university of groningen / faculty of science and engineering / artificial intelligence

Investigating Mode-Switching and Reward Stream Separation in Hard-Exploration Problems

Bachelor's Project Thesis

Peter van den Bempt, s4377400, p.j.van.den.bempt@student.rug.nl,
Supervisors: Dr M. Sabatelli & R.F. Cunha, M.Sc.

Abstract: This paper showcases the importance of segregating extrinsic and intrinsic reward streams in the context of a novel hard-exploration task. Various mode-switching algorithms (algorithms with distinct 'modes' for exploration and exploitation) employing different mechanisms, are introduced, and their performance is evaluated based on discounted returns. Unlike the Q-learning agent that was used as a baseline, which failed to escape the environment's local reward maximum, the mode-switching agents successfully navigated the environment by locating the treasure consistently. While some agents benefited from the separation of reward streams, it resulted in decreased performance for other agents, especially during training. The findings suggest that future research on the efficacy of reward stream separation should explore environments where pure exploration through intrinsic motivation is not the optimal strategy.

1 Introduction

In the field of reinforcement learning (RL), agents learn how to behave in an environment by performing actions and receiving feedback in the form of rewards. The goal of the agent is to learn a policy; a strategy that maps states to actions, which maximizes the cumulative reward over time. Reinforcement learning agents learn from interaction with the environment, can make decisions autonomously and are applicable to a wide range of problems.(Sutton & Barto, 2018). Within the field of reinforcement learning, the intricate trade-off between exploration and exploitation presents a fundamental challenge. Exploration is defined as the process by which an agent actively seeks out and takes actions with the goal of gaining new information about the environment. Conversely, during exploitation the agent seeks to maximize the cumulative reward over time, using the information it already has.

Environments in which rewards are provided infrequently are referred to as sparse-reward environments. In a maze scenario, for instance, an agent might only receive a reward upon successfully exiting the maze. This poses a substantial learning challenge, because the agent does not get much useful feedback to update its beliefs about the environment. Apart from being sparse, rewards can also be deceptive. Deceptive rewards appear favorable from the perspective of the agent, but, in reality, they hinder the agent's ability to achieve optimal performance. In solving these environments, the agent is compelled to forego immediate rewards in pursuit of a potentially higher reward. Formally, deceptive-reward environments can be said to contain local reward maxima. When an environment contains sparse rewards, deceptive rewards, or a combination of both, it poses what is known as a hard-exploration problem (Ecoffet et al., 2021).

Hard-exploration problems are effectively addressed with the utilization of intrinsic rewards (Barto, 2013; Parisi et al., 2021; Burda et al., 2018), as opposed to solely using extrinsic rewards. Intrinsic rewards are often a representation of curiosity, wherein the agent actively pursues a measure of novelty, whereas extrinsic rewards are directly provided by the environment. Additionally, the recently introduced mode-switching paradigm proposes two distinct modes for agents: exploration mode and exploitation mode. Further clarifications of these concepts will be provided later.

In this paper, the significance of separating extrinsic rewards from intrinsic rewards is demon-

strated, by introducing a novel environment: TunnelVision, and assessing the performance of several algorithms with and without separated rewards streams. In the TunnelVision environment, the agent is tasked with navigating a grid world characterized by sparse rewards, explicit local maxima and a single optimal reward, exemplifying a challenging hard-exploration problem. The performance of a baseline algorithm (Q-learning) will be assessed, as well as the performance of mode-switching algorithms with and without separated reward streams. Subsequently, the aim is to answer the question whether the separation of reward streams positively affects performance, and, if confirmed, to identify the contributing factors to this effect.

A classic exploration technique is Upper Confidence Bound (UCB) (Auer, 2003). UCB is based on the principle of optimism in the face of uncertainty. It selects actions that maximize a combination of their estimated value and a confidence term that reflects the uncertainty in the estimate. By placing value on uncertain actions, exploration is maintained. Another technique is softmax or Boltzmann exploration (Cesa-Bianchi et al., 2017). This method selects actions probabilistically based on their estimated values. Actions with higher estimated values have a higher probability of being chosen, but all actions have a non-zero probability, introducing a degree of exploration.

As previously mentioned, hard-exploration problems are typically approached using intrinsic rewards. Intrinsic rewards are generated within the agent itself and serve as a means to encourage exploration and learning in the absence of frequent or informative extrinsic rewards. However, hard-exploration problems continue to pose significant challenges. A useful illustration of the difficulties associated with deceptive rewards is found in the work by Burda et al. (2018). In the game Montezuma's Revenge, their agent failed to complete the first level because it could not refrain from using the keys it found due to the immediate reward.

Pîslar et al. (2022) initiated a shift in approach by introducing the concept of mode-switching, directing attention towards the timing of exploration as opposed to solely considering exploration ratios. This paper also introduced a critical consideration that was not explicitly emphasized: the separation of extrinsic and intrinsic reward streams. The mode-switching mechanism that they introduced proposes two distinct modes for agents: exploration mode and exploitation mode. In exploitation mode, the agent simply pursues the extrinsic reward provided by the environment. Conversely, in exploration mode, the agent solely focuses on an intrinsic reward which can take various forms. It is crucial to note that, during exploration mode, the agent completely ignores extrinsic rewards; in previous research these reward streams are combined, usually through summation, instead of being treated individually (Parisi et al., 2021; Burda et al., 2018). This temporary indifference towards extrinsic rewards allows the agent to engage in prolonged periods of exploration, discovering entirely new paths without being drawn to local reward maxima.

