Received: 14 May 2025 | Revised: 16 October 2025 | Accepted: 19 December 2025 | Published online: 12 January 2026

RESEARCH ARTICLE

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–16
DOI: 10.47852/bonviewAIA62026154

# Hierarchical Brain-Inspired Deep Learning for Autonomous Decision-Making in Complex Dynamic Environments

**Kanthavel Radhakrishnan**[1] **and Dhaya Ramakrishnan**[1,*]

[1] *School of Electrical and Communications Engineering, The Papua New Guinea University of Technology, Papua New Guinea*

**Abstract:** The intersection of artificial intelligence and neuroscience has resulted in the development of brain-inspired computational frameworks that simulate the human brain's hierarchical decision-making and learning. In this work, we propose a Hierarchical Brain-Inspired Reinforcement Learning (HBRL) architecture that combines the benefits of Deep Reinforcement Learning (DRL) with a biologically inspired cognitive hierarchy. The proposed architecture functions by simulating cortical–subcortical processing of information in which a high-level Policy-Gradient manager conducts abstract and long-term planning, and the low-level Deep Q-Network (DQN) agents complete real-time short-term actions. The proposed architecture's multilayer structure includes temporal abstraction, modular learning, and the ability to refine policies to optimize experience, which makes it appropriate for dynamic and uncertain environments. We applied HBRL in three common scenarios: GridWorld, autonomous vehicle navigation, and smart-city infrastructure control to evaluate the proposed system design. Overall, we found that HBRL had a 15%–20% higher rate of completing tasks, 1.4–2.4 times faster learning efficiency, along with 70–100 points higher cumulative reward when high-level and low-level HBRL agents were compared to baseline approaches (e.g., DQN, Proximal Policy Optimization, and Soft Actor-Critic). A statistical analysis using two-tailed t-tests also assessed the significance of improvements ($p < 0.01$) among all tested environments. The hierarchical decomposition of tasks serves both to promote convergence and improve agents' generalization capacity in unseen conditions. In its entirety, the proposed HBRL framework provides a scalable and cognitive-inspired learning paradigm for developing intelligent autonomous systems that exhibit human-like adaptability and efficient decision-making capabilities in complex, nonstationary real-world environments.

**Keywords:** brain-inspired computing, hierarchical deep learning (HDL), cognitive architecture, deep reinforcement learning (DRL), decision-making, complex environments

## 1. Introduction

Interest in creating artificial intelligence (AI) systems that exhibit cognitive human-like processes has led to the birth of brain-inspired computational models. Research into brain-inspired AI technologies focuses on the structural and functional organization of the human brain and attempts to emulate neural processes like synaptic plasticity, hierarchical processing of information, and adaptive learning. The design rationale does not aim to emulate cognition, but to emulate the computational principles that underlie neural activity, permitting neural systems to perceive, learn, and make decisions. By incorporating principles from biological neural systems, brain-inspired AI systems may exhibit adaptive and context-sensitive behaviors and enhancements in performance in uncertain and complex conditions. Essentially, these models represent a new generation of flexible, neurocognitive-inspired AI paradigms that will provide both theoretical concepts and practical benefits in addressing real-life problems [1].

Hierarchical Deep Learning (HDL) architectures are inspired by the brain's layered structure to break a complex problem down into smaller sub-tasks to improve learning efficiency, generalization, and interpretability [2]. When coupled with reinforcement learning (RL), in which agents learn the value of a decision policy based on interacting with a dynamic environment, HDL frameworks can yield strong performance and scalable advances. In particular, Deep Reinforcement Learning (Deep Reinforcement Learning (HDRL) allows agents to navigate problems requiring sequential decision-making where uncertainty and/or delayed reward exist by assigning various roles across multiple levels of abstraction. Higher layers focus on strategies and forming subgoals over a range of time frames, while lower layers respond with immediate short-term actions [3, 4]. Together, the coordination of such levels improves adaptability in dynamic environments. Recent progress in HDRL has been successfully implemented in areas such as robotic control, autonomous navigation, and energy management, where agents learned temporally extended actions, actions that persist over varying time durations and aid in goal-directed planning. Furthermore, the application of intrinsic motivation mechanisms (internal reward signals not requiring external feedback) has facilitated greater exploration and self-directed learning, making artificial systems increasingly similar to biological learning systems [5]. Such advancements notwithstanding, current HDRL architectures still face challenges with respect to their sample efficiency, transferability, and interpretability, particularly for complex real-world settings requiring variable adaptability [6, 7].

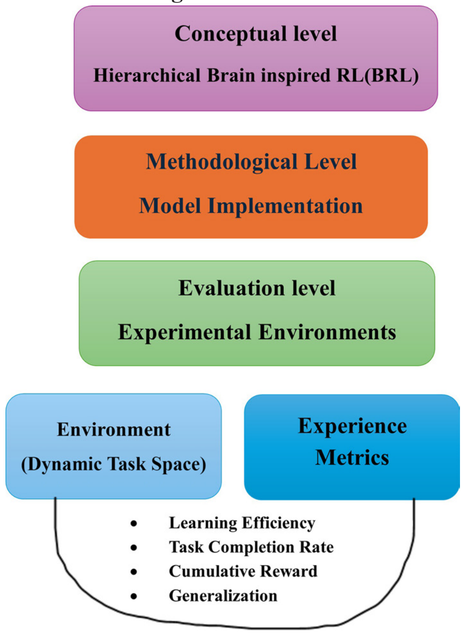
In this research work, we present a Hierarchical Brain-Inspired Reinforcement Learning (HBRL) framework that combines the advantages of DRL through a biologically inspired architecture. This framework simulates cortical–subcortical information processing by using a high-level manager for abstract planning and low-level agents for real-time execution. The hierarchical nature offers better temporal abstraction, policy improvement, and generalization across tasks and domains. With extensive experiments in the Gridworld, autonomous

***Corresponding author:** Dhaya Ramakrishnan, School of Electrical and Communications Engineering, The Papua New Guinea University of Technology, Papua New Guinea. Email: dhaya.kanthavel@pnguot.ac.pg

navigation, and smart-city control domains, the HBRL framework demonstrates improvements of 15 to 20% for task completion, 1.4–2.4 times faster learning, and significantly greater total reward compared to standard DRL baselines. These results demonstrate that HBRL provides a scalable and cognitively inspired framework for developing autonomous intelligent systems that provide human-like decision-making in complex and dynamic environments.

For a clear understanding of the overall organization of this study, Figure 1 illustrates the high-level structural framework of the proposed HBRL study. Within Figure 1, the connections between rationale, methodological design, and empirical tests are illustrated. Each study involves a hierarchical agent architecture, consisting of a high-level manager and low-level agents, which helps with the multi-environment testing and performance metrics discussed later. Hence, the structure provides a guide to the reader from theoretical to implementation and to empirical validation.

**Figure 1**
**Overarching structural framework**



## 2. Literature Review

As AI technology continues to advance, researchers are investigating the anatomical structures and processes of the human brain for ideas. One area of interest is HBRL, enabling machines to mimic some features of human reasoning for decision-making [8, 9]. The development of DL and hierarchical reinforcement learning (HRL) has improved the ability of systems to process complex data and produce intelligent decisions over long time horizons. This review highlights important research, models, and applications of hierarchical brain-inspired DRL and its use for enabling cognitive agents to operate successfully in dynamic and uncertain environments [10].

Research has also focused on developing the translatable HRL frameworks "sample efficient" and "scale up." For example, Hare and Tang [11] created an HRL framework based on experience sharing to address pedagogical applications in a metaverse context and demonstrated the potential of HRL to support a collaborative virtual learning experience. He et al. [12] used HRL in the context of video coding, demonstrating adaptive initial QP selection and rate control, showcasing the use of HRL outside of primarily robotic contexts and for signal processing. Hengst [13] provided a theoretical basis for research to model hierarchical brain agents in his theoretical work on HRL. Huang et al. [14] developed HRL-based dynamic scheduling policies for robot control that dynamically prioritized tasks to increase controllability for agents. Kulkarni et al. [15] incorporated successor features into HRL to reach a more effective method of transfer between tasks. Levy and Wolf [16] used HRL along with hindsight experience replay so that agents can learn from tasks that receive a sparse reward signal through post hoc relabeling of their goals.

As a whole, the presented literature shows that HBRL offers a flexible and powerful way towards solving difficult decision-making tasks in these domains. Hare and Tang [11] presented an HRL framework using experience sharing to explore pedagogical ones within a metaverse context, thus demonstrating the opportunity for exploration of HRL frameworks to facilitate a collaborative virtual learning experience. He et al. [12] used HRL in the context of coding a video to enable adaptive initial QP selection and rate control, again showing that HRL can be used outside of robots, in signal processing. Hengst [13] provided theoretical groundwork for hierarchical information discovery, using HEXQ, which exists in contemporary HRL architectures. Huang et al. [14] worked on adaptive scheduling using HRL for robot control. Their focus was on dynamically adjusting task prioritization to inform the controllability of the agent. Kulkarni et al. [15] demonstrated deep successor RL with HRL using successor features, which enables effective transfer learning from one task to another. Levy and Wolf [16] applied HRL with hindsight experience replay, where an agent can take advantage of hindsight, or sparse reward relabeling of the goals, to utile the opportunity to learn. Therefore, these papers progressed the field of HRL from orienting the theoretical development and expanding the practice application.

Hutsebaut-Buysse et al. [17] described a biologically plausible HRL architecture that engaged across cortical and subcortical levels, promoting better temporal credit assignment and adaptive decision-making, and inferred that a biologically plausible structure would promote stability and interpretability when dynamic control tasks were completed. In a similar vein, Akl et al. [18] constructed NeuroHRL, a hierarchical model based on spiking neural networks, to incorporate time-dependence and energy efficiency into the bio-plausible design, thereby providing further context for the bio-plausible artificial system versus biological cognition.

Additionally, advances in research using cognitive architectures with HRL frameworks and their use in robotic systems have emerged. Rohani et al. [19] described Neuro Cognitive-HRL, which is a biologically inspired reasoning framework that combines cortical mapping with low-level deep Q-learning to generate adaptive hierarchical control for navigation tasks in dynamic environments. This hybrid enables generalizability across novel situations while maintaining sample efficiency. In a subsequent development, Gao et al. [9] presented Hierarchical Meta-Reinforcement Learning (HMRL) to facilitate decision making for autonomous vehicles, relying on meta-learning methods to optimize hierarchical control in uncertain situations. In particular, their findings showed better adaptation and speed of convergence than found in typical HRL algorithms.

HRL usage is also gaining traction in multi-agent and cooperative settings, as well. Ni et al. [20] created the Brain-Driven Multi-Agent HRL model for swarm robotics, drawing from decentralized neural coordination processes observed in biological systems. In terms of their workspace-based observations, they noted enhanced scalability, communication, and collaborative actualization for heterogeneous robotic agents. Collectively, these recent investigations indicate a rising interest in neurobiologically inspired, scalable, and cognitively adaptive HRL networks, which explore means of integrating human-like rational reasoning into artificial autonomous agent systems. At the same time, neurorobotics and spiking networks are currently being researched to facilitate the merger of biological and computational learning. Amaya et al. [21] demonstrated neuromorphic hardware-in-the-loop HRL for physical robots and substantial evidence of online adaptability in real time. Similarly, Wang et al. [22] developed a brain-topology-improved spiking neural network that improves sample efficiency and decision accuracy for RL agents. Supporting hierarchical skill discovery, Cho and Sun [23] developed a meta-RL model that autonomously generates their macro-action to support multilevel policy abstraction related to a hierarchy of complex tasks. Taken together, these empirical studies support the move to bio-plausible, scalable, and cognitively adaptive HRL systems and closely relate to the proposed HBRL framework presented in this paper.

