# A Probability-Based Approach to the K-Armed Bandit Problem

Gavin Hull
Memorial University of Newfoundland and Labrador
ghull@mun.ca

June 12, 2024

**Abstract**

This paper presents a novel probabilistic strategy for addressing the k-armed bandit problem, leveraging probability computations derived from observed rewards. Our method is benchmarked against standard algorithms, including greedy, epsilon-greedy, optimistic, and gradient bandit strategies. Experimental results demonstrate that the proposed method achieves a 5% improvement in average accumulated reward (AAR) and 10% improvement in percent optimal action (POA) over most standard algorithms in stationary environments, and an increase over all standard algorithms in some non-stationary environments, making it a competitive alternative to existing methods. All code is made available on GitHub [1].

## 1 Introduction

The k-armed bandit problem, originally proposed by Herbert Robbins in 1952 [5], was developed to study the trade off between exploration and exploitation. Over the past 70 years, the Multi-Armed Bandit problem has become a touchstone in the field of reinforcement learning with hundreds of proposed solutions. This problem is critical in various applications, including online advertising [1, 2] and clinical trials [6]. Traditional strategies include greedy, epsilon-greedy, optimistic, and gradient bandit algorithms. Despite their utility, these methods often face limitations in balancing exploration and exploitation, especially in non-stationary environments where reward distributions change over time. This paper introduces a new probabilistic strategy that computes the likelihood of each bandit yielding the highest reward based on historical data, potentially offering a more robust solution across different environments. We systematically compare this novel approach with established methods in both stationary and non-stationary settings to demonstrate its efficacy.

## 2 Literature Review

The k-armed bandit problem has been extensively studied with numerous strategies having been developed to balance the exploration-exploitation trade-off. Greedy algorithms exploit current

[1] All code required to replicate results can be found here

knowledge but may converge to suboptimal actions. Epsilon-greedy strategies introduce exploration with a small probability of selecting random actions. Optimistic initial values foster exploration by initializing reward estimates higher than expected. Gradient bandit methods employ preference-based approaches to guide action selection. Recent advancements include Bayesian approaches [4, 3] and contextual bandit algorithms, which incorporate contextual information to improve decision-making. Our proposed probabilistic strategy differs by directly computing the probability of each bandit yielding the highest reward based on historical data, offering a theoretically grounded and computationally efficient alternative. Compared to methods like Bayesian inference, which often rely on complex posterior updates, our approach simplifies the computation while maintaining competitive performance.

This paper serves as a proof of concept for our proposed strategy, and for that reason we compare our algorithm only to the most basic of algorithms; that is to say, excluding Bayesian approaches. Comparison amongst more recent advancements in newer environments like contextual bandits is reserved for future work.

# 3 Methodology

Our proposed strategy utilizes mathematical derivations to optimally balance exploration and exploitation. We derive the probability that the mean reward of a given bandit exceeds that of all others. This probability guides bandit selection, theoretically favoring bandits with higher mean rewards until the uncertainty of the sample mean is sufficiently reduced.

Let $Y \sim N(\mu_Y, \sigma_Y^2)$ represent the reward distribution of the selected bandit and $X_i \sim N(\mu_{X_i}, \sigma_{X_i}^2)$ represent the distributions of the other bandits. The probability $p(Y > \max(X_1, X_2, ..., X_k))$ is computed as follows:

$$
\begin{aligned}
& p(Y > \max(X_1, X_2, ..., X_k)) \\
= \ & p(Y > X_1, Y < X_2, .., Y > X_k) \\
= & \int_{-\infty}^{\infty} p(Y > X_1, Y < X_2, .., Y > X_k | Y = y)\phi_Y(y)dy \\
= & \int_{-\infty}^{\infty} \prod_{i=1}^{k} p(Y > X_i | Y = y)\phi_Y(y)dy \\
= & \int_{-\infty}^{\infty} \prod_{i=1}^{k} \Phi_{X_i}(\frac{y - \mu_{X_i}}{\sigma_{X_i}})\phi_Y(y)dy
\end{aligned}
\tag{1}
$$

where $\Phi_{X_i}$ is the CDF of $X_i$ and $\phi_Y$ is the PDF of $Y$.

### 3.1 Algorithmic Enhancements

Three challenges must be addressed for practical implementation:

**Efficient CDF Calculation:** The normal CDF lacks a closed-form solution, but it can be approximated using trigonometric functions. We employ the formula $\Phi(x) \approx 0.5 \cdot (1 + \tanh(1.142x \cdot (1 + 0.043595x^2)))$, which has an absolute error less than $10^{-2}$.