2 Method

2.1 Preliminaries

First, Markov Decision Processes (MDP's) will be covered. MDP's are the mathematical framework for reinforcement learning problems. The components of an MDP are

- Set of possible states: $\mathcal{S}$ where $s_t \in \mathcal{S}$ is the current state.
- Set of possible actions: $\mathcal{A}$ where $a_t \in \mathcal{A}$ is the current action.
- Transition function: $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$.
- Reward function: $\Re : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ which returns r_t.

Together, these components define a Markov Decision Process $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, \mathcal{R} \rangle$. The MDP framework assumes the Markov property, which requires that the future state depends only on the current state and action. In other words, all the information influencing the transition to the next state, is encapsulated within the current state. This property is assumed throughout the study. The interaction between the agent and the environment under the MDP framework is shown in Figure 2.1 below. The

next section will introduce the TunnelVision environment, which aligns with the framework of an MDP.

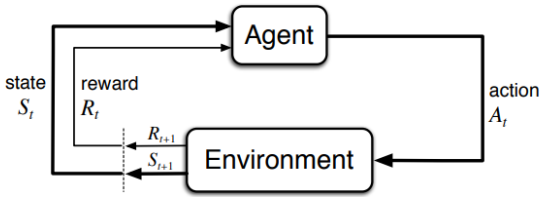

Figure 2.1: The agent–environment interaction in a Markov decision process.

In order to learn how to act in an MDP, the agent uses and improves a policy $\pi(a|s)$. The policy determines the action a taken in state s. By following the policy, the agent generates a sequences of states, actions and rewards that make up a trajectory. The goal of the agent is to find a policy that maximizes the estimated cumulative sum of rewards (the return). In this study, the agents will be using *action value functions* to compare the desiredness of performing certain actions in certain states. The action value function is defined as

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t \,|\, S_t = s, A_t = a]. \tag{2.1}$$

Here, the estimated value of the state-action pair (or q-value) is the expected return (G_t) when performing action a in state s, and adhering to policy π for the remainder of the trajectory. Using these values, policies can be compared with each other. A policy π is considered superior to or equal to another policy π', if the expected return under π is greater than or equal to that under π' for all states.

2.2 Q-learning

All of the algorithms in this study are derived from the Q-learning algorithm (Watkins & Dayan, 1992). Q-learning attempts to find an optimal policy for a given finite MDP. Since the actual estimated rewards associated with state-action pairs are unknown, the algorithm works by iteratively updating its q-values from sampled experience. The algorithm maintains a Q-table; a tabular structure containing value estimates for each state-action pair. The Q-table is initialized with all values set to 0. Equation 2.2 shows the update equation.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \tag{2.2}$$

$Q(s_t, a_t)$ is the q-value, representing the estimated cumulative reward for taking action a_t in state s_t, at a certain time t. The learning rate α regulates the degree to which new information overrides the existing Q-values, and r_{t+1} is the immediate reward received after taking action a_t in state s_t. The discount factor γ determines the balance between future and immediate rewards. A larger value for γ signifies greater emphasis on future rewards. Finally, $\max_a Q(s_{t+1}, a)$ is the maximum Q-value among all possible actions a in the next state s_{t+1}.

At every step, the q-value for the experienced state-action pair is nudged towards the difference between the observed immediate reward plus the expected value of the next state, and the previous estimate of the state-action pair.

2.3 Mode-switching

The mode-switching paradigm is a novel way to balance exploration (performing actions with the goal of gaining new knowledge about the environment), and exploitation (exploiting the gained knowledge to perform well at the task). In this paradigm, the exploitation mode is defined as the greedy pursuit of extrinsic rewards. In contrast, during exploration mode, the agent exclusively seeks an intrinsic reward, for which there are many possible implementations. In this study the intrinsic reward is based on visitation counts, which is a particularly simple approach that is well-suited for solving small discrete Markov decision processes (Kolter & Ng, 2009; Tang et al., 2017; Parisi et al., 2021), such as TunnelVision. The intuition behind visitation counts is that more frequently visited states imply reduced uncertainty regarding the rewards associated with those states, and should therefore yield a low intrinsic reward. The intrinsic reward (r_i) is defined as

$$r_i = \frac{1}{N + 1}, \tag{2.3}$$

where N is the visitation count for a given state. In exploration mode, the agent employs an *epsilon-greedy* policy, incorporating random actions into

the otherwise greedy pursuit of intrinsic rewards. An epsilon-greedy policy works as follows: with a probability of ϵ, the agent chooses to explore and randomly selects an action. Conversely, with a probability of $1 - \epsilon$, the agent opts to exploit by selecting the action with the highest value estimate. The value for epsilon decays over time and can be tuned as a hyperparameter.

In this study we explore three distinct mechanisms for mode-switching: blind switching, epsilon-visitation switching and trigger-state switching. Blind switching is the most straightforward method of the two, involving a switch after a predetermined number of steps. The other two mechanisms are informed switching mechanisms, where a trigger signal akin to a heuristic is used to switch between modes. In this study the signal that is used is simply the visitation count of the encountered state. If this value surpasses a designated threshold, the agent switches to exploitation mode. Conversely, if the count drops below a threshold, the agent reverts to exploration mode. The epsilon-visitation mechanism expands on this signal by introducing stochasticity. A random value is generated, and if it is lower than the epsilon value, the trigger signal is disregarded and the agent switches modes. If not, the switching process proceeds as described above.