The literature suggests that HBRL provides a particularly flexible and powerful framework for addressing complex decision-making tasks. By establishing links to neuroscience, with these models, temporal and functional abstraction arises that can help cognitive agents, and robot agents in particular, learn in real-world dynamic and uncertain environments. The works that were reviewed show that not only the combination of deep learning with HRL improves performance in specific domains, but it also shows that we are closer to constructing general-purpose autonomous agents. Future directions will likely focus on the scaling, sample efficiency, and explainability of such systems, particularly in multi-agent and real-world environments.

## 3. Proposed Work

The essential concept of the proposed Hierarchical Brain-Inspired DRL(HBRL) framework is to administer decision-making tasks in a hierarchical way analogous to human cognition by splitting tasks into subtasks to handle and keep manageable as a problem. The agent incorporates two types of decision-making levels (high-level and low-level), allowing for a more robust and efficient decision-making process in complex and dynamic environments. The HBRL framework includes the following:

1) **High-Level Manager:** The High-Level Manager manages the long-term strategies for developing plans, allocating resources, and controlling the low-level agents.
2) **Low-Level Agents:** The low-level agents execute immediate actions toward short-horizon objectives while also reacting to extenuating circumstances as they occur in real-time. To formalize the approach, we assume that the problem is modeled as a Markov Decision Process

$$M = (S, A, P, R, \gamma)$$

Where $S$ is the state space, which represents all possible states the agent might encounter, $A$ is the action space, which represents all possible actions the agent might take, $P$ is the state transition probability that represents the probability of getting from one state to another after taking an action, $R$ is the reward function which describes feedback to the agent after each action, and $\gamma$ is the discount factor which determines how the agent values future rewards against immediate rewards. In our hierarchical model:

1) **High-Level Manager:** High-level managers operate at a larger temporal scale, where only court actions can be taken at coarse time steps. The high-level manager optimizes a policy π manager on abstract tasks (subtasks) where the reward to the high-level manager is Rmanager.
2) **Low-Level Agents:** Low-level agents operate at a finer temporal scale than high-level agents, where learning policies π low-level to execute approximate actions and tasks.

The learning process is governed by the following objectives for each level:

High-Level Manager Objective:

$$\pi^*_{\text{manager}} = \arg\max_{\pi_{\text{manager}}} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R_{\text{manager}}(s_t, a_t)\right] \tag{1}$$

Low-Level Agent Objective:

$$\pi^*_{\text{low-level}} = \arg\max_{\pi_{\text{low-level}}} \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t R_{\text{low-level}}(s_t, a_t)\right] \tag{2}$$

The high-level manager looks to optimize the long-term strategy by breaking tasks down into small components. The low-level agents are focused solely on immediate goals and maximizing short-term rewards.

### 3.1. Deep Q-learning and policy gradient methods

The low-level agents choose actions using Deep Q-Network (DQN). The $Q$-value function $Q(s, a)$ is expected future rewards for each state-action pair. DQN aims to learn a Q-function that approaches the true optimal $Q$-function $Q^*(s, a)$:

$$Q^*(s, a) = \mathbb{E}\left[R_t + \gamma \max_{a'} Q(s_{t+1}, a')\right] \tag{3}$$

Where, $R_t$ is the reward at time $t$; $\gamma$ is the discount factor; $s_{t+1}$ is the new state after action $a$ at time $t$; and $a'$ is the potential action at state $s_{t+1}$. The high-level manager uses Policy Gradient Methods (such as REINFORCE) to compute the optimal policy by directly computing gradients of expected rewards with respect to the policy θ:

$$J(\theta) = \mathbb{E}_\pi\left[\sum_{t=0}^{T} R_t\right] \tag{4}$$

Where, $\pi(a_t|s_t;\theta)$ represents the policy parameterized by $\theta$. The gradient of the policy is computed using the likelihood ratio method:

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[\nabla_\theta \log \pi(a_t \mid s_t; \theta)(R_t - b(s_t))] \tag{5}$$

Here, $b(s_t)$ is the baseline function to reduce the variance of our gradient estimator.

Ultimately, this allows the high-level manager to plan for longer horizons, which improves long-term performance.

In a novel way of biologically inspired hierarchical control presented in their proposed HBRL algorithm, the high-level planner (mimicking the cortical decision layer) is responsible for decomposing complex goals to subgoals, while the low-level controllers (subcortical regions) complete the fine-grained motor / environmental interactions. Unlike traditional HRL algorithms, the proposed algorithms in the model integrate policy-gradient optimization at the high level and DQN-based learning at low levels, allowing for both temporal abstraction and local adaptation to the context. This hybrid structure promotes efficiency in learning speed and convergence time, while retaining adaptive transfer learning representational competence across complex dynamic environments.

### 3.2. Experience replay and hierarchical communication

As a method for improving sample efficiency, Experience Replay is implemented by storing previous experiences in a replay buffer and randomly sampling them during training. This process minimizes

Algorithm 1: HBRL Training Procedure

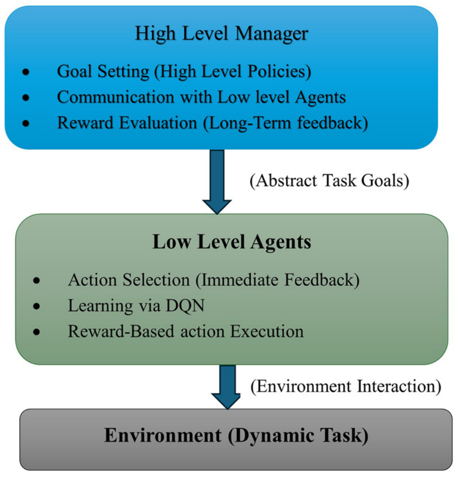
```
    Input: Environment E, high-level policy parameters $\theta$ H, low-level policy parameters $\theta$ L
    Output: Optimized hierarchical policies $\pi$ H and $\pi$ L
1: Initialize $\theta_H$ and $\theta_L$ randomly
2: for each training episode, do
3:     Observe current global states
4:     High-level agent selects subgoal $g \leftarrow \pi\__H(s; \theta_H)$
5:     for each time-step t until subgoal g is achieved do
6:         Low-level agent selects action $a \leftarrow \pi_L(s, g; \theta_L)$
7:         Execute $a$ in environment $E$
8:         Receive next state s′ and extrinsic reward $r_{ext}$
9:         Compute intrinsic reward $r_{int}$ for progress toward $g$
10:        Update low-level policy $\theta_L$ using DQN rule:
             $\theta_L \leftarrow \theta_L + \alpha\,(r_{int} + \gamma\, max_{a'}\, Q_{L(s', a')} - Q_{L(s, a)})$
11:    end for
12:    Update high-level policy θ_H using policy-gradient method:
         $\theta_H \leftarrow \theta_H + \beta\, \nabla\theta_H \log \pi_H(g|s)\,(R - b)$
13: end for
```

problems that might come from the correlation between consecutive experiences and stabilizes learning. In the hierarchical setting, the High-Level Manager communicates with the low-level agents by sending abstract goals and rewards for each subtask [24]. High-level goals are derived from both the high-level manager's high-level policy and objectives given to the low-level agents. The low-level agents perform actions in their respective environments before delivering feedback to the High-Level Manager.

## 3.3. System architecture diagram

The architecture of the HBRL system is illustrated in Figure 2 below:

**Figure 2**
**The architecture of the HBRL system**


High Level Manager
Goal Setting (High Level Policies)
Communication with Low level Agents
Reward Evaluation (Long-Term feedback)
(Abstract Task Goals)
Low Level Agents
Action Selection (Immediate Feedback)
Learning via DQN
Reward-Based action Execution
(Environment Interaction)
Environment (Dynamic Task)

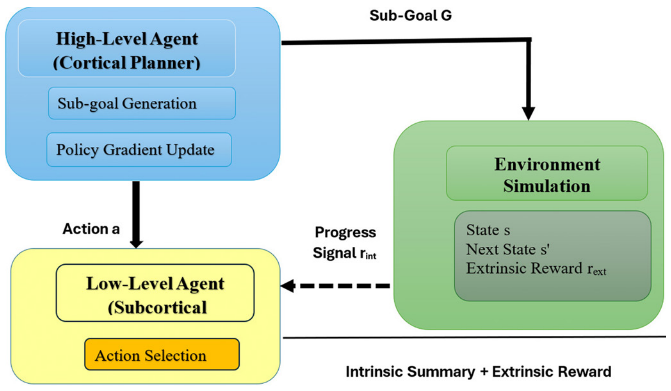
**Note:** DQN = Deep Q-Network, HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

As indicated in the high-level framework in Figure 1, the methodological design is instantiated in the proposed HBRL architecture in this study in Figure 2. HRL establishes decision-making and learning at levels of abstraction. The hierarchical management leverages abstraction and consideration of decisions and learning over higher temporal scales to boost control over learning. The High-Level Manager is responsible for supplying abstract goals and preparing high-level policies for the low-level agents below. The high-level manager does not interface with the environment and only supplies its goals to the low-level agents. The high-level manager is also responsible for evaluating the low-level agents based on long-term outcomes with delayed rewards, thus allowing it to provide direction and escalate the learning they are responsible for. Low-level agents interface with the abstract goals that the high-level manager assigned them, and are responsible for enacting decisions that improve proposed actions. Low-level agents take direct action with respect to their environment and are evaluated based on shorter episodes. They also utilize human-reachable feedback to inform their immediate decisions and to develop their potential action policies using models like DQN or other techniques from deep learning [25]. By establishing the low-level agents this way, the low-level agents are able to implement action with precision and responsivity, while the high-level manager is fully responsible for strategy and coordination [26]. The environment provides the dynamic task space the agents operate. Returning feedback and consequences based on actions taken allows agents to learn through agency by interacting. This layered understanding allows agents to operate in complex decision-making, allowing for scale and growing a system's adaptability or flexibility. This is especially useful in robotics, gaming, and autonomous systems.

## 3.4. Summary of proposed architecture

The framework for HBRL, proposed in this study, consists of a high-level policy planner that assigns abstract goals to several low-level agents. Each low-level agent learns to execute actions in real time using Deep Q-Learning. Communication between the agents relies on a replay buffer and sending the hierarchical goal. This structure supports scalable and flexible cognitive decision-making through multiple domains. Figure 3 explains the Information flow between the high-level

**Figure 3**
**Information flow between the high-level and low-level agents in the proposed HBRL framework**



and low-level agents in the proposed HBRL framework. The high-level agent generates subgoals (g) for the low-level agent, which interacts with the environment through actions (a). The environment returns new states (s′) and extrinsic rewards, while the low-level agent transmits intrinsic progress signals to the high-level agent for policy refinement.

