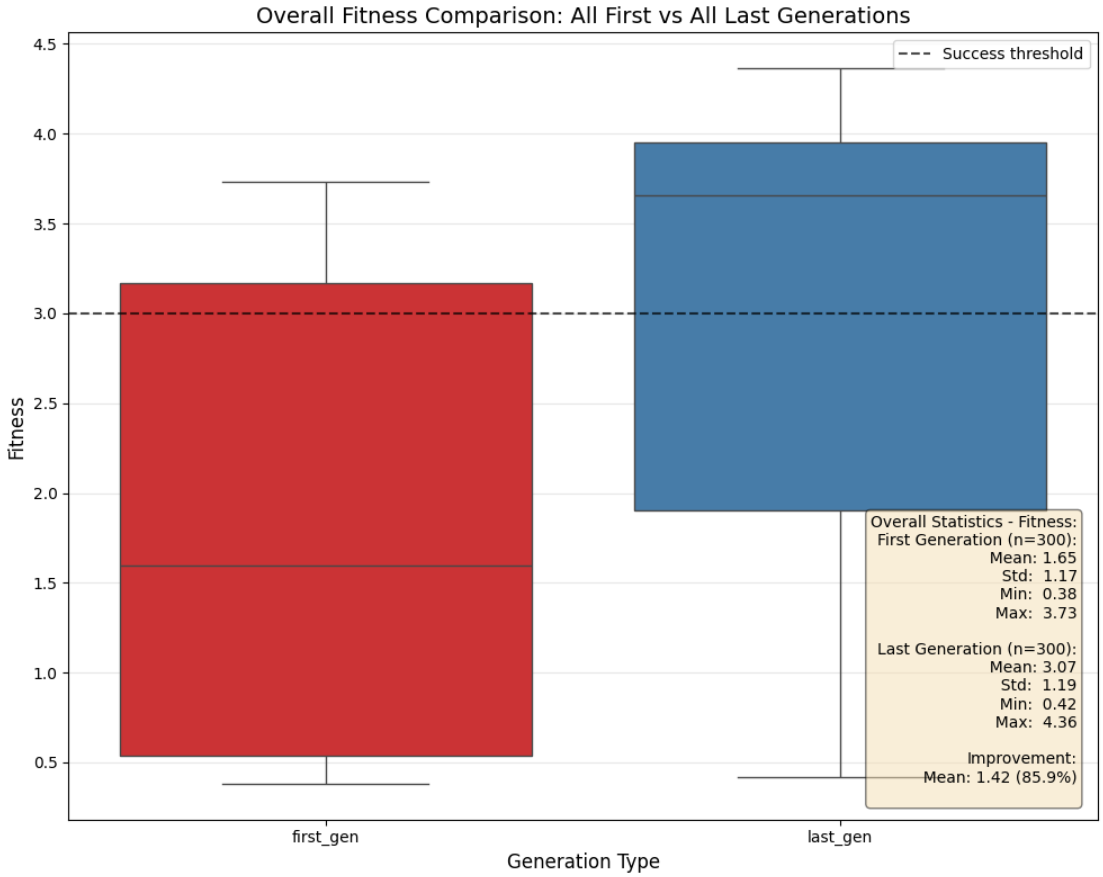

Figure 4.16: Box plot showing the overall fitness distribution between the first and last generations. The final generation exhibits a higher median and a more compressed interquartile range.

To rigorously determine whether these observed improvements were the direct result of the optimization process rather than random sampling variance, formal statistical hypothesis testing was conducted with a designated significance level of $\alpha = 0.05$. Given that fitness distributions in reinforcement learning tasks often deviate from strict normality, both a parametric Student's independent t-test and a non-parametric Wilcoxon test were applied to ensure analytical robustness.

The statistical results are definitive. For the Success Rate, both the t-test ($t = -11.2297$) and the Wilcoxon test ($U = 27900.0$) yielded p-values effectively equal to zero ($p < 0.001$). Crucially, unlike Scenario A, the global Fitness improvement in Scenario B was also found to be highly **statistically significant** across both tests ($t = -10.0676$; $U = 3742.0$, $p < 0.001$).