In the trigger-state switching approach, a list of so-called trigger states is maintained. States present in the list are exempt from causing the agent to switch again. If we consider the trigger signal as a heuristic, this approach makes sense intuitively. An intelligent agent, having witnessed the starting square on numerous occasions, would not view this as an appropriate signal to repeatedly shift to exploitation mode. On the other hand, encountering a previously unknown square and surpassing the threshold after multiple instances serves as a valuable indication that the agent has gained substantial knowledge about the environment and should transition to exploitation mode.

2.4 Separation of reward streams

To completely separate extrinsic rewards from intrinsic rewards for the purpose of this study, a second Q-table is introduced. In exploitation mode, the agent references the first Q-table (Q_e) to determine its action, whereas, in exploration mode, it references the second Q-table (Q_i). During the update step the Q-tables are independently updated: Q_e is updated using the extrinsic reward, and Q_i is updated using the intrinsic reward. Notably, during exploitation mode, Q_i remains unaltered. Algorithm 2.1 provides the pseudocode for a generic Q-switching algorithm with separated reward streams.

Algorithm 2.1 Q-Switching_Separate

Require: State space (S) and action space (A)
Require: Initialize $Q_e(s, a)$ is 0 for all $s \in S, a \in A$
Require: Initialize $Q_i(s, a)$ is 0 for all $s \in S, a \in A$
Require: Learning rate α, discount factor γ, epsilon ϵ

for each episode **do**
  Initialize state s
  while episode is not terminated **do**
    Update mode via switching mechanism
    if in exploration mode **then**
      if random number $< \epsilon$ **then**
        Choose a random action a from A
      else
        Choose action a with the highest Q_i value for state s
      end if
    end if
    if in exploitation mode **then**
      Choose action a with the highest Q_e value for state s, and set $r_i := 0$
    end if
    Take action a, observe reward r_e, reward r_i and new state s'
    Update $Q_e(s, a) \leftarrow Q(s, a) + \alpha \cdot [r_e + \gamma \cdot \max_{a'} Q_e(s', a')]$
    Update $Q_i(s, a) \leftarrow Q(s, a) + \alpha \cdot [r_i + \gamma \cdot \max_{a'} Q_i(s', a')]$
    Move to the new state s'
  end while
end for

2.5 Environment

The standard TunnelVision environment is a grid world with dimensions (4, 12). The state and action spaces are both discrete. The grid consists of 48 states, each corresponding to a square on the grid. The action space includes four possible actions: up, down, right, and left. Every transition is deterministic; when the agent performs a specific action in a given state, it will invariably end up in the same next state. The agent is spawned in the top-left corner at the beginning of every episode,

and its objective is to locate the treasure situated at the bottom-right corner. Episodes terminate under three conditions: encountering toxic gasses at specific grid locations, resulting in an immediate episode termination with a reward of zero; encountering inferior reward states, leading to an episode termination with a reward of 0.25; or successfully reaching the treasure, resulting in an episode termination with a reward of 1. A notable property of the standard environment is its stationarity; the environment dynamics of the MDP remain constant. Therefore, once an agent has successfully found the optimal policy, there is no requirement for it to undergo changes, because the environment itself does not change. Hence the need for continued exploration eliminated from that point onward.

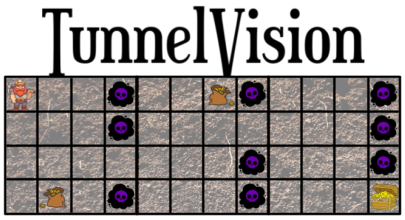

Figure 2.2: Representation of the TunnelVision environment. The agent is depicted in the top-left corner. The toxic gasses are indicated with skulls and the treasure is shown at the bottom-right corner.

2.6 Algorithms

For this study, a total of seven algorithms were created and assessed. Their distinctions arise along two dimensions: the mode-switching mechanism as described in section 2.3, and the separation of reward streams as outlined in section 2.4. As a baseline, conventional Q-learning featuring exponential epsilon decay was used. Below is an overview of the algorithms.

- Q-learning with epsilon-decay

The Q-learning algorithm as explained in section 2.2, with an epsilon-greedy (see section 2.3) *behavior policy*. The behavior policy is the policy that is used to select the agents actions a, whereas the *target policy* is used to update the agents value estimates. For Q-learning, the target policy is a greedy policy with respect to the value estimates; the agent updates its estimates based on the next action with the highest Q-value. Additionally, the epsilon value for the epsilon-greedy policy decays exponentially over time.

- Q-learning with blind switching

A Q-learning algorithm with the addition of two distinct modes as specified in section 2.3. This algorithm uses the blind switching mechanism to switch between modes.

- Q-learning with blind switching and separated reward streams

Similar to the preceding algorithm, except for the addition of separated reward streams, as outlined in section 2.4.

- Q-learning with epsilon-visitation switching

Also a mode-switching algorithm, but this algorithm uses the epsilon-visitation switching mechanism.

- Q-learning with epsilon-visitation switching and separated reward streams

The separated variant of the preceding algorithm.

- Q-learning with trigger-state switching

This mode-switching algorithm uses the trigger-state switching mechanism (section 2.3).