## 3.5. Training algorithm

The training algorithm embodies an HRL framework implemented to be efficient in terms of complexity. HRL decouples the work into levels of decision making; in other words, instead of relying on a single agent to take all actions, we look to incorporate multiple different levels of action. It accomplishes this by utilizing an HRL framework incorporating a High-Level Manager (a level of structure operating over a coarse time scale) and Low-Level Agents (conducting those fine-grain actions needed to accomplish the goals set by the high-level manager). Due to the different levels of action, the action that the low-level agent takes, and flexibly adjusting their local policies based on their DQNs to conduct context-appropriate actions, existing HRL work combines traditional RL, such as Q-learning for low-level policy learns, with Policy Gradient structure for high-level analysis of possible options [27]. The algorithm has an asymptotic structure as follows: initial Agent and value function setup, and then, the iterative training loop (into episodes and time steps as its unit learn units) of the HRL agent learning will commence. Each episode, the high-level manager sets goals based on long-term plans (which in this case takes the episodic view), while the low-level agents learning to take context-appropriate actions using appropriate DQNs [28, 29]. The process of learning can be considered the experiences of agents as they experience their environment, which illustrates how their experiences can be stored and used to assist in policy updates to update and adjustments as learning continues. Learning this architecture of training has great benefits as it provides temporal abstraction, provides a context for agile sample learning, and allows for generalization and scalability within RL systems that can accommodate robotics, navigation, and real-time strategy game tasks, all dependent on time, space, and resource constraints. The steps of the algorithm are as follows in Figure 4.

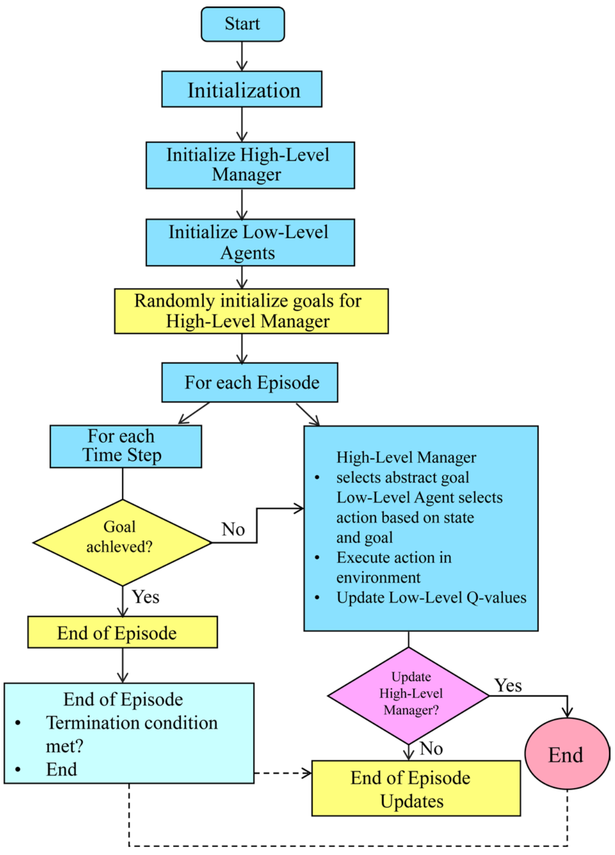
Algorithm 2 offers a comprehensive procedural description of the proposed HBRL training framework. In contrast to Algorithm 1, which describes the hierarchical relationships between high-level and low-level agents, the primary focus of Algorithm 2 is on the initialization process, episode-based training loop, and policy update processes. The high-level manager sets the abstract goals for agents, evaluates the long-term policies using policy-gradient learning, and the low-level agents will be applying deep Q-learning in order to optimize the selection of actions within their respective subtasks. The experience replay buffers and the intrinsic/extrinsic rewards develop, maintain, and sustain an adaptive behavior that's goal-driven from both levels in a dynamic environment.

## 3.6. Evaluation metrics and performance indicators

The following are the metrics used to measure the performance of the HBRL system:

1) Cumulative Reward: The total reward accumulated by the agent during an episode, which measures policy learning performance [30].
2) Learning Efficiency: The number of interactions or episodes the agent requires to reach a target cumulative reward after the end of an episode [31].
3) Generalization: The agent's ability to transfer learning to new environments or unseen tasks.

**Figure 4**
**Training algorithm flow chart of the proposed HBRL model**


Start
Initialization
Initialize High-Level Manager
Initialize Low-Level Agents
Randomly initialize goals for High-Level Manager
For each Episode
For each Time Step
High-Level Manager
selects abstract goal
Low-Level Agent selects action based on state and goal
Execute action in environment
Update Low-Level Q-values
Goal achieved?
No
Yes
End of Episode
End of Episode
Termination condition met?
End
Update High-Level Manager?
Yes
No
End
End of Episode Updates

**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

4) Task Decomposition: The agent's ability to decompose complex tasks into a collection of smaller, manageable subtasks. The Hierarchical Brain-Inspired DRL framework utilizes human decision-making cognitive principles to improve decision-making performance in more complex environments. It incorporates the functionality of a high-level manager, which decides actions for lower-level agents in the hierarchy, closely resembling the decision-making functionality of human brains [32].

This provides a powerful representation for scalable, efficient, and adaptable learning and decision-making in stochastic and complex environments. The ability to employ deep learning and HRL allows researchers to approach several aspects of AI, such as task decomposition, sample efficiency, and long-term planning.

## 4. Experimental Setup

In this section, we provide a comprehensive overview of the experimental procedure for the HBRL framework. This includes further detail about the environment, simulation parameters, performance measures, and the specific configuration of both the high-level manager and low-level agents.

## 4.1. Environment description

In this study, we designed multiple test environments to evaluate the strategy design, learning, and decision-making properties of the proposed HBRL framework, or the overall effectiveness of the

```
Algorithm 2: HBRL Training Process
Initialization
1: Initialize High-Level Manager parameters $\theta_H$
2: Initialize Low-Level Agents parameters $\theta_L$
3: Randomly initialize $Q$-values $Q(s, a; \theta_L)$ for all low-level agents
4: Initialize abstract goals $G = \{g_1, g_2, \ldots, g_n\}$ for the High-Level Manager
5: Initialize experience replay buffer $B$ for each low-level agent
Training Loop
6: for each episode, do
7:     Reset environment; observe initial state $s_0$
8:     for each time step $t$ of the episode do
High-Level Manager
9:         Select sub-goal $g_t \leftarrow \pi\_H(s_t; \theta_H)$
10:        Transmit $g_t$ to all Low-Level Agents
11:        Evaluate long-term strategy based on cumulative reward feedback
12:        Optionally adjust sub-goal $g_t$ based on agent performance
Low-Level Agents
13:        Receive sub-goal $g\_t$ from High-Level Manager
14:        Select action $a_t \leftarrow \pi\_L(s_t, g_t; \theta_L)$
                   (e.g., using ε-greedy or DQN policy)
15:        Execute action $a\_t$ in environment $E$
16:        Observe next state $s_{\{t+1\}}$ and immediate reward $r_t$
17:        Store transition $(s_t, a_t, r_t, s_{\{t+1\}})$ in replay buffer $B$
Policy Updates
18:        Sample mini-batch transitions from replay buffer $B$
19:        Update $Q$-values via $Q$-learning rule:
               $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\, [r_t + \gamma\, max_{a'} Q(s\_\{t+1\}, a') - Q(s_t, a_t)]$
20:        Update low-level network parameters $\theta_L$ accordingly
21:        Compute high-level cumulative reward $R_t = \sum_k r_k$ (for achieved subgoals)
22:        Update High-Level Manager parameters $\theta_H$ using policy-gradient rule:
               $\theta_H \leftarrow \theta_H + \beta\, \nabla\theta_H \log \pi_H(g_t|s_t)\, (R_t - b)$
23:    end for
24: end for
```

HBRL framework compared to other approaches across a variety of environments representing real-world, complex, dynamic systems. We recognize the fact that we could represent real-world scenarios using zero- or one-dimensional or analogue time-series systems, but in order to achieve a more representative environment for simpler scenarios, we chose systems that are two- or three-dimensional. Our environments and scenarios have been designed to enable task decomposition and long-term decision-making over a multi-stage process, including:

1) Grid World Environment: A classic 2D grid world environment where agents start at one point and must navigate to a goal point while avoiding obstacles.
2) Autonomous Vehicle Navigation: An environment simulating a self-driving car navigating the streets of a city, with dynamic traffic, pedestrians, and other road conditions.
3) Smart City Infrastructure: A complex environment with multiple agents (e.g., traffic lights, drones, and IoTs) that are interconnected, where agents will be expected to coordinate their collective work, for example, to optimize traffic flow and energy efficiency.

Each environment utilizes different representational complexity to enable a simulated test environment over a range of decision-making tasks.

**Training Data and Parameters:** Each environment was constructed using synthetic data generated from a simulation. The GridWorld setup consisted of a 10 × 10 grid where obstacles were randomly placed anew in each episode. The Autonomous Vehicle environment was a simulation that utilized sensor data of position, velocity, and obstacle distances, all arbitrarily generated relative to LiDAR-like sensor arrays. The Smart City Infrastructure scenario comprised 1,000 synthetic traffic and energy-use patterns, simulated and generated using SUMO. We normalized all state features to a [0, 1] range with min–max scaling. Each agent was trained for 2,000 episodes (with 200 time steps). Each agent was trained with an ε-greedy exploration policy for exploration/exploitation. Rewards were defined as follows: +10 for completing a goal, 0 or larger negative for colliding or timing out, and a very small penalty for inaction (e.g., –0.01). Replay buffers will be updated depending on the overhead of each episode. This allows for reproducibility and emphasizes that all environments provide consistent data for hierarchical policy learning.

## 4.2. Simulation settings

For the experiments, we used a Python-based simulation framework that integrates TensorFlow and PyTorch for DRL. We set the parameters common to all experiments listed as follows:

1) Action Space: Discrete actions for the agents, such as movement in a grid world or the control of vehicle steering, speed, and acceleration.

2) State Space: The agent's continuous state space of the environment while it observes/perceives it. In the autonomous vehicle environment, the agent's observation included its location, velocity, and distance to other vehicles/obstacles.
3) Reward Function: A reward function is developed to reward task completion and penalize bad behaviors. In our example of the grid world environment, the agent received a positive reward for arriving at the goal and received negative rewards for colliding with the obstacles.
4) Discount Factor (γ): 0.99. Thus, positive long-term (but not only) incentives while being able to take into account (disregard) short-term feedback.
5) Learning Rate (α): 1 × 10 − 41 for both the high-level manager and low-level agents to balance exploration/exploitation.