**Constant-Memory Standard Deviation Calculation:** We use the update rule $\mu_{t+1} = \mu_t + \frac{x_{t+1} - \mu_t}{t+1}$ for mean calculation because it can be calculated in constant time. For standard deviation, let $S$ be the sum of rewards and $S_2$ be the sum of squared rewards. The standard deviation at time $t$ can be defined as $\sigma_t = \sqrt{\frac{1}{t}(S_2 - \frac{S^2}{t})}$. Therefore, by the central limit theorem, the sample mean's standard deviation is $\sqrt{\frac{1}{t^2}(S_2 - \frac{S^2}{t})}$. Since $t, S$, and $S_2$ have trivial constant memory formulations, so does the standard deviation.

**Adaptation to Non-Stationary Environments:** To prevent overconfidence in non-stationary settings, we use geometrically weighted means and standard deviations with weight (or 'discount') $0 < \gamma < 1$, ensuring our algorithm remains adaptive to changes.

**Algorithm 1** $\mathbb{O}(1)$ Standard Deviation Calculation

```
Initialize: $\Gamma \leftarrow \gamma, S \leftarrow 0, S_2 \leftarrow 0, n \leftarrow 0$
while true do
    $S_2 \leftarrow \gamma S_2 + r_i^2$
    $S \leftarrow \gamma S + r_i$
    $\Gamma \leftarrow \gamma \Gamma + 1$
    $n \leftarrow n + 1$
    $\sigma_i = \sqrt{\frac{1}{\Gamma^2}(S_2 - \frac{S^2}{\Gamma})}$
end while
```

# 4 Experimental Setup

Experiments were conducted on 10 bandits in stationary and non-stationary environments. Stationary bandit means $\mu_i$ were sampled from $N(0, 1)$ with rewards drawn from $N(\mu_i, 1)$. Non-stationary bandits were tested under drifting, mean-reverting, and abruptly changing scenarios. Each strategy was evaluated over 1,000 simulations for stationary bandits (1,000 timesteps each) and non-stationary bandits (10,000 timesteps each).

### 4.1 Hyperparameter Optimization

Hyperparameters were optimized via grid search. For each strategy, the following ranges were tested:

1. **Epsilon-Greedy (Fixed Step Size)**
    (a) $\epsilon$: $[0.01, 0.5]$ in increments of 0.01
    (b) stepSize: $[0.01, 0.5]$ in increments of 0.01
2. **Epsilon-Greedy**
    (a) $\epsilon$: $[0.01, 0.5]$ in increments of 0.01
3. **Optimistic**
    (a) initValue: $[0.0, 5.0]$ in increments of 1.0
4. **Gradient**
    (a) $\alpha$: $[0.1, 1.0]$ in increments of 0.1
5. **Proposed**
    (a) discount: $[0.0, 1.0]$ in increments of 0.1

The final parameters were selected by maximizing the average accumulated reward (AAR) across all simulations.

### 4.2 Performance Metrics

For stationary bandits, three main performance metrics were used:

1. **Average Accumulated Reward (AAR):** The average total rewards received by an agent across a simulation. This metric answers the question, 'how high is the reward?'
2. **Percent Optimal Action (POA):** The percentage of all actions taken (across simulations and timesteps) which were optimal. This metric answers the question, 'how optimal is the agent at choosing the best bandit?'
3. **Average Normalized Accumulated Reward (ANAR):** The total rewards across a simulation are normalized between 0 and 1 by comparing it to the worst possible total rewards across that simulation and the best possible total rewards across that simulation. This is then averaged across all simulations to produce ANAR. This statistic answers the question, 'how good is this agent compared to how good it could be?'

For non-stationary bandits, the average reward at the final timestep was used as the primary metric. Additional metrics from the stationary environment were computed for comparison.

### 4.3 Update Rules for Non-Stationary Environments

The following update rules were used to compute the mean of the non-stationary reward distributions at each timestep:

$$\begin{aligned} \text{Drifting: } & \mu_{t+1} = \mu_t + \epsilon \quad \text{(Where } \epsilon \sim N(0, 0.001^2)\text{)} \\ \text{Mean Reverting: } & \mu_{t+1} = \kappa \mu_t + \epsilon \quad \text{(Where } \kappa = 0.5 \text{ and } \epsilon \sim N(0, 0.01^2)\text{)} \\ \text{Abrupt: } & \mu_{t+1} = \mu_t \quad \text{(Where bandit means permute with probability } \epsilon = 0.005\text{)} \end{aligned} \tag{2}$$

# 5 Results

## 5.1 Stationary Bandits

The performance of each agent on stationary bandits is summarized in Table 1. The proposed method showed competitive performance, closely trailing the optimistic agent in AAR, POA, and ANAR.