- Q-learning with trigger-state switching and separated reward streams

This is the variant of the prior algorithm featuring separated reward streams.

3 Experimental setup

For all of the experiments conducted in this study, agents were trained for 100,000 episodes. Throughout this training period, the agents were evaluated every ten episodes. During these evaluations, agents were configured to be greedy with respect to their

Q-values for a single episode, and the resulting returns were recorded. Agents featuring separated reward streams were configured to be greedy with respect to their Q_e-values. Additionally, a measure of the overall proportion of exploration to exploitation ($p_{\mathcal{X}}$) was tracked. This metric was computed using the respective frequencies of the two modes across every step of every episode. For the baseline which did not feature modes, the proportion of random exploratory steps was computed. All of the algorithms that were tested contain some degree of stochasticity, in order to prevent them from repetitively exploring the same states in every run. Because of this, ten runs were conducted for each experiment, and the results were averaged.

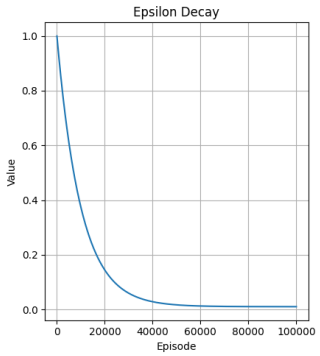

Figure 3.1: The value of epsilon throughout a run of 100,000 episodes. This exponentially decaying function asymptotically approaches a value of 0.01.

3.1 Hyperparameter optimization

The learning rate (α) and the discount factor (γ) are two hyperparameters native to the Q-learning algorithm that are present in all five variants presented in this study. To limit the amount of confounding variables, both parameters were consistently set to fixed values of 0.1 and 0.99, respectively, across all algorithms. These values are based on example problems from the textbook by Sutton & Barto (2018). The optimality of these values for this specific environment is irrelevant; it is only their uniformity across agents that is crucial, since this eliminates confounding variables. The exponential epsilon decay that is used for the baseline algorithm follows a simple formula that is applied at the end of every episode:

$$\epsilon = 0.01 + 0.99e^{-\text{decay\_rate}\cdot\text{episode}} \tag{3.1}$$

This dynamic epsilon value was also used for the epsilon-visitation switching mechanism, as well as the epsilon-greedy policy employed during exploration mode, as explained in section 2.3. Figure 3.1 illustrates how this value progresses during a run of 100,000 episodes. The decay rate was kept constant at a value of 0.0001. This value was determined through empirical testing with the aim of ensuring that the the lower bound of 0.01 was reached within a reasonable timeframe.

The three switching mechanisms introduce various tunable hyperparameters. The blind switching mechanism was set to switch to exploitation mode after 10,000 steps, and reverted back to exploration mode after 1,000 steps. The step count was not reset at the end of an episode. These numbers were chosen based on the observation that a high $p_{\mathcal{X}}$ is needed to succeed in this environmnent. For the epsilon-visitation switching mechanism, two hyperparameters were also fine-tuned: the visitation count threshold for switching to exploitation mode and the complementary threshold for switching back. Aimed at maintaining a high proportion of exploration, these values were set to 1024 and 32, respectively. Ultimately, these values did not have a significant impact, as the visitation count of the starting square, which was almost exclusively responsible for the switching in this mechanism, quickly exceeded any threshold. The trigger-state switching mechanism was created to address this problematic observation. For this switching mechanism, the threshold for transitioning to exploitation mode was kept low at a value of 2, while the threshold for switching back was set to 3200, and later 200. All of these hyperparameter values for the switching mechanisms were determined through empirical adjustments conducted in simulations.

3.2 Assessment criteria

The agents were assessed based on two main criteria: evaluation performance and training performance. The metric used to assess performance across these two domains was the (discounted) re-

turn G_t realized by the agent per episode. The return is defined as

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots = \sum_{k=0}^{N} \gamma^k r_{t+k+1} \tag{3.2}$$

where r_t is the extrinsic reward provided by the environment at time step t, T is the final time step and γ is the discount factor. Since it is only possible to obtain an extrinsic reward once per episode in this environment, this simplifies to

$$G_t = \gamma^T \cdot r_T, \tag{3.3}$$

where, once again, T is the final time step of the episode. From the standard environment representation in Figure 2.1, it can be inferred that reaching the treasure, which has an extrinsic reward of 1, requires a minimum of 16 steps. Therefore, the maximum episodic return is calculated as $\gamma^{16} \cdot 1 = 0.851$. In this study, a path is defined as the full sequence of steps taken by an agent within an episode. The episodic return for the optimal path to the first inferior reward of 0.25 is four steps long. This path represents one of the local reward maxima in the environment, with a return of $\gamma^4 \cdot 0.25 = 0.240$. Both of these calculated returns will frequently appear in the results.

Furthermore, the evaluation of agents will involve considering the proportion of exploration to exploitation ($p_{\mathcal{X}}$), as specified in the beginning of Section 3. The equation for ($p_{\mathcal{X}}$) is

$$p_{\mathcal{X}} = \frac{N_{\text{exploratorysteps}}}{N_{\text{steps}}}, \tag{3.4}$$

Where $N_{\text{exploratorysteps}}$ is the total amount of exploratory steps in a run, and N_{steps} represents the total amount of exploitation steps in a run. Note that exploitation steps can involve any method of exploration, be it intrinsically motivated through visitation counts, or random. A lower overall proportion of exploration, coupled with good evaluation and training performance, indicates quick yet effective exploration. The expectation is that agents exhibiting effective exploration will not only outperform purely exploratory agents during training, but will also demonstrate superior performance in greedy evaluation when facing more challenging environments. This expectation is grounded in the observation that agents employing pure exploration excel in the standard TunnelVision environment.