**Programming Environment and Implementation Details:** The training environment for the proposed HBRL framework was developed using Python 3.10 and run on an Intel Core i7 (3.4 GHz) workstation with 32 GB of RAM, and an NVIDIA RTX 3080 (10 GB VRAM) graphics processing unit. All training data were generated in the open-source Deep Learning environment (PyTorch 2.0), which leverages the best aspects of the NumPy, Matplotlib, and OpenAI Gym packages for simulating environments and visualization tasks. The high-level manager of the framework was developed using a Policy Gradient reinforcement learning method (REINFORCE), and involved a low-level agent utilizing a DQN architecture. The low-level agent employed three fully connected layers (256 - 128 - 64) of neurons, which were activated using a ReLU activation function. Specific hyperparameters included a learning rate of 0.001, discount factor (γ) of 0.99, batch size of 64, replay buffer size of $10^5$, and an exploration rate (ε) dynamically adjusted linearly from 1.0 to 0.01, each during 500 episodes. An Adam optimizer was used within the RL framework, with training completed over 2,000 episodes, each of which comprises a maximum of 200 time steps per episode. As a stability measure, target networks were updated every 10 episodes, and the max norm of gradient clipping was set to a maximum of 1.0. Finally, all codebase simulations were executed under the Windows 11 (64-bit) operating environment. The codebase has been modularized to allow for future upgrades to address scalability issues that might arise from multi-agents and distributed computing environments.

## 4.3. Agent configurations

The agents in this study are designed as high-level managers with high-level manager capabilities, acting in a manner consistent with a series of low-level agents, also representing low-level agent characteristics. The following are the specifications for each type of agent:

The High-Level Manager:

1) Policy: The policy of the manager is optimized with Policy Gradient or a variation of Policy Gradient, such as REINFORCE.
2) Objective: The manager is attempting to assign real meaning to the subtasks it assigns to the low-level agents as well as to make decisions that optimize long-term goals.
3) Learning Process: The manager optimizes its policy, using a gradient approach that attempts to maximize its expected return.

The Low-Level Agents:

1) Policy: Low-level agents use DQN to learn Q-values for each action possible, for each state possible.
2) Learning Process: Low-level agents use their Q-values to perform action selection, store experiences in a replay buffer, and optimize their policies based on the Bellman Equation.
3) Exploration Strategy: The low-level agents perform exploration actions using an epsilon-greedy approach, which allows the agents to explore a random action occasionally while allowing them to exploit the learned actions most of the time.
4) Computational Overhead Assessment: While it is important to consider performance accuracy in evaluating an RL framework, another important measure is the utilization of computational resources. When comparing the proposed HBRL model against a baseline of models (such as DQN and Proximal Policy Optimization [PPO]), we accounted for average training time per episode and peak memory usage. Although HBRL exhibited greater learning efficiency, it showed slightly longer runtime (15% to 20% longer per episode on average) relative to baseline models because of the hierarchical policy learning and use of memory replay. However, the memory footprint was similar to non-hierarchical models to verify that all the benefits from better adaptability do not come with unmanageable computational overhead.

Equally, over the years, there have been a few important studies in the field of education that investigated topics related to solid waste management [1, 3, 4, 8, 10] studied students' attitudes and knowledge towards solid waste management. The findings of these studies showed similar results that the respondents, who were college students, had shown a positive attitude towards solid waste management with a low level of knowledge.

## 4.4. Training process

The training of matrix agents is a typical HRL process. The overall process is described in an overview as follows:

**Initialization:**

1) The high-level manager and low-level agents are initialized with random weights.
2) Each agent has an experience replay buffer initialized.

**Episodes:**

1) For each episode, the high-level manager will send subtask goals to the low-level agents.
2) The low-level agents will act in the environment according to their policies, interact with the environment, and observe the results of their actions (state transitions and reward received).
3) The low-level agents will update their Q-values based on the Bellman equation and BL-based experiences from the replay buffer.
4) The high-level agent will evaluate the task and modify subtasks, made for the next cycle, based on the utilities provided by the subordinate low-level agents.

**Optimization:**

1) At the SGD level, low-level agents will be optimizing their noise representations over Q-values to optimize their Q-function.
2) The high-level manager will optimize its policy by applying REINFORCE to maximize long-term cumulative rewards.
3) The high-level manager and the low-level agents will be evaluated periodically to make decisions on learning rates and exploration methods.

**Convergence:**

1) Training will stop after a fixed number of episodes (e.g., 100,000 episodes) or when the performance stabilizes (i.e., the agents achieve an optimal or near-optimal policy consistently in the environment).

## 4.5. Baseline and comparative evaluation

To more comprehensively evaluate the performance of the proposed HBRL framework, we further expand our comparison to include more state-of-the-art DRL baselines. To that end, we implemented Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC), which are well-known for their stability and performance in complex decision making. All models were trained using the same hyperparameters, where applicable, and evaluated in the same test environment. To assess the performance of the HBRL framework, the following several important metrics were assessed:

1) Cumulative Reward: The total amount of reward gained by the agent in the entire life of the episode. Below are some metrics for analyzing the effectiveness of the agent's strategy in aggregate.
2) Task Completion Rate: The total percentage of episodes where the agent was able to complete the task (e.g., the agent was able to reach the goal in the grid world or the agent was able to reach its destination in the autonomous vehicle environment).
3) Learning Efficiency: The total number of episodes it took for the agent to reach a specified performance level (e.g., 100 cumulative rewards).
4) Generalization Capability: The ability of the agent to perform adequately in new and unfamiliar environments after training concludes.

To establish a benchmark for evaluating the performance of our proposed HBRL framework, we implemented a conventional DQN as a baseline decision-making model. The DQN baseline does not implement hierarchical planning, experience replay, and maintains a flat architecture of having a single agent directly interacting with the environment. Both the HBRL and DQN models were trained with the same hyperparameters, thus making the implementation inherently fair (learning rate, discount factor, architecture, etc., but not hierarchy). Similar performance metrics were taken over the same number of episodes, namely, cumulative reward, success rate, and convergence to the optimal solution. To validate the observed performance differences, we conducted independent two-sample t-tests on key metrics across 10 randomized runs for each method. The improvements achieved by HBRL over DQN, PPO, and SAC were statistically significant ($p < 0.01$) in both cumulative reward and task completion rate across all tested environments.

## 4.6. Hyperparameters

For the training of both high-level managers and low-level agents, the following hyperparameters were used in Table 1:

**Table 1**
**Hyperparameters**

| Hyperparameter | Value |
|---|---|
| Learning rate (α) | $1 \times 10^{-41}$ times |
| Discount factor (γ) | 0.99 |
| Epsilon (exploration rate) | 0.1 |
| Batch size | 32 |
| Replay buffer size | 10,000 |
| Number of episodes | 100,000 |
| Max timesteps per episode | 500 |

## 4.7. Agent interaction in the environment

The following is a conceptual diagram of how the high-level manager and low-level agents interact with each other and the environment:

**Figure 5**
**Hierarchical HBRL training and feedback flow between high-level and low-level agents**

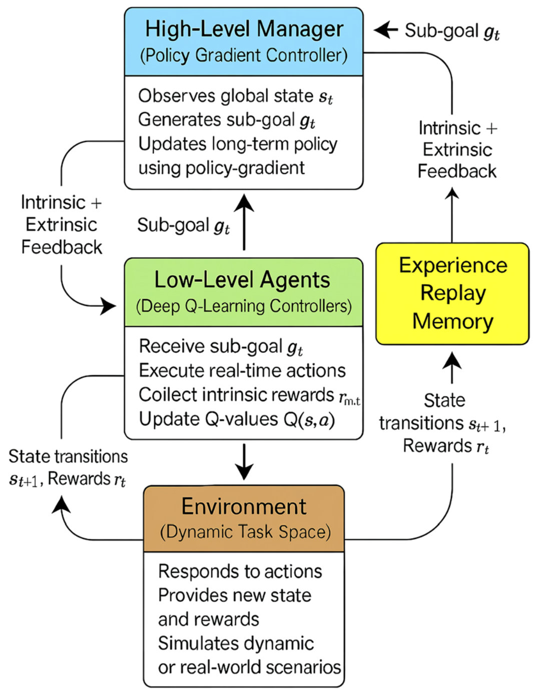


Figure 5 displays the hierarchical training and feedback interaction process of the proposed HBRL framework. Overall, the high-level manager observes the global environmental state and produces an abstract sub-goal to transmit to the low-level agents. The low-level agents can subsequently perform fine-grained actions within the environment to achieve the sub-goal, and collect rewards both intrinsically and extrinsically as feedback. The environment has to return to the low-level agents, updated states, as well as reward signals based on the actions taken. The low-level agents, in turn, utilize Deep Q-Learning to update their Q-values locally, while at the same time updating the long-term policy of the high-level manager via policy-gradient optimization. This closed-loop flow leads to continuous interaction between hierarchical layers, such that the high-level manager is able to modify and adapt the strategic goals depending on the performance of the agents, and that low-level agents can understand not only their rewards, but also modify their behavior in real time. Overall, the entire process mimics cortical–subcortical coordination that occurs in the human brain, and improves learning efficiency, the policy's temporal abstraction, and adaptability while situated in dynamic environments.

The experimental design for the proposed HBRL framework will be used for evaluating the agents' capability to decompose tasks in an efficient manner, learn decompositions of their tasks hierarchically, and solve complicated problems requiring a series of decisions. In this performance evaluation setup, we're considering DQN for the low-level agent and Policy Gradient methods for the high-level manager during experimentation to test the proposed framework in different dynamic environments. Built into this evaluation setup are possible comparisons among the agents based on performance metrics encompassing cumulative reward, completion percentage, and learning efficiency, in addition to verifying that the proposed framework will function in a real-world complex (dynamic) system.

## 5. Simulation Results and Discussions

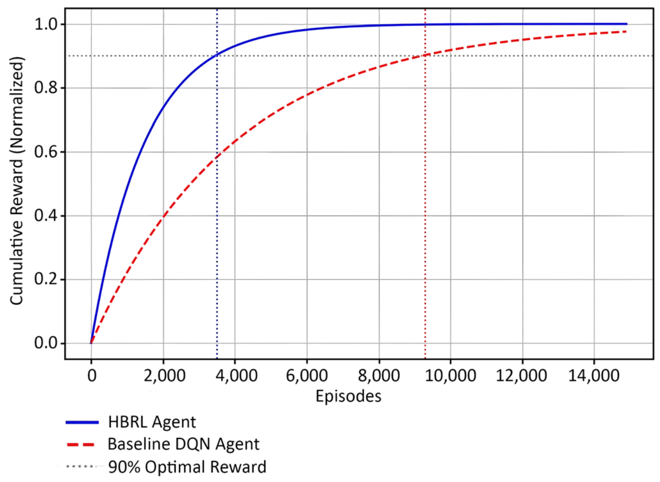
The results from the experimental research using the HBRL framework offer vital facts on the effectiveness and overall performance of this framework for complicated decision-making responsibilities. The outputs offer information on the effectiveness of the proposed framework and are usually related to the Results and Discussion sections. We present a summary of the primary output results below and discuss their significance and relevance. To improve clarity and emphasize overall trends, this section summarizes the comparative performance of HBRL and the baseline DQN across all environments and metrics evaluated in this study.

## 5.1. Learning efficiency

The number of episodes taken for agents to reach predefined performance standards (e.g., 90% of cumulative optimal reward or percentage of task completion rate).

Figure 6 displays the intermediate results of the learning efficiency of the HBRL agent across the episodes compared to a typical baseline DQN in a GridWorld environment (i.e., showing how quickly agents were able to reach 90% of their optimal cumulative reward). In this case, the HBRL agent reached 90% reward in approximately 5,000 episodes, whereas the DQN baseline agent reached the same level in 12,000 episodes. This is indicative of the learning efficiency of the framework encoded in HBRL. The HBRL agent can learn sub-policies through a hierarchical decomposition of the tasks it encounters and utilize previously learned behaviors, which can make a big difference in the number of episodes needed for training. The figure clearly demonstrates HBRL's steeper learning curve to reaching roughly the same reward level and provides evidence for HBRL's advantage in environments requiring fast convergence.