| **Agent** | **AAR** | **POA** | **ANAR** |
|---|---|---|---|
| Greedy | 1019.226 | 36.791% | 0.762 |
| Epsilon-Greedy ($\epsilon = 0.1$) | 1292.591 | 67.513% | 0.823 |
| **Optimistic (initValue** $= 5.0$**)** | **1488.013** | **85.742%** | **0.866** |
| Gradient ($\alpha = 0.2$) | 1402.777 | 74.721% | 0.848 |
| Proposed (discount $= 1.0$) | 1447.838 | 81.424% | 0.857 |

Table 1: Performance on Stationary Bandits

The optimistic agent exhibits superior performance across all metrics in the stationary environment, closely followed by the proposed method. This gap is likely due to the optimistic agent's inherent bias towards exploration in the early timesteps and complete lack of exploration in the later stages. While this strategy is beneficial in stationary environments where initial approximate reward means maintain accuracy over time, this is not likely to succeed in nonstationary environments.

## 5.2 Non-Stationary Bandits

For non-stationary bandits, the proposed method demonstrated strong performance, particularly in environments with drifting means and abrupt changes. Figures 1-3 show the distribution of average terminal rewards across different settings.

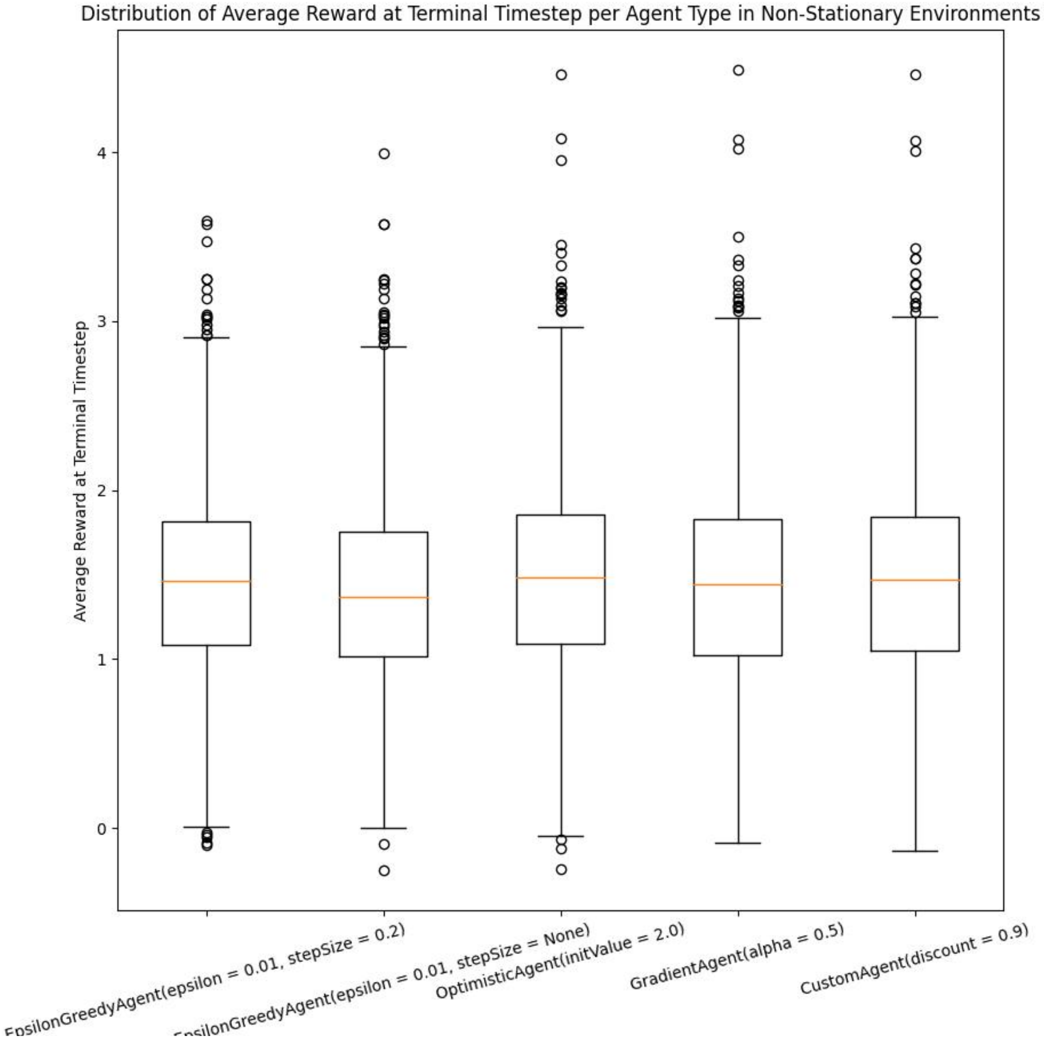

Figure 1: Distribution of average terminal rewards for Drifting mean environments.