4 Results

As previously mentioned, the following parameters were fixed across all agents:

- Number of runs: 20
- Number of episodes per run: 100000
- Learning Rate (α): 0.1
- Discount Factor (γ): 0.99

4.1 Baseline

Figure 4.1 illustrates the outcomes of the baseline Q-learning agent, revealing a notable inability to find the optimal policy. Instead, the agent becomes trapped in a local reward maximum, specifically that of the first inferior reward. The ($p_{\mathcal{X}}$) measure is at a moderate value of 0.33 for this agent. In contrast to other agents, the exploratory steps for this agent were not intrinsically motivated; rather, they were random, leading to a comparatively lower effectiveness.

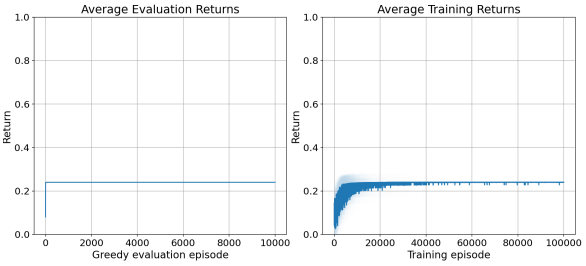

Figure 4.1: Q-Learning with Epsilon Decay Ratio of Exploration ($p_{\mathcal{X}}$): 0.32690

Upon closer inspection (Figure 4.2), it becomes evident that the variance in returns across the 20 runs is only observable during the initial evaluation; subsequent evaluations consistently yielded identical values across all runs.

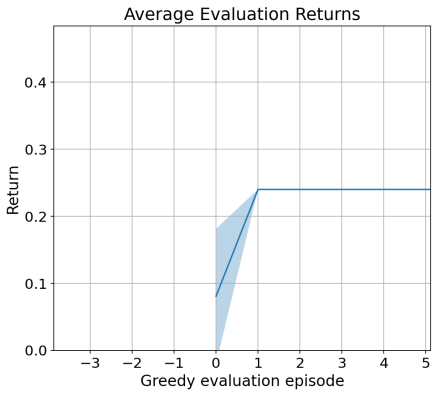

Figure 4.2: A closer look at the Q-learning evaluation performance averaged over 20 runs.

Recall that the epsilon decay adheres to equation 3.1. During training, the agent found the goal twice across all 20 runs: in episode 4963 of the second run, and in episode 2870 of the third run. During these relatively early episodes, the values for epsilon were 0.61 and 0.75, respectively. While infrequent, this shows that this agent does occasionally discover the treasure when epsilon is still high. In figure 4.3, the spike in standard deviation caused by one of these occurrences is observed.

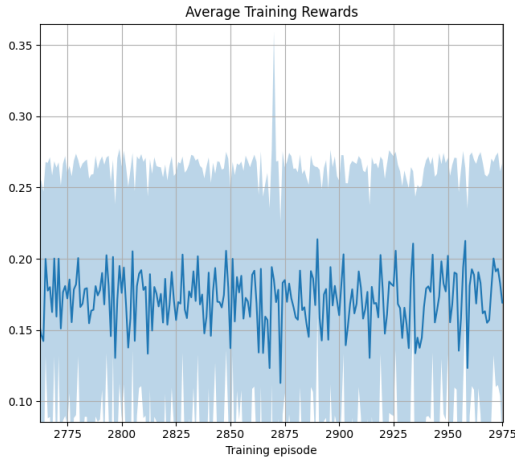

Figure 4.3: A spike in standard deviation caused by a chance encounter with the treasure.

Despite these chance encounters, the agent was unable to convert them into an optimal policy. To understand why, a deeper analysis of the agent's value estimates is needed. In Figure 4.4, a heatmap depicts the agent's rounded maximum Q-values for each state at the end of the third run. There are four Q-values per state, corresponding to each possible action. The maximum among these values is displayed, to represent the anticipated value of the action that the agent is most likely to choose. While there is some minimal information gained about the treasure's location, evident in the heightened Q-value adjacent to the treasure, the agent would need to reach the goal many times (via similar paths) for this information to propagate back toward the starting location through the Q-values.

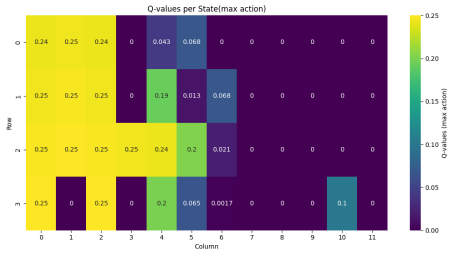

Figure 4.4: Maximum Q-values per state at the end of the third run for the baseline Q-learning agent.