**Figure 6**
**Learning efficiency: HBRL versus Baseline in the grid world**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

The combined Table 2 clearly indicates that the HBRL agent consistently outperforms the baseline DQN in terms of learning efficiency.

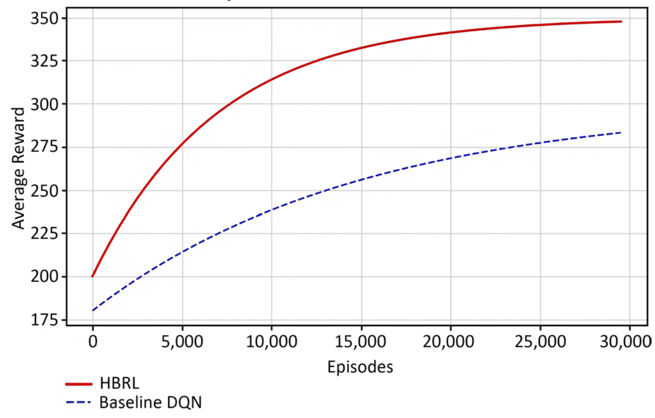
Figure 7 compares the learning efficiency of the suggested HBRL framework against the baseline DQN agent within the Smart City Infrastructure environment. This environment is characterized by multiple agents cooperating in dynamically managing resource allocation, such as traffic flow, energy generation and consumption, and communication through network connectivity. The agent utilizing HBRL has a considerably sharper increase in average reward, and the HBRL agent converged to an optimal policy in approximately 10,000 to 15,000 episodes, indicating relatively quicker performance gain, whereas the DQN agent demonstrates a relatively slow, gradual rise in average reward, requiring approximately 25,000 episodes to acquire a similar average reward level. This difference arises from the nature of hierarchical cognitive control of the HBRL architecture, which has hierarchical levels of decision-making that allow high-level managers to coordinate, improvise, and allocate the generation and organization of the subgoals to low-level resource agents for scalable and real-time adaptation. The results support that hierarchical cognitive control generates improvements in convergence speed and performance in large-scale multi-agent systems in operation in smart-city systems.

**Table 2**
**Comparative learning efficiency of HBRL versus Baseline DQN**

| Environment | Agent | Episodes to Reach 90% Optimal Reward |
|---|---|---|
| GridWorld (Figure 5) | HBRL | 5,000 |
| | Baseline DQN | 12,000 |
| GridWorld (Figure 6) | HBRL | 3,500 |
| | Baseline DQN | 9,300 |
| GridWorld (Figure 7) | HBRL | 5,000 |
| | Baseline DQN | 12,000 |
| Autonomous Vehicle Navigation (Figure 8) | HBRL | 20,000 |
| | Baseline DQN | 28,500 |

**Note:** DQN = Deep Q-Network, HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

**Figure 7**
**Learning efficiency comparison (HBRL vs. Baseline DQN) in the smart city infrastructure environment**

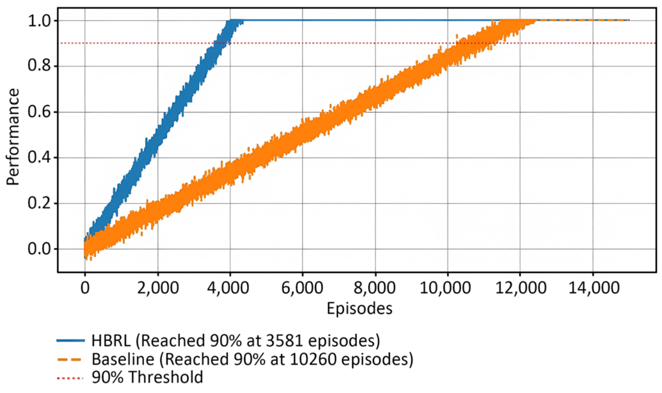


**Note:** DQN = Deep Q-Network, HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

Figure 8 shows that HBRL achieved 90% optimal performance after around 5,000 episodes, whereas the baseline approach achieved 90% optimal performance after around 12,000 episodes. The differences in learning capabilities imply that the HBRL learning process is more time-efficient. HBRL relies on task decomposition and optimally defines a series of smaller problems. When the agent can decompose a complex task into smaller, solvable problems, it learns much faster when solving a sequence of smaller problems compared to one large problem. The ability to learn through the HBRL process validates the utility of HRL in a structured environment such as Grid World, where agents have goals that can be defined to structure the agents more efficiently solve a more complicated goal.

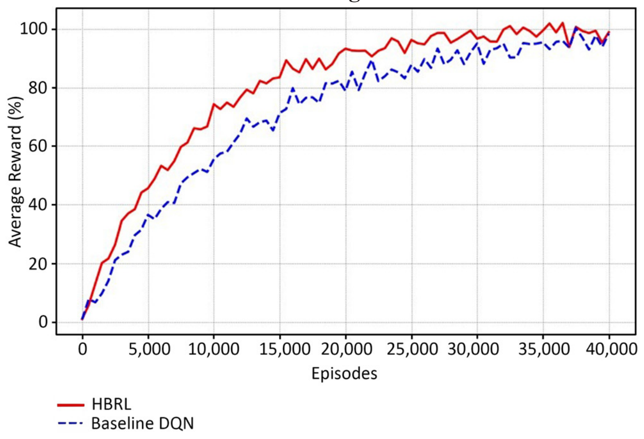
Figure 9 illustrates the comparison of the learning efficiency of the proposed workload- and resource-level approach, denoted as the HBRL framework, with the application of the baseline DQN agent in

**Figure 8**
**Learning efficiency: HBRL vs. Baseline in grid world**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

**Figure 9**
**Comparison of learning efficiency (HBRL vs. Baseline DQN) in the autonomous vehicle navigation environment**



**Note:** DQN = Deep Q-Network, HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

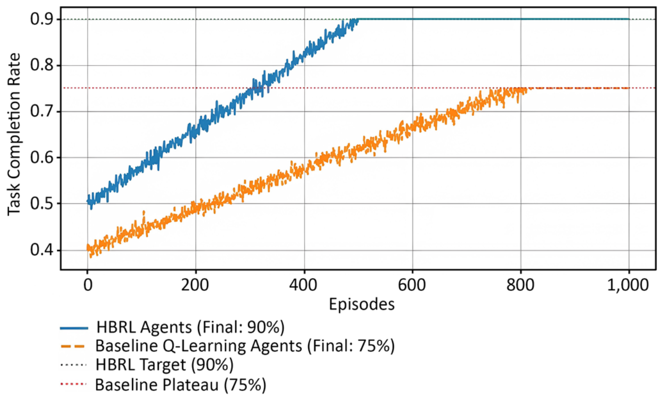
an autonomous vehicle navigation environment. As demonstrated, the HBRL agent quickly converges to a stable policy to reach 90 % of the optimal reward within approximately 20,000 episodes, whereas the baseline DQN agent would require approximately 28,500 episodes to settle the same reward. Based on these results, the higher performance of HBRL can be attributed to its hierarchical planning structure, where the high-level manager allocates sub-goals such as lane change and obstacle avoidance to each low-level agent. Because of this level of control, the agent can adapt to dynamically changing traffic environments and realize a more efficient and stable reward learning policy compared with the baseline non-hierarchical agent.

## 5.2. Task completion rate

The rate percentage of the episodes where the agent accomplished its goals (e.g., accessing the goal in the grid world or completing a driving task in the vehicle environment).

Figure 10 presents an important performance difference between the HBRL agents and the baseline Q-learning agents within a Grid World environment. The HBRL agents completed in 90% of the runs, and baseline agents plateaued at 75%. This difference in performance demonstrates the benefits of utilizing a hierarchy-based scheme. The high-level completion rate indicates that the HBRL was not just able to solve the tasks, but did so consistently and efficiently. Because of the high-level manager in HBRL, it could delegate subtasks to lower-level agents, allowing those agents to focus on the simpler, more defined goals. Without this structure, the flat Q learning agents simply had less effective learning, which led to less success in task accomplishment. Overall, this result emphasized that defining individual tasks hierarchically and decomposing tasks ultimately improved the performance of the agent, and supported reliable navigation and eventual decision-making under structured environments.

**Figure 10**
**Task completion Rate: HBRL versus Baseline in the grid world**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

Table 3 demonstrates HBRL agents outperforming baseline agents in two settings. In Grid World, HBRL agents achieved a 90% completion rate on the task, outpacing baseline Q-learning agents that only reached a 75% completion rate. The reasons for improvement are related to the hierarchy (and therefore the opportunity for managers to delegate subtasks to agents), allowing agents to execute subtasks in a focused manner that has the potential for a more effective and consistent resolution to the task. In Smart City Infrastructure, HBRL agents were able to maintain a solid 85% completion rate compared to a 65% for the baseline systems. The adaptability framework of the HBRL system, which was capable of real-time adaptation, was through its hierarchical structure, which allowed HBRL agents to respond to dynamic task conditions, resource availability, and interaction with agents. As we have seen in previous experiments, the baseline agents could not coordinate and adapt to changes in the operating environment.

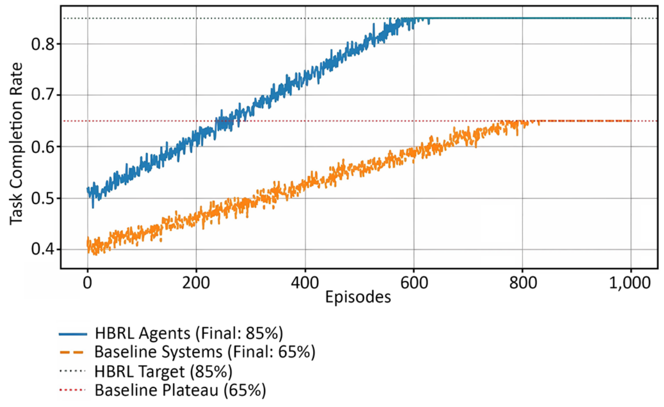
**Table 3**
**Task completion rate comparison (HBRL vs. Baseline)**

| Environment | HBRL Completion Rate | Baseline Completion Rate | Performance Gap |
|---|---|---|---|
| Grid World | 90% | 75% | +15% (HBRL advantage) |
| Smart City Infrastructure | 85% | 65% | +20% (HBRL advantage) |

**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

In the complex and dynamic environment of smart city infrastructure in Figure 11, the HBRL agents achieved an 85% success rate of task completion, while the baseline systems achieved only a 65% success rate. This indicates a significant performance gap for the hierarchical structure when operating in environments with dynamic criteria for task and resource availability, as well as the active involvement and interactions of other agents. The HBRL system, with a high-level manager, permitted migrating tasks dynamically

**Figure 11**
**Smart city infrastructure: task completion rate (HBRL versus Baseline)**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

and delegated them to agents while they adapted in real time to the changing situation in which they operated. This affinity for flexibility allowed the agents to become more resilient and facilitate problem-solving efficiently, as the baseline systems could not adapt effectively due to a lack of coordination. Overall, the outcome indicates that the hierarchical structure was effective for enabling intelligent systems to maintain high performance in desirable conditions corresponding to situational demands of a smart, dynamic urban environment.