This dual statistical significance marks a fundamental distinction from the purely spatial task. In Scenario B, fortuitous initialization is mathematically insufficient to sustain long-term survival due to the active context penalties. The strictly significant p-values prove unequivocally that the Genetic Algorithm did not merely fine-tune existing spatial mappings; it actively and necessarily constructed the dynamic exploration-exploitation switching mechanisms required to survive, structurally transforming failing networks into highly adaptive contextual agents.

Behaviour Analysis To fully understand the generalization and online adaptation capabilities of the last generation of evolved controllers in Scenario B, it is necessary to analyze their specific capture behaviors during the testing phase. The statistical distributions of the targeted colors (Yellow, Red, and Green) across 30 unseen random seeds provide direct insight into how the final elite agents manage the exploration-exploitation dilemma when contextual reward contingencies suddenly shift.

In the initial phase, Environment 1 (`TYPE1`), where only Yellow Colorbots provide an energy reward, the controllers demonstrate near-perfect contextual discrimination. As illustrated in Figure 4.17, the exploitation of the correct target is highly deterministic: Yellow captures dominate with a mean of 123.39 and a tightly clustered median of 124. Conversely, incorrect targets are successfully ignored, with Red and Green captures averaging merely 5.54 e 5.46, respectively. This confirms that the initial learning phase is highly efficient, allowing the agent