4.2 Blind switching agents

The performance of the blind Q-switching agent is shown in figure 4.5. After an initial spike originating from the first local reward maximum, the agent's evaluation returns average at a value of zero for a brief period. This behavior can be attributed to the decreasing value of the intrinsic reward (as per Equation 2.3) as the run progresses. Initially, the intrinsic reward for reaching unknown states, even those with a reward of zero (toxic gasses), surpasses the appeal of reaching the first inferior reward. This occurs due to the lack of separation between extrinsic and intrinsic during evaluation. As N increases, the intrinsic reward diminishes and the agent temporarily returns to the local reward maximum.

Subsequently, the agent succeeds in discovering and consistently reaching the treasure, although this outcome does not occur in every run. This variability accounts for the considerable magnitude of the standard deviation in the evaluation plot.

The shaded region, which represents the standard deviation from the mean across runs, is bounded by adding and subtracting its magnitude from the mean. It's important to note that this shaded region does not necessitate the inclusion of every possible value within its range in the data.

Even after the averaged evaluation returns have converged, there remains a significant variance in training return averages. This is attributed to the misalignment of modes across runs; the agent has a fixed step size interval for switching between modes, but the number of steps per episode is variable. Although all 20 runs begin in exploration mode, the time it takes for them to switch modes varies. This agent's $p_{\mathcal{X}}$ is 0.91, indicating that the majority of its time was spent in exploration mode.

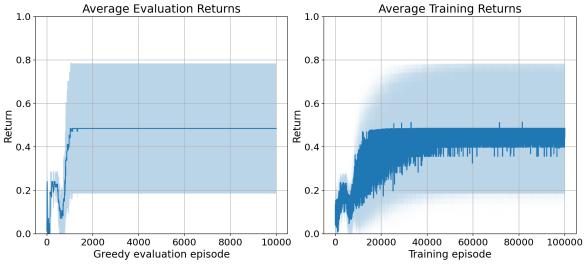

Figure 4.5: Blind Q-Switching
Ratio of Exploration ($p_{\mathcal{X}}$): 0.90967

The blind Q-switching agent with separated reward streams emerged as a successful agent. Figure 4.6 illustrates its performance. In contrast to the non-separated blind Q-switching agent, its evaluation performance does not dip to zero at the beginning. This is attributed to the agent solely relying on its knowledge of extrinsic rewards during evaluation, facilitated by its separate Q-tables.

The agent quickly finds the optimal policy, consistently achieving the highest possible return. However, the training performance never becomes optimal because the agent keeps exploring indefinitely. During its exploration intervals, the agent exclusively pursues intrinsic rewards, resulting in a low extrinsic return.

As expected, the agent's $p_{\mathcal{X}}$ is almost identical to that of the non-separated blind Q-switching agent. However, its performance is significantly better in both training and evaluation. This suggests that the separation of reward streams proved to be exceptionally valuable for the blind switching variants.

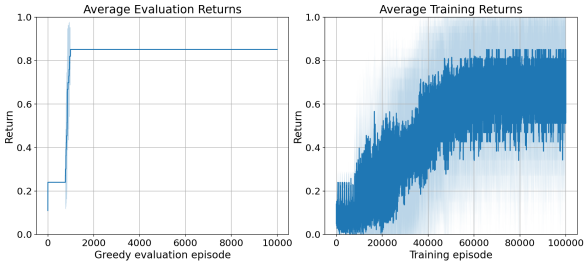

Figure 4.6:
Blind Q-Switching with Separate Q-Tables
Ratio of Exploration ($p_{\mathcal{X}}$): 0.90913

4.3 Informed switching agents

Next, the performance of the informed Q-switching agents will be assessed, starting with the epsilon-visitation switching mechanism variants. Figures 4.7 and 4.8 present the results of the non-separate and separate agents equipped with this switching mechanism.

The main observation is that, for both agents, the switching mechanism yields a $p_{\mathcal{X}}$ that is too low for this environment (approximately 0.1). As outlined in Section 3.1, the visitation count of the starting square predominantly influences the switching in this mechanism. As a result, after a certain number of episodes, the agent consistently switches to exploitation mode at the start of each new episode. This drawback prevents either agent from being able to escape the local reward maximum consistently.

There is one notable difference between the two agents: the non-separate variant was able to find the optimal policy precisely once out of 20 runs using this switching mechanism. This discrepancy also accounts for the significant standard deviation in Figure 4.7, which is absent in Figure 4.8. We anticipate that this dissimilarity does not signify generalized superior performance, as repeating the stochastic simulation of 20 runs for this agent twice using different random seeds resulted in no successful runs.

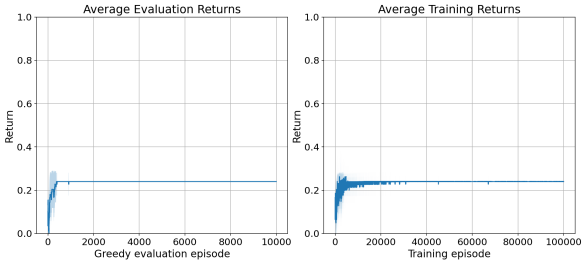

Figure 4.7:
Informed Q-Switching (Epsilon-Visitation)
Ratio of Exploration ($p_{\mathcal{X}}$): 0.09523

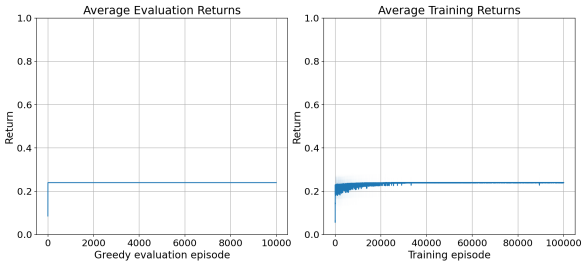

Figure 4.8:
Informed Q-Switching (Epsilon-Visitation) with Separate Q-Tables
Ratio of Exploration ($p_{\mathcal{X}}$): 0.11093

Now the performance of the informed Q-switching agents under the trigger-state switching mechanism will be presented. Figures 4.9 and 4.10 display the performance of the non-separate and separate variants, respectively.