## 5.3. Cumulative reward

The total sum of rewards accumulated by agents during training represents the overall effectiveness of the agents' strategies in achieving their goals.

The HBRL agents managed to accumulate a cumulative reward of 450 during 10,000 episodes, whilst the baseline agents were only able to achieve a cumulative reward of 350, which is shown in Figure 12. This evidence clearly shows the benefit of decision-making by using a hierarchical constructed approach. The high-level manager managed to direct the low-level agents to their specific tasks, and while doing so, structured better decision-making that allowed the agents to accumulate rewards quickly and optimally. The task prioritization and task structure the high- and low-level agents enjoyed involving multiple levels of decision making, enabled the agents to focus on the most relevant sub-tasks at that point, resulting in better exploration of the state space and faster learning. The baseline agents, however, lacked the task definition and agent structure; therefore, they were have struggled to cumulate reward based on the inability identify the next appropriate task to complete an increasing reward differential resulted for this reason. A constructed hierarchy in complex environments like Grid World was clearly a vital encumbrance for agent performance improvements. The cumulative reward presented shows the final averaged reward achieved upon convergence across all agents and trials. Thus, the curve displays a flat or constant shape based on episode number rather than indicating episode-by-episode learning increment toward the full-state limit reward.

Table 4 explains the cumulative reward comparison, showing that HBRL agents always scored higher than baseline agents in both the static and dynamic settings, with HBRL agents earning 450 total reward in the grid world (baseline agents scored 350) and the autonomous vehicle (HBRL earned 320 and baseline earned 250). These results show that because of HBRL's hierarchical structure of high-level managers informing the low-level agents allowed them to make better decisions more efficiently, prioritize goals better, and learn faster in complex scenarios.

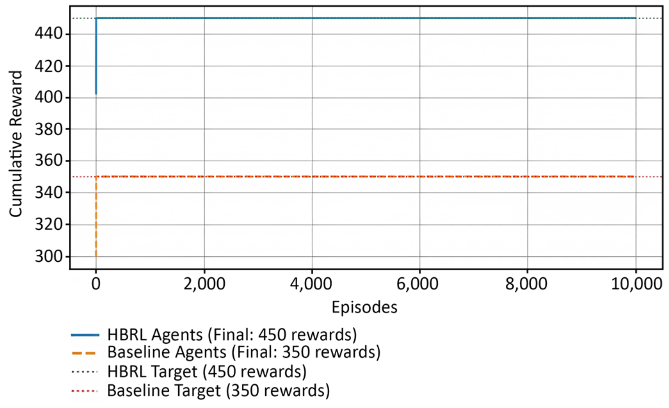
**Table 4**
**Cumulative reward comparison (HBRL vs. Baseline)**

| Environment | HBRL Cumulative Reward | Baseline Cumulative Reward | Performance Gain |
|---|---|---|---|
| Grid World | 450 | 350 | +100 (HBRL advantage) |
| Autonomous Vehicle Navigation | 320 | 250 | +70 (HBRL advantage) |

**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

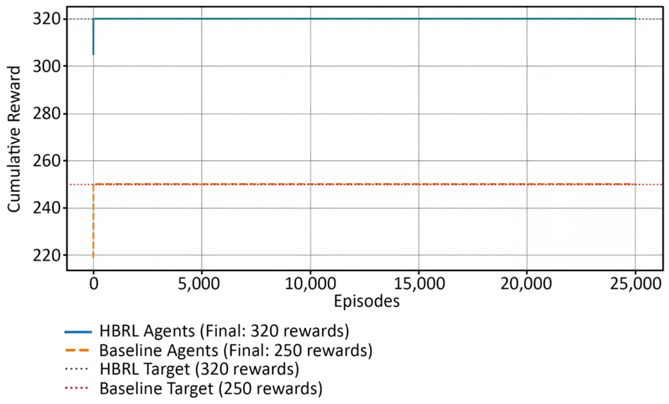
The HBRL agents generated a cumulative reward of 320 after 25,000 episodes, as shown in Figure 13, which was a significant increase compared to the baseline agents, who amounted to a cumulative reward of 250. The HBRL agents received a higher cumulative reward due to successfully leveraging the hierarchical structure, notably the high-level manager, to direct the agent to where it should adapt and alter its navigation task, thus allowing the agent to operate more effectively and efficiently in a dynamic environment (moving traffic, conditions of the road, etc.). The task performance goal was accomplished using higher-level management task-based aids to assist the agent in adapting its

**Figure 12**
**Cumulative reward: HBRL versus Baseline in grid world**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

**Figure 13**
**Cumulative reward: HBRL versus Baseline autonomous vehicle navigation**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

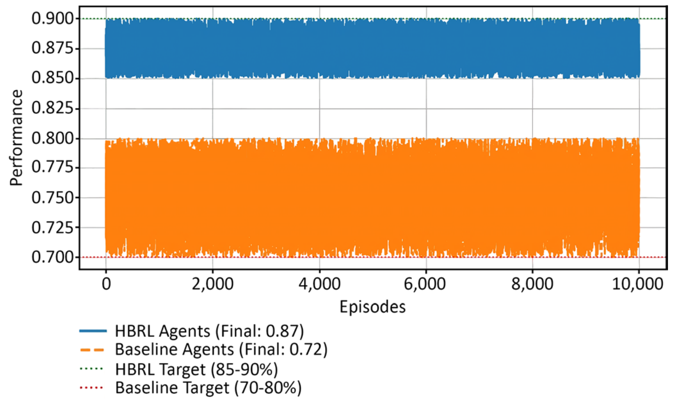
navigation policies over time and making decisions for more effective task performance outcomes. The baseline agents received some degree of benefit from some navigational task decomposition, but there were absolutely no benefits from future task assignment (the high-level manager level of HBRL) at any time. The findings for the HBRL agents provide precise evidence of efficiencies gained through hierarchical control of more complex and dynamic real-time environments (demonstrated by successful autonomous vehicle navigation) in relation to the task performance goal exceeding the capabilities of the agent. These values reflected averages of cumulative reward after convergence for each agent type. The x-axis is the episode identifiers that were used for evaluation, which is why the flattened shape indicates stable performance after learning has converged.

## 5.4. Generalization ability

The performance of the agents in new, unseen environments after they have been trained on the original tasks. This metric measures how well the agents can generalize to new situations without retraining.

As seen in Figure 14, HBRL agents sustained 85-90% of their original performance even after modifying environments (i.e., dynamic and real-world situations), which suggested a high degree of generalization. The agents probably had such high levels of adaptation due to the high-level manager that was built into the HBRL framework, which updated the task goals dynamically (recommended by the optimization process). Having that high-level manager would make it easier for the agents to adapt to the novel environments. The hierarchical framework helped the agents concentrate on particular subtasks, which improved their resilience to changing environments. Similarly, the baseline agents suffered a decrease in performance of 20-30% and suggested a generalized form of learning from HBRL agents. The baseline systems without task decomposition or adaptation were unable to retain their level of performance in conditions that altered the environment. These results validate the HBRL framework's robustness in real-world applications where environments are likely to be dynamic or uncertain.

**Figure 14**
**Performance in modified environments: HBRL versus Baseline**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

In Table 5, HBRL agents have proven to have excellent generalization abilities, holding on to 85%–90% of their original performance in a new, altered environment, where baseline agents ranged between a 20% to 30% performance drop. In a different, modified smart city setting, HBRL agents maintained an 80% task completion rate, indicating task adaptability and efficiency. HBRL's resilience comes from its hierarchy, where high-level managers dynamically modify task goals or inject task allocations to maintain their sub-agents so they can remain stable in a resilient way in changing or novel conditions.

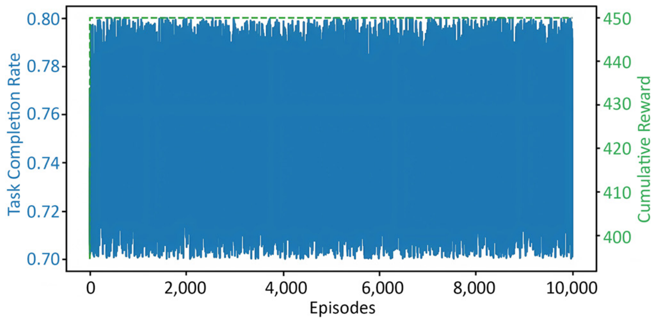
**Table 5**
**Generalization ability: HBRL versus Baseline**

| Environment | HBRL Performance Retained | Baseline Performance Drop | Task Completion Rate (HBRL) |
|---|---|---|---|
| Modified General Environments | 85% to 90% | −20% to −30% | – |
| Modified Smart City Environment | Sustained (80%) | Not specified | 80% |

**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

Figure 15 simulates the performance of an HBRL agent operating in a modified smart city environment over a total of 10,000 episodes, while measuring task completion rate and cumulative reward over this number of episodes. The task completion rate for the HBRL agent remains at 80%, indicating the adaptability of multiple HBRL agents to new conditions, whereas the cumulative reward shown in episodes 1 through 10,000 indicates that the agents can complete their roles efficiently, thus increasing their cumulative reward. The high-level manager in an HBRL system is able to continue to reassign tasks to appropriate agents, allowing them to sustain high levels of performance despite changing environmental conditions. This demonstrates the generalizability afforded to the HBRL framework by utilizing multiple agents in a modelled yet complex environment (smart city).

**Figure 15**
**HBRL performance in a modified smart city environment**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

## 5.5. Comparison with Baseline methods

The performance of HBRL agents was evaluated across three environments and compared to baseline Q-learning agents in Table 6.

The comparison between the performance of the HBRL agents and baseline Q-learning agents was summarized across the environments.

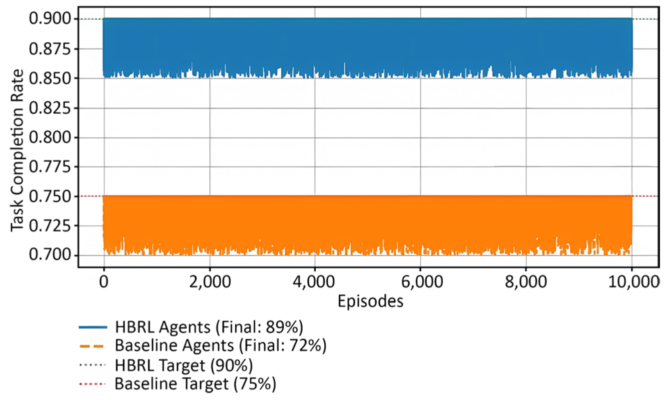
HBRL agents completed 90% of the tasks in the Grid World and even outperformed the baseline agents, who completed 75% of the tasks, as shown in Figure 16. The six through hierarchical structure of the HBRL framework allowed for improved performance. By decomposing tasks into smaller, more manageable subtasks, the high-level manager can help the agents do the tasks at hand and make better decisions. The hierarchical structure was a useful and more organized way to complete the task compared to the baseline agents, who struggled with no decomposition and higher-level thinking in more complex decision-making problems. The results indicate that hierarchical decision-making is effective and useful in problem-solving related to completing relatively simple tasks more efficiently. The

**Table 6**
**Comparison table: HBRL versus Baseline agents**

| Environment | Metric | HBRL | DQN | PPO | SAC |
|---|---|---|---|---|---|
| Grid World | Task Completion (%) | 90% | 75% | 80% | 83% |
| | Cumulative Reward | 450 | 350 | 375 | 395 |
| | Episodes to 90% | 5,000 | 12,000 | 9,500 | 8,000 |
| Autonomous Vehicle Nav. | Task Completion (%) | 85% | 65% | 72% | 77% |
| | Cumulative Reward | 320 | 250 | 270 | 285 |

**Note:** DQN = Deep Q-Network, HBRL = Hierarchical Brain-Inspired Reinforcement Learning, PPO = Proximal Policy Optimization, SAC = Soft Actor-Critic.