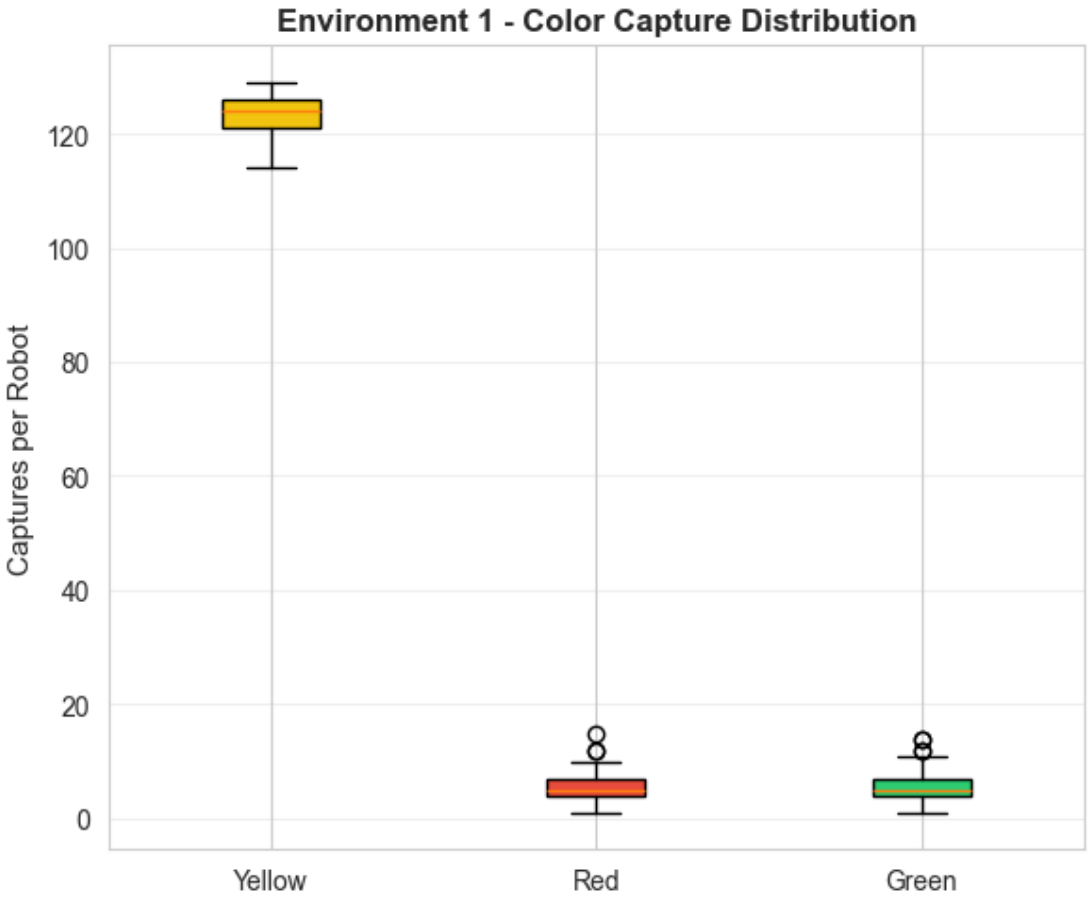

Figure 4.17: Box plot showing the choices counters of the colours that the robot captured in the environment 1, where Yellow is the only Colorbot with reward.

to rapidly map the visual input to the correct behavioural head.

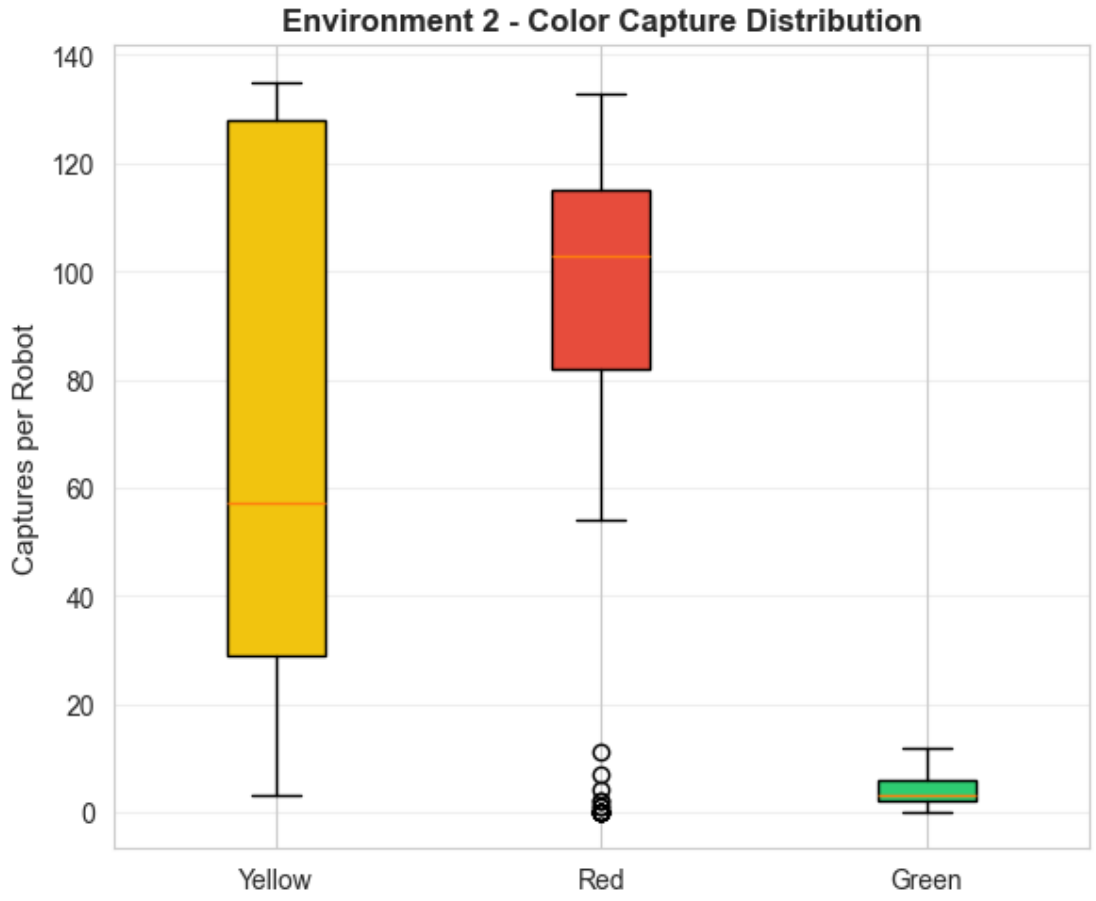

Figure 4.18: Box plot showing the choices counters of the colours that the robot captured in the environment 2, where Red is the only Colorbot with reward.