The trigger-state switching mechanism encounters an issue opposite to that of epsilon-visitation switching, particularly for the chosen threshold value of 3200. Namely, its $p_{\mathcal{X}}$ is extremely high. Unsurprisingly, it becomes evident that in this environment, the most effective strategy for mode-switching agents with a potent intrinsic reward function, such as the one defined in Equation 2.3, is to concentrate solely on exploration. Both agents quickly identify the optimal policy as soon as the extrinsic reward propagates back to the starting area. Similarly to the non-separate blind switching agent, the non-separate trigger-state switching agent briefly averages an evaluation return of zero because the intrinsic reward outweighs the appeal of the first inferior reward during this period. It is also observed that the separate variant fails to find the optimal policy in 3 out of 20 runs.

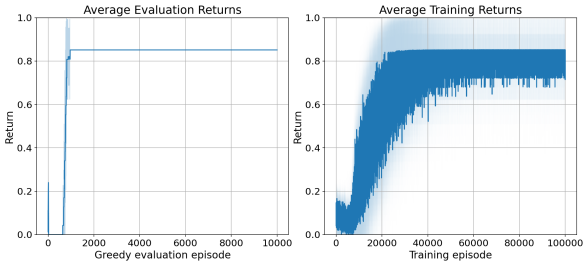

Figure 4.9:
Informed Q-Switching (Trigger-States)
Ratio of Exploration ($p_{\mathcal{X}}$): 0.99997

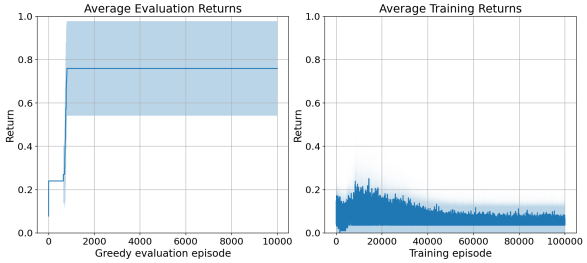

Figure 4.10:
Informed Q-Switching (Trigger-States) with Separate Q-Tables
Ratio of Exploration ($p_{\mathcal{X}}$): 0.99790

Particularly interesting is the stark contrast in training performance between the non-separate and separate variants; the non-separate variant exhibits excellent performance, whereas the separate variant performs poorly. In order to understand this difference it must be considered that the agent was almost exclusively in exploration mode during these runs. The separate variant exclusively pursues intrinsic rewards during exploration mode, hence it values every state approximately uniformly, regardless of extrinsic reward. Consequently, its training performance is severely neglected. The non-separate variant maintains an incentive to reach the treasure because it updates its single Q-table using a sum of the extrinsic and intrinsic reward. To illustrate this retention of a drive for extrinsic rewards during exploration mode, Figure 4.11 exhibits the maximum Q-values per state at the end of a run for the non-separate variant.

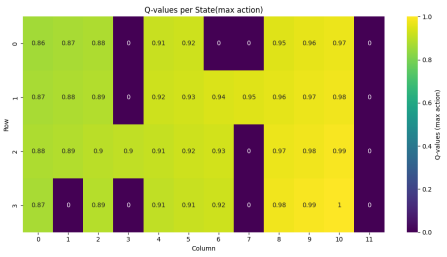

Figure 4.11: The maximum Q-values per state at the end of a run for the Informed Q-Switching agent with Trigger-State Switching.

In contrast, Figure 4.12 showcases the maximum Q_i values per state for the separate variant. These values give insights into the poor training performance of the separate variant; the agent systematically visits as many non-terminal states as possible before the episode is concluded at any terminal state due to the stochastic moves stemming from its epsilon-greedy exploration policy.

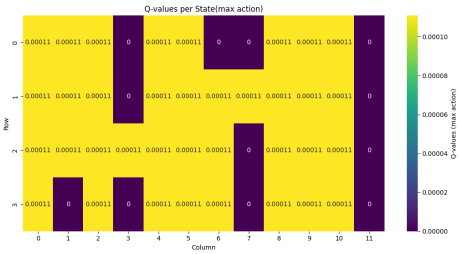

Figure 4.12: The maximum Q_i-values per state at the end of a run for the Informed Q-Switching agent with Trigger-State Switching and Separate Q-Tables.

5 Discussion

In this paper, we demonstrated the significance of separating rewards streams within mode-switching approaches for hard-exploration problems. Specifically, we conducted experiments by applying different mode-switching algorithms to an environment featuring intentionally deceptive rewards, simulating local reward maxima.

The baseline Q-learning algorithm was never able to escape the local reward maximum corresponding with the inferior reward at the bottom left corner of the environment. This is not surprising, since its only method of exploring is through random steps determined by a decaying epsilon. When the agent did find the goal by chance, it was unable to convert these encounters to an optimal policy. To achieve conversion, the agent would need to reach the goal multiple times to allow the higher reward to propagate backwards through the Q-values to the starting area.