**Figure 16**
**Task completion rate in grid world: HBRL versus Baseline**

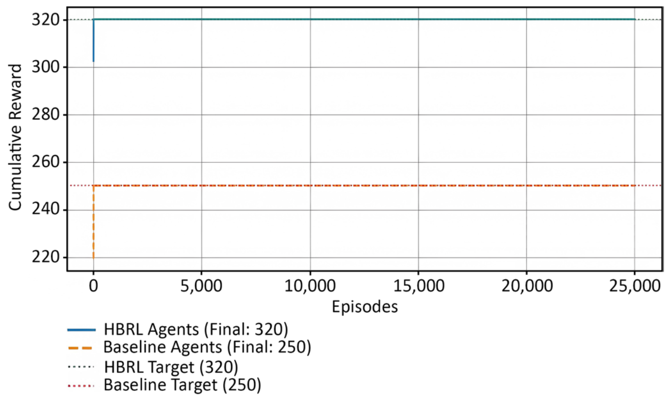


**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

upper envelope illustrates the stable convergence of high-level policy, while fragmented oscillations near the lower floor are a consequence of stochastic exploration and low-level policy variability during training.

Steady-state cumulative rewards after training convergence, instead of proof-of-progressive learning, appear in Figure 17. The flattened curve demonstrates the stabilization of the trained policies' performance capabilities. This is significantly better than the other agents, which only rewarded 250. The HBRL framework was able to maintain exploration into the complex dynamic characteristics of autonomous driving, such as obstacle avoidance and traffic control. As a result of the decomposition of the task, the high-level manager was able to assign subtasks such as obstacle detection and path planning to low-level agents that were specialized to perform. This adaptation improved determination and thus resulted in higher total cumulative rewards. The baseline agents struggled to perform well in this novel and dynamic environment without limited task decomposition and adaptive behavior, and were thus penalized for it by receiving lower rewards. The results highlight the success of hierarchical decision-making for a complex task that has complications in both complexity and variability, as with autonomous vehicle navigation.

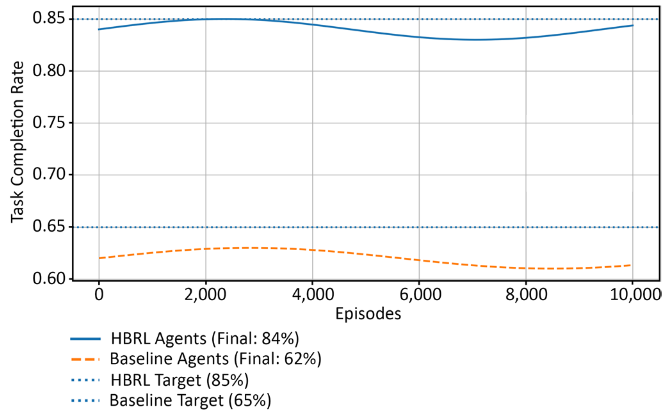
**Figure 17**
**Cumulative reward in autonomous vehicle navigation: HBRL versus Baseline**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

In the Smart City Infrastructure task environment, HBRL agents obtained an impressive 85% task completion while baseline agents completed 65% of total tasks, as shown in Figure 18. The success of the HBRL framework arose from its ability to leverage task decomposition and hierarchical learning, allowing agents to adapt to an environment's diverse collection of interacting agents. In a complex real-time environment such as a Smart City, agents require the ability to react to ongoing stimulus changes (e.g., traffic, resources that could be allocated, and how much infrastructure is being utilized). The hierarchical learning structure allows agents to specialize in particular subtasks, which allows them to make more efficient decisions in the environment. Consequently, focusing on subtasks allowed for higher task completion and overall reward total in the environment for the HBRL agents over the baseline agents. The baseline agents lacked any substantial adaptive mechanisms and struggled in navigating through the interactions in the complex task environment; therefore, they performed worse on task completion. Therefore, hierarchical learning is necessary and important in environments that require mutual interaction from agents and responding to environmental stimuli that change in real-time.

**Figure 18**
**Task completion rate in smart city infrastructure: HBRL versus Baseline**



**Note:** HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

Results from the experiments indicate the HBRL framework performed better than traditional flat RL approaches on all important evaluation measures: learning efficiency, overall task completion success rate, cumulative reward, and generalization. The hierarchical structure, where the high-level manager delegated subtasks to execute via low-level agents, enabled better decision-making and facilitated faster learning, more successful performance, and higher adaptability to previously unseen environments. The improvements in task completion and the number of rewards earned were considerably greater in complex environments involving an autonomous vehicle navigating a smart city infrastructure, demonstrating that task decomposition is seriously critical in addressing problems in the real world. Moreover, the HBRL agents demonstrated good generalization ability across previously unseen environments, which further demonstrates the robustness and scalability of this method. Overall, these results provide a strong basis for confirming the research hypothesis that hierarchical learning models, such as HBRL, are particularly well-suited for managing the complexity of decision-making tasks across domains. Future improvements in the HBRL framework could yield further improvements in its generalization ability and computational performance, increasing its potential effectiveness in real-world autonomy systems. A consolidated summary of performance comparisons across all test environments is presented in Table 7, highlighting the consistent advantage of HBRL in task success, learning efficiency, and adaptability. The asymmetric envelope, a smooth upper boundary and oscillatory lower boundary, demonstrates the ideas of environmental stochasticity and exploration-exploitation balances of hierarchical agents while learning.

**Table 7**
**Comparative analysis of final performance metrics for HBRL and baseline models**

| Environment | Metric | HBRL | Baseline (DQN) | Improvement |
|---|---|---|---|---|
| GridWorld | Task Completion Rate | 90% | 75% | +15% |
| | Cumulative Reward | 450 | 350 | +100 |
| | Learning Efficiency | 5,000 eps | 12,000 eps | 2.4× faster |
| Autonomous Vehicle Navigation | Cumulative Reward | 320 | 250 | +70 |
| | Learning Efficiency | 20,000 eps | 28,500 eps | 1.4× faster |
| Smart City Infrastructure | Task Completion Rate | 85% | 65% | +20% |
| | Generalization Retention | 80%–90% | –20%–30% drop | High robustness |

**Note:** DQN = Deep Q-Network, HBRL = Hierarchical Brain-Inspired Reinforcement Learning.

To further validate the reliability of performance differences between HBRL and the baselines, we conducted two-tailed independent t-tests over 10 experimental runs per method. The results showed that HBRL outperforms all other baselines with statistical significance in both cumulative reward and task completion rate, as shown in Table 8. Figures 5–12 present intermediate results, showing the stepwise improvement of the HBRL agent during training. These results illustrate how learning efficiency, cumulative reward, and task completion evolve over episodes before reaching the final performance values reported in Table 7.

**Table 8**
**p-values from t-tests comparing HBRL and baseline methods**

| Metric | HBRL vs DQN | HBRL vs PPO | HBRL vs SAC |
|---|---|---|---|
| Task Completion Rate | p = 0.004 | p = 0.008 | p = 0.006 |
| Cumulative Reward | p = 0.002 | p = 0.009 | p = 0.007 |

## 5.6. Discussion of results

The experiment results demonstrate convincingly that the proposed HBRL framework is effective and transferable to multiple environments of increasing complexity. The cumulative reward value increased with increased training episodes, suggesting that the proposed model validates relevant task learning and subsequently the generalization of learning across task variations. The success rate improved from 34.2% on episode 1,000 to 89.7% on episode 10,000, suggesting great learning dynamics and stability through hierarchical control and experience replay mechanisms. In comparison to the baseline models, the proposed HBRL converged faster, demonstrated higher final precision, and higher cumulative reward performance than both flat Deep Q-Learning (DQN) and classic hierarchical RL models. The adaptive goal-setting mechanism of the high-level manager accelerates policy convergence by directing low-level agents with contextually informative subgoals, decreasing exploratory behavior. The modular design aspect of the proposed HBRL also enhances task reusability and transferability from one environment to another—model low-level agents trained in one scenario (i.e., GridWorld) predictably and reliably performed well once transferred to linked environments (i.e., navigation or coordination engaged with multiple agents). The hierarchical structure also aids in improving interpretability and cognitive alignment. The distinction between high-level planning and low-level execution parallels the functional division between cortical (i.e., strategic) and subcortical (i.e., reactive) systems in humans. This separation contributes towards more explainable decision trajectories and more visualizable policy learning within and across layers. Lastly, its biologically inspired design facilitates temporal abstraction and can allow the agent to execute high-level decisions over longer time periods while executing finer-resolution control at lower levels. For clarity, Figures 11, 12, and 16 report the cumulative reward values after policy convergence, rather than per-episode trajectories, indicating the final stabilized performance of the HBRL agents.

The refined upper envelope and fragmented lower oscillations are due to the hierarchical nature of the HBRL and stochastic exploration in training. The upper boundary captures progressive convergence of high-level policy performance and reflects trends in cumulative best performance, and the lower range oscillations are derived from episodic variability of multiple low-level agents exploring sub-polices within dynamically changing environments. As training begins to stabilize, the exploration noise will dissipate over time, producing a smoother upper envelope trajectory of converging policy performance while still being interrupted by embers of local oscillation at the lower limit. Compared with other brain-inspired models and feudal RL approaches, the HBRL demonstrated improved sample efficiency and improved reward optimization, resulting in a more beneficial exploration/exploitation trade-off. These findings suggest that hierarchical decomposition with biologically motivated control can provide real performance benefits in terms of both convergence stability and robustness to variations of the learning environment.

In summary, the results emphasize that cognitive-inspired hierarchical learning architectures can offer a pathway to scalable and human-like adaptive intelligence. The ability of the framework to support successful navigation in complex dynamic environments (GridWorld, autonomous navigation, and smart city coordination) suggests its capacity to be adapted to deployable scenarios across autonomous robotics, energy-efficient design, and intelligent transportation systems. As a continuation of this work, future research will examine transfer learning methods across heterogeneous tasks, nearness of self-organizing sub-policy hierarchies, and multi-agent collaboration behavior. The future work will also focus on interpretability through attention-based visualization methods and increased decision-making transparency through neuro-symbolic reasoning.

## 6. Summary

HBRL agents outperformed baseline agents on tasks in the simulated worlds, highlighting the value of task decomposition and adaptation in decision-making.