The true test of online adaptation occurs during the transition to Environment 2 (`TYPE2`), where the rule silently changes and Red becomes the sole rewarding colour. Figure 4.18 reveals the behavioural dynamics of this transition. The agent successfully pivots its strategy, with Red captures surging to become the dominant choice (mean = 90.93, median = 103). However, the data also highlights the expected "unlearning" overhead: Yellow captures drop significantly but maintain a notable presence (mean = 70.81). This variance perfectly illustrates the dynamic Soft-max exploration mechanism in action. When Yellow captures suddenly yield

penalties, the agent's energy drops, triggering an explosive increase in behavioural stochasticity. The robot actively samples the environment (capturing both Yellow and Red) until the Critic's TD-error solidifies the new Red mapping. Crucially, Green targets, which were never rewarded, remain ignored (mean = 4.02), proving the multi-headed architecture prevents random, total policy collapse.

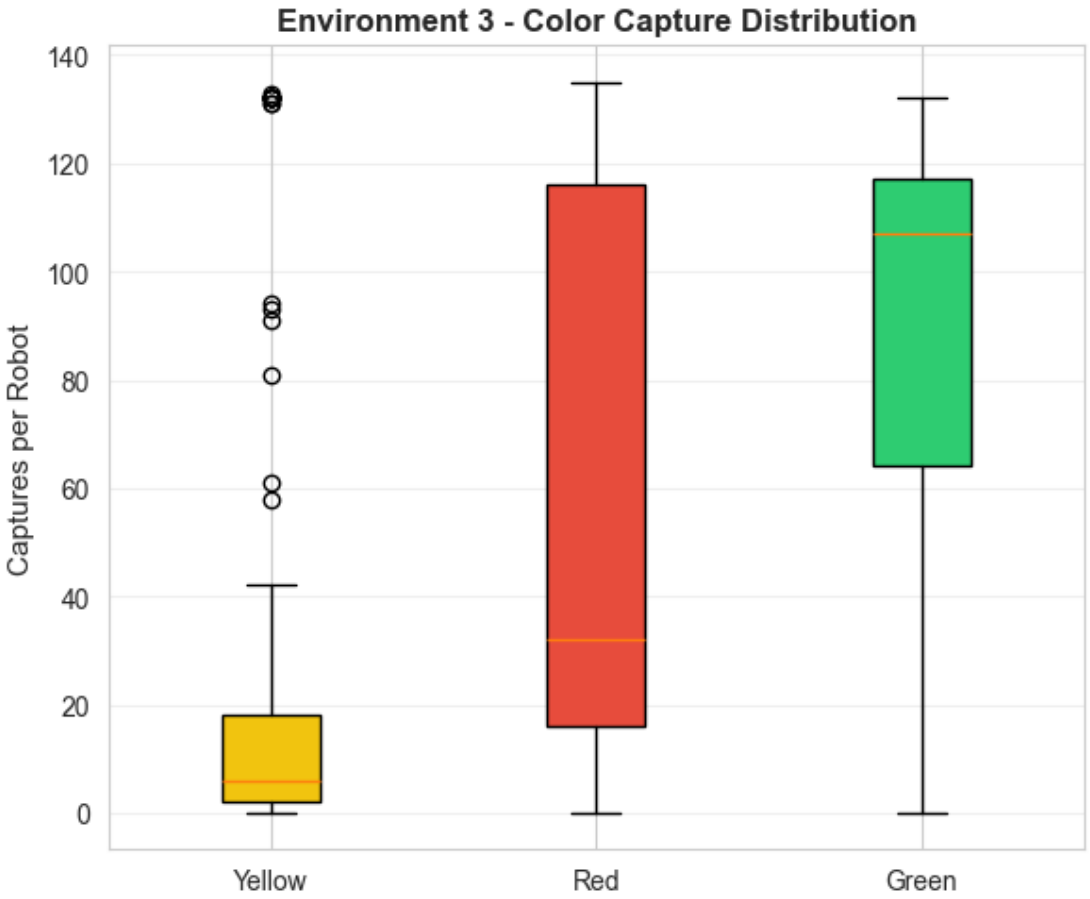

Figure 4.19: Box plot showing the choices counters of the colours that the robot captured in the environment 3, where Green is the only Colorbot with reward.

The final phase, Environment 3 (`TYPE3`), further validates the robustness of the cognitive switching. As the reward shifts to Green targets, Figure 4.19 shows the agent successfully identifying the new rule, with Green captures taking the lead (mean = 85.00, median = 107). The previously rewarded Red targets experience a

sharp decline (mean = 54.55), and the initially rewarded Yellow targets are almost entirely abandoned (median = 6).

Overall, these behavioural statistics definitively prove that the evolutionary process successfully synthesized the expected adaptive logic. The agents do not simply memorize a single static trajectory; instead, they utilize the Critic's error signal and the energy-dependent exploration thresholds to dynamically rewire their contextual preferences on the fly, successfully navigating serial rule changes without falling victim to catastrophic forgetting.

It is important to note, however, that these results depend on the length of the execution phase. The adaptation we see is a snapshot of a process that is still happening; it is likely that with more time, the shift toward the new color would be even more distinct, as the agents would have a longer opportunity to settle into their new behavior.

Chapter 5

Conclusion and Future Work

The primary objective of this thesis was to address the inherent limitations of static, pre-trained control models when deployed in non-stationary robotic environments. Real-world applications demand autonomous agents capable of continuously adapting their decision-making policies to unpredictable shifts in resource availability, environmental constraints, and task rules. To solve this, this work investigated the hybridization of Evolutionary Robotics and the Actor-Critic RL paradigm. By utilizing a Genetic Algorithm to optimize the learning meta-parameters and the initial network topologies, the robots were endowed with a highly efficient "innate" predisposition for lifetime learning.

The validity of this Evolutionary RL approach was demonstrated through two distinct, highly dynamic experimental scenarios evaluated within the ARGoS3 simulator.

In **Scenario A**, the agent was tasked with surviving in an environment characterized by fluctuating energy resources. The results demonstrated that dynamically modulating the Soft-max exploration temperature (β) based on the robot's internal battery level is a highly robust mechanism for managing the exploration-exploitation dilemma. Instead of executing a rigid routine, the evolved controllers successfully learned to exploit known resources when energy was sufficient, and intelligently transitioned into extreme conservation strategies (extended waiting periods) or explosive stochastic exploration when survival was threatened by resource scarcity.

In **Scenario B**, the research tackled the profound challenge of *catastrophic forgetting* in a multi-colour contextual reward task. As the environmental rules silently shifted, changing which coloured targets provided energy and which caused penalties, standard shared-weight architectures proved vulnerable to policy collapse. The introduction of a specialized, multi-headed linear architecture successfully mitigated this issue. The behavioural analysis proved that the evolved agents could actively use the Critic's Temporal Difference error to rapidly "unlearn" obsolete mappings and rewire their contextual preferences on the fly, accurately discriminating between visual targets without destroying their fundamental motor capabilities.

Ultimately, this thesis confirms that integrating evolutionary optimization with continuous, error-driven RL provides a powerful, bio-inspired framework for synthesizing resilient, fully autonomous robotic systems capable of surviving and adapting to the profound complexities of non-stationary tasks.

Future Work

While the experimental results achieved in this thesis are highly promising, the current framework relies on controlled simulations. Transitioning these adaptive models to broader applications presents several exciting avenues for future research and improvement:

- **Testing in Noisy Environments:** The current ARGoS3 simulations assume relatively clean sensory inputs. Real-world sensors (such as cameras and proximity detectors) are inherently affected by Gaussian noise, varying lighting conditions, and partial occlusions. Future work should evaluate the robustness of the multi-headed Actor-Critic architecture by systematically injecting high levels of sensory noise during both the evolutionary training phase and the online lifetime adaptation phase.
- **Adaptation in Error-Prone Environments:** Beyond sensory noise, physical robots frequently suffer from mechanical degradation, actuator misalignment, or sudden hardware failures. Future iterations of this research could test the agent's ability to autonomously compensate for localized hardware

malfunctions. Observing whether the RL Critic can detect the performance drop caused by a faulty wheel and subsequently force the Actor to learn an asymmetric driving policy would represent a significant step toward extreme fault-tolerance.

- **Scaling to Highly Complex Environments:** The state spaces used in this thesis, while sufficient for demonstrating the core adaptive principles, are relatively low-dimensional. Future developments should expand the input space to include more complex scenarios. Additionally, the non-stationary tasks could be scaled up to include moving targets, dynamic obstacles, and procedurally generated mazes to further stress-test the limits of the exploration mechanisms.

- **Multi-Agent and Swarm Dynamics:** Finally, the current implementation focuses on a single autonomous agent. Expanding this framework into a multi-robot swarm scenario would open up complex dynamics regarding cooperative vs. competitive learning. Investigating how individual agents adapt their policies not only in response to the environment, but also in response to the emergent, non-stationary behaviours of other adapting agents, represents a thrilling frontier for Evolutionary RL.

In writing this dissertation, generative artificial intelligence tools (Gemini 1.5 Pro, released by Google in 2024) were used for support in content creation and to improve the clarity of the text.

Bibliography

[Cap10] Genci Capi. Online robot strategy adaptation by learning and evolution. *Journal of Intelligent Systems*, 19(1):1–16, 2010.

[CD06] Genci Capi and Kenji Doya. Application of evolutionary computation for efficient reinforcement learning. *Applied Artificial Intelligence*, 20(1):35–55, 2006.

[GBLB12] Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, 2012.

[KT02] Vijay R. Konda and John N. Tsitsiklis. *Actor-critic algorithms*. Massachusetts Institute of Technology, 1 2002.

[NF00] Stefano Nolfi and Dario Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 11 2000.

[PTO+12] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, and Marco Dorigo. Argos: a modular, parallel, multi-physics simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[SB05] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 16(1):285–286, 2005.

Acknowledgements

ITA

Dopo questi anni di grandi sforzi ed esperienze spettacolari, ci tengo a ringraziare le tantissime persone che mi hanno supportato e che hanno reso possibile il raggiungimento di questo importante traguardo della mia vita.

Un ringraziamento speciale va alla mia famiglia: a mia mamma, a mio babbo e a Barbara, per aver reso possibile questo traguardo con i vostri sforzi. Sapere che ogni giorno avete lavorato duramente, supportandomi e credendo in me, è stata la motivazione di cui avevo bisogno per rialzarmi e andare avanti anche nei momenti più difficili e pesanti. Se oggi concludo questo percorso, è solo grazie all'immensa fiducia che avete riposto in me. Ve ne sarò sempre profondamente grato. *Vi voglio tantissimissimo bene!*

A mio fratellino Felipe: grazie per esserci stato sempre, anche quando non riuscivo a farmi sentire perché troppo preso dalla tesi. Nonostante la distanza di mezzo mondo, hai avuto la pazienza di aspettarmi e di continuare a supportarmi con le tue parole, dandomi una forza incredibile in questa avventura.

Desidero inoltre ringraziare ogni singolo membro della mia famiglia. Ognuno di voi, con la propria presenza e il proprio affetto, ha saputo apportare quel fondamentale 'granello di sabbia' che, giorno dopo giorno, mi ha permesso di costruire le basi e completare l'opera di questo mio importante obiettivo.

Un ringraziamento speciale va ad Andrea Roli, mio supervisore e persona fondamentale per la realizzazione di questa tesi. Grazie per il supporto, la disponibilità e per le dritte preziose che mi hanno guidato in questo percorso. È stato un onore imparare dalle sue lezioni. Sono orgoglioso del lavoro svolto e, soprattutto,

di essermi divertito tantissimo nel portarlo a termine: non potevo chiedere una conclusione migliore per la mia carriera universitaria.

Alla mia fidanzata, Sara: grazie per il supporto e la presenza costante che mi hai donato durante tutto questo percorso. La motivazione e il sostegno morale che mi hai trasmesso ogni giorno sono stati essenziali per aiutarmi a concludere questo progetto. Sono davvero felice (*E molto fortunato heheh*) di avere avuto una persona così spettacolare al mio fianco in questa fase della mia vita.

Ringrazio profondamente anche la famiglia della mia fidanzata, per il supporto costante ricevuto sotto ogni punto di vista. Grazie per la vostra presenza e per la spinta che mi avete dato durante tutto questo percorso; il vostro incoraggiamento è stato fondamentale.

A tutti i miei amici dell'università, che sono diventati ormai compagni di vita. Grazie a Borgo, Detu e Cri per gli spettacolari momenti vissuti insieme e per essere stati presenti in ogni situazione in cui ho avuto bisogno di una mano o di un amico con cui parlare. Grazie a Betta e Vale per tutte le conversazioni e i bellissimi momenti di compagnia in cui ho potuto distrarmi, sbizzarrirmi e parlare di qualsiasi argomento in questi ultimi anni. Senza di voi questo percorso non sarebbe stato lo stesso. Un grazie a Luca ed Edo per i nostri scambi: quei pochi momenti in cui ci beccavamo sono stati densi di conversazioni meme, fighe e profonde. Grazie a Salvo e Giulio: nonostante la distanza, siete sempre stati presenti, pronti a coinvolgermi e a fare tantissime risate. Grazie a Fra e Barto per tutte quelle occasioni nelle quali ho potuto staccare la testa e parlare di qualsiasi cosa con voi, distrarmi e godermi delle belle risate e momenti di calma. Un ringraziamento dovuto a Monta: tra un affare e l'altro, mentre è impegnato a macinare soldi con il suo business, ha trovato miracolosamente il tempo e la pazienza di aiutarmi con infiniti progetti in questa magistrale. Un ringraziamento speciale a Fulvio per le uscite, grigliate, le risate e le infinite conversazioni e grazie anche a Marco, per tutte quelle occasioni di confronto e i momenti di tranquillità in cui ho potuto parlare di qualsiasi cosa, trovando sempre un ascolto sincero e piacevole.

Un ringraziamento va anche ai miei amici più vicini. In particolare ad Ale, per

tutte le serate passate insieme tra discussioni di ogni genere: dai momenti di svago e gioco fino ai confronti sulle questioni più serie e tecniche possibili.

Grazie anche a tutti i compagni che ho conosciuto e con cui ho condiviso pensieri durante questo percorso. Un grazie speciale a Shaikha, per tutti i momenti, le conversazioni e gli incontri al bar dove abbiamo parlato di ogni cosa, e ad Ema, per l'incredibile e costante supporto che mi ha dato lungo tutto il cammino di questa magistrale.

Finalmente, grazie a tutte le persone con le quali ho conversato, ho condiviso una risata o che mi hanno teso la mano in tutta questa avventura spettacolare!

ESP

Un agradecimiento especial también para mis amigos desde el *otro lado del charco*, a Fersito, Majo y Andres que después de tantos años, aún están presentes en mi vida y me soportan a pesar de la distancia, la comunicación y tantas circunstancias. *Tqm fersito! Gracias Majo :3 Mucha' gracia' Illo!*

Muchísimas gracias a Juan, Jorge, Santiago, Sebas y Rafa por todas las veces que me han soportado, hecho reír y por los momentos de conversación en donde pude soñar, sentirme en confianza y tener siempre un espacio especial en donde sé que puedo sentirme en paz y hablar de prácticamente cualquier cosa. En especial, gracias a Juan por todas las veces que siempre ha estado ahí para escucharme y reírse conmigo en cada situación, y por la paciencia inmensa que ha tenido conmigo. *Chancla XD!*

Gracias a Saray por esas veces en las que pude desahogarme y saber que siempre tengo una amiga presente al otro lado del mundo, triunfando y lista para escucharme y soportarme a pesar de las pocas veces que logramos hablar.

Finalmente, gracias a todas las personas que he conocido online, sobre todo a Javier por esas ocasiones en las que, a pesar del paso del tiempo, aún puedo contar

con alguien con quien hablar y distraerme de cualquier cosa sin ser juzgado.