As a result of prolonged predefined periods of continuous exploration, the blind switching agents successfully found the optimal policy. However, a notable distinction emerged between the separate and non-separate variants. The separate variant consistently achieved the optimal policy in every run, whereas the non-separate failed to do so in many instances. This difference is attributed to the separation of reward streams. The separated variant can disregard inferior rewards entirely during exploration, eliminating any distractions in its pursuit of locating the treasure. In contrast, the non-separate agent updates its value estimates based on a combination of extrinsic and intrinsic rewards, introducing a form of 'leakage' that hinders exploration efficiency. Because the intrinsic reward gradually diminishes due to its nature (Equation 2.3), the agent only has limited time to reach the treasure frequently enough to convert it into an optimal policy. After the intrinsic reward has diminished, its incentive to explore the right side of the environment disappears, leading to the suboptimal performance observed in some runs. We can conclude that the separation of reward streams has a significant positive effect on the performance of the blind switching agents under these parameters.

The informed switching agents equipped with the epsilon-visitation switching mechanism suffered from an exploration rate that was too low for this environment. This inadequacy is attributed to the switching mechanism's heavy dependence on the visitation count of the starting square, leading to a consistent switch to exploitation mode at the beginning of each new episode after the visitation count threshold has been exceeded. This limitation prevents us from drawing any definitive conclusions about the efficacy of separated reward streams for these agents.

The trigger-state switching mechanism produced a very high proportion of exploration for both agents, introducing challenges of its own. How-

ever, two insightful observations were made. Firstly, there was a significant disparity in training performance between the non-separate and separate variants. This distinction is attributed to the same 'leakage' effect that impeded the non-separate blind switching agent, yet in this context, it results in the sustained motivation to reach extrinsic rewards during training. While training performance may not be the primary focus, this grants a slight advantage to the non-separate variant. Secondly, additional experiments revealed that the non-separate variant achieves superior performance when subjected to lower 'switch-back' threshold values. At low values for this threshold, only the non-separate variant managed to sustain a favorable exploration ratio. The reasons for this remain unclear, and could serve as a topic for future investigation.

There are two factors that limit the power of this study. The proportion of exploration $p_{\mathcal{X}}$ emerges as a crucial confounding variable, significantly limiting the capacity to make definitive conclusions about agent performance. This limitation is compounded by the observation that, under the current evaluation criteria, the most effective strategy for this environment involves pure exploration, as discussed in section 4.3. Hence, for future research, the introduction of new environments that challenge this optimal strategy would provide valuable insights. Two possibilities are non-stationary environments, and environments with continuous state and/or action spaces.

References

Auer, P. (2003, March). Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, *3*(null), 397–422.

Barto, A. G. (2013). Intrinsic Motivation and Reinforcement Learning. In G. Baldassarre & M. Mirolli (Eds.), *Intrinsically Motivated Learning in Natural and Artificial Systems* (pp. 17–47). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from `https://doi.org/10.1007/978-3-642-32375-1`$_2$ doi: 10.1007/978-3-642-32375-1$_2$

Burda, Y., Edwards, H., Storkey, A., & Klimov, O. (2018, October). *Exploration by Random Network Distillation.* arXiv. Retrieved 2023-10-03, from `http://arxiv.org/abs/1810.12894` (arXiv:1810.12894 [cs, stat]) doi: 10.48550/arXiv.1810.12894

Cesa-Bianchi, N., Gentile, C., Lugosi, G., & Neu, G. (2017, November). *Boltzmann Exploration Done Right.* arXiv. Retrieved 2024-01-17, from `http://arxiv.org/abs/1705.10257` (arXiv:1705.10257 [cs, stat]) doi: 10.48550/arXiv.1705.10257

Ecoffet, A., Huizinga, J., Lehman, J., Stanley, K. O., & Clune, J. (2021, February). *Go-Explore: a New Approach for Hard-Exploration Problems.* arXiv. Retrieved 2023-10-02, from `http://arxiv.org/abs/1901.10995` (arXiv:1901.10995 [cs, stat]) doi: 10.48550/arXiv.1901.10995

Kolter, J. Z., & Ng, A. Y. (2009). *Near-Bayesian exploration in polynomial time.* Proceedings of the 26th Annual International Conference on Machine Learning, 513520.

Parisi, S., Dean, V., Pathak, D., & Gupta, A. (2021). Interesting Object, Curious Agent: Learning Task-Agnostic Exploration. In *Advances in Neural Information Processing Systems* (Vol. 34, pp. 20516–20530). Curran Associates, Inc.

Pîslar, M., Szepesvari, D., Ostrovski, G., Borsa, D., & Schaul, T. (2022, March). *When should agents explore?* arXiv. Retrieved 2023-11-09, from `http://arxiv.org/abs/2108.11811` (arXiv:2108.11811 [cs])

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: an introduction* (Second edition ed.). Cambridge, Massachusetts: The MIT Press.

Tang, H., Houthooft, R., Foote, D., Stooke, A., Xi Chen, O., Duan, Y., ... Abbeel, P. (2017). #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.

Watkins, C. J. C. H., & Dayan, P. (1992, May). Q-learning. *Machine Learning*, *8*(3), 279–292. Retrieved 2024-01-15, from

https://doi.org/10.1007/BF00992698 doi:
10.1007/BF00992698