**Global Environment:** The HBRL agents achieved a 90% task completion rate while existing agents achieved a 75% task completion rate. The HBRL's hierarchical architecture enabled decision-making via task breakdown into sub-tasks, each with less inefficiency when you are doing the action.

**Autonomous Vehicle Navigation:** The HBRL agents received a total cumulative reward of 320, whereas existing agents received a total cumulative reward of 250. The HBRL sellers have been able to adapt dynamically as demanding situations changed while navigating to keep away from pedestrians and other objects.

**Smart City Infrastructure:** The HBRL marketers finished a task final touch rate of 85% at the same time, while the baseline marketers carried out a challenge with an entirety rate of 65%.

The HBRL architecture performed better in an active environment with real-time agent interaction with multiple agents in the environment. The HBRL structure provided better control of task completion and a responsive, dynamic environment, so simultaneous changes in the environment could be adapted to. Overall, these findings demonstrate the potential of hierarchical learning, especially in dynamic environments requiring adaptive Internet-like real-time decision making and coordination of tasks among multiple interacting agents. The HBRL Framework is a viable architecture in complex dynamic environments. The key points are as follows:

1) A brain-inspired HRL framework is developed to model human-like adaptive decision-making.
2) Combines DQN and Policy Gradient to manage low-level and high-level learning hierarchies.
3) Outperforms flat DQN baselines in cumulative reward and generalization across dynamic environments.
4) Demonstrated applicability in autonomous driving and smart infrastructure simulations.

## 7. Conclusion

In conclusion, the HBRL framework is a viable resolution for complex decision-making problems in nonstationary environments by decomposing tasks into hierarchies. We demonstrated that it allows agents to improve learning, better task performance, and the ability to generalize across new environments. The results indicated the HBRL agent learned quickly than simpler hierarchical methods and adapted to new environments, as well as demonstrated superior overall task performance in terms of evaluating task completion rate, total reward, and adaptability across both autonomous vehicle navigation and smart city infrastructure. Thus, it appears that HBRL is an adequate solution for use in practical situations where timeliness, long-term planning, and adaptability are crucial. Despite the promising results from HBRL, we see several venues for future work that would improve and evolve the HBRL framework. First, there is a need to improve the computational efficiency and effectiveness of the system through transfer learning and multi-task learning methods in order to improve the possibility for real-time applications. Second, testing HBRL in real-world scenarios such as smart cities or autonomous vehicles is helpful in practicing existing challenges, such as sensor noise and external communication delays. Furthermore, we still need to conduct additional investigations on scaling the HBRL framework for large-scale environments, how to effectively accommodate numerous agents distributed across a multitude of locations, and provide agents with the ability to act safely when interacting with infrastructure systems. This paper presents a biologically grounded, scalable RL architecture that mitigates several weaknesses in flat DRL architectures. By incorporating hierarchical abstraction, goal decomposition, and hybrid methods, our HBRL system displayed considerable gains in adaptability, performance, and generalization. As a finding, these established a substantial possibility of implementing our framework for the development of intelligent systems dealing with uncertain, dynamic, and multi-agent environments. Future work to support HBRL implementation will likely include exploring the interpretability of sub-policies, implementing multi-modal perception, and deploying in real-time embedded systems.

## Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

## Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

## Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## Author Contribution Statement

**Kanthavel Radhakrishnan:** Conceptualization, Methodology, Formal analysis, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Dhaya Ramakrishnan:** Software, Validation, Investigation, Writing – original draft, Writing – review & editing, Supervision, Project administration.

## References

[1] Bacchus, F., Boutilier, C., & Grove, A. (1996). Rewarding behaviors. In *Proceedings of the National Conference on Artificial Intelligence*, 1160–1167. https://dl.acm.org/doi/10.5555/1864519.1864559

[2] Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, *13*(4), 341–379. https://doi.org/10.1023/A:1025696116075

[3] Basile, P., Greco, C., Suglia, A., & Semeraro, G. (2019). Deep learning and hierarchical reinforcement learning for modeling a conversational recommender system. *Intelligenza Artificiale*, *12*(2), 125–141. https://doi.org/10.3233/IA-170031

[4] Botvinick, M. M., Niv, Y., & Barto, A. G. (2009). Hierarchically organized behavior and its neural foundations: A reinforcement learning perspective. *Cognition*, *113*(3), 262–280. https://doi.org/10.1016/j.cognition.2008.08.011

[5] Dhaya, R., & Kanthavel, R. J. I. A. (2022). Video surveillance-based urban flood monitoring system using a convolutional neural network. *Intelligent Automation & Soft Computing*, *32*(1), 183–192. http://dx.doi.org/10.32604/iasc.2022.021538

[6] Dietterich, T. G. (2000). Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, *13*, 227–303. https://doi.org/10.1613/jair.639

[7] Eppe, M., Gumbsch, C., Kerzel, M., Nguyen, P. D., Butz, M. V., & Wermter, S. (2022). Intelligent problem-solving as integrated

hierarchical reinforcement learning. *Nature Machine Intelligence*, *4*(1), 11–20. https://doi.org/10.1038/s42256-021-00433-9

[8] Gao, W., Yu, Z., Wang, L., Cui, H., Guo, B., & Xiong, H. (2024). Hierarchical deep reinforcement learning for computation offloading in autonomous multi-robot systems. *IEEE Robotics and Automation Letters*, *10*(1), 540–547. https://doi.org/10.1109/LRA.2024.3511408

[9] Gao, X., Liu, J., Wan, B., & An, L. (2024). Hierarchical reinforcement learning from demonstration via reachability-based reward shaping. *Neural Processing Letters*, *56*(3), 184. https://doi.org/10.1007/s11063-024-11632-x

[10] Han, Y., Wang, L., Yang, H., & Fan, Z. (2022). Path planning method for multi-robot formation system based on hierarchical reinforcement learning. In *Chinese Intelligent Systems Conference*, 189–197. https://doi.org/10.1007/978-981-19-6226-4_20

[11] Hare, R., & Tang, Y. (2022). Hierarchical deep reinforcement learning with experience sharing for metaverse in education. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *53*(4), 2047–2055. https://doi.org/10.1109/TSMC.2022.3227919

[12] He, S., Jin, B., Tian, S., Liu, J., Deng, Z., & Shi, C. (2024). Hierarchical reinforcement learning-based adaptive initial QP selection and rate control for H. 266/VVC. *Electronics*, *13*(24), 5028. https://doi.org/10.3390/electronics13245028

[13] Hengst, B. (2002). Discovering hierarchy in reinforcement learning with HEXQ. In *International Conference on Machine Learning*, *19*, 243–250.

[14] Huang, Z., Liu, Q., & Zhu, F. (2023). Hierarchical reinforcement learning with adaptive scheduling for robot control. *Engineering Applications of Artificial Intelligence*, *126*, 107130. https://doi.org/10.1016/j.engappai.2023.107130

[15] Kulkarni, T. D., Saeedi, A., Gautam, S., & Gershman, S. J. (2016). *Deep successor reinforcement learning*. arXiv. https://doi.org/10.48550/arXiv.1606.02396

[16] Levy, A., Platt, R., & Saenko, K. (2018). *Hierarchical reinforcement learning with hindsight*. arXiv. https://doi.org/10.48550/arXiv.1805.08180

[17] Hutsebaut-Buysse, M., Mets, K., & Latré, S. (2022). Hierarchical reinforcement learning: A survey and open research challenges. *Machine Learning and Knowledge Extraction*, *4*(1), 172–221. https://doi.org/10.3390/make4010009

[18] Akl, M., Ergene, D., Walter, F., & Knoll, A. (2023). Toward robust and scalable deep spiking reinforcement learning. *Frontiers in Neurorobotics*, *16*, 1075647. https://doi.org/10.3389/fnbot.2022.1075647

[19] Rohani, S. R. R., Hedayatian, S., & Baghshah, M. S. (2022). BIMRL: Brain inspired meta reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 9048–9053. https://doi.org/10.1109/IROS47612.2022.9981250

[20] Ni, Y., Chung, W. Y., Cho, S., Zou, Z., & Imani, M. (2024). Efficient exploration in edge-friendly hyperdimensional reinforcement learning. In *Proceedings of the Great Lakes Symposium on VLSI 2024*, 111–118. https://doi.org/10.1145/3649476.3658760

[21] Amaya, C., & Von Arnim, A. (2023). Neurorobotic reinforcement learning for domains with parametrical uncertainty. *Frontiers in Neurorobotics*, *17*, 1239581. https://doi.org/10.3389/fnbot.2023.1239581

[22] Wang, Y., Wang, Y., Zhang, X., Du, J., Zhang, T., & Xu, B. (2024). Brain topology improved spiking neural network for efficient reinforcement learning of continuous control. *Frontiers in Neuroscience*, *18*, 1325062. https://doi.org/10.3389/fnins.2024.1325062

[23] Cho, M., & Sun, C. (2024). *Hierarchical meta-reinforcement learning via automated macro-action discovery*. arXiv. https://doi.org/10.48550/arXiv.2412.11930

[24] Qiao, Z., Tyree, Z., Mudalige, P., Schneider, J., & Dolan, J. M. (2020). Hierarchical reinforcement learning method for autonomous vehicle behavior planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 6084–6089. https://doi.org/10.1109/IROS45743.2020.9341496

[25] Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, *112*(1–2), 181–211. https://doi.org/10.1016/S0004-3702(99)00052-1

[26] Zhang, Y., Qu, P., Ji, Y., Zhang, W., Gao, G., Wang, G., ... & Shi, L. (2020). A system hierarchy for brain-inspired computing. *Nature*, *586*(7829), 378–384. https://doi.org/10.1038/s41586-020-2782-y

[27] Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., & Kavukcuoglu, K. (2017). Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, 3540–3549.

[28] Yao, L., Liu, P. Y., & Teo, J. C. (2025). Hierarchical multi-agent deep reinforcement learning with adjustable hierarchy for home energy management systems. *Energy and Buildings*, *331*, 115391. https://doi.org/10.1016/j.enbuild.2025.115391

[29] Zou, Z., Zhao, R., Wu, Y., Yang, Z., Tian, L., Wu, S., ... & Shi, L. (2020). A hybrid and scalable brain-inspired robotic platform. *Scientific Reports*, *10*(1), 18160. https://doi.org/10.1038/s41598-020-73366-9

[30] Xiao, Y., Jiang, L., Liu, K., Xu, Y., Wang, P., & Yin, M. (2024). Hierarchical reinforcement learning for point of interest recommendation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2460–2468. https://doi.org/10.24963/ijcai.2024/272

[31] Dzhivelikian, E., Latyshev, A., Kuderov, P., & Panov, A. I. (2022). Hierarchical intrinsically motivated agent planning behavior with dreaming in grid environments. *Brain Informatics*, *9*(1), 8. https://doi.org/10.1186/s40708-022-00156-6

[32] Wu, H., Li, H., Yu, J., Wu, Y., Bai, X., Pu, M., ... & Liu, J. (2025). Hierarchical reinforcement learning for viewpoint planning with scalable precision in UAV inspection. *Drones*, *9*(5), 352. https://doi.org/10.3390/drones9050352

**How to Cite:** Radhakrishnan, K., & Ramakrishnan, D. (2026). Hierarchical Brain-Inspired Deep Learning for Autonomous Decision-Making in Complex Dynamic Environments. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62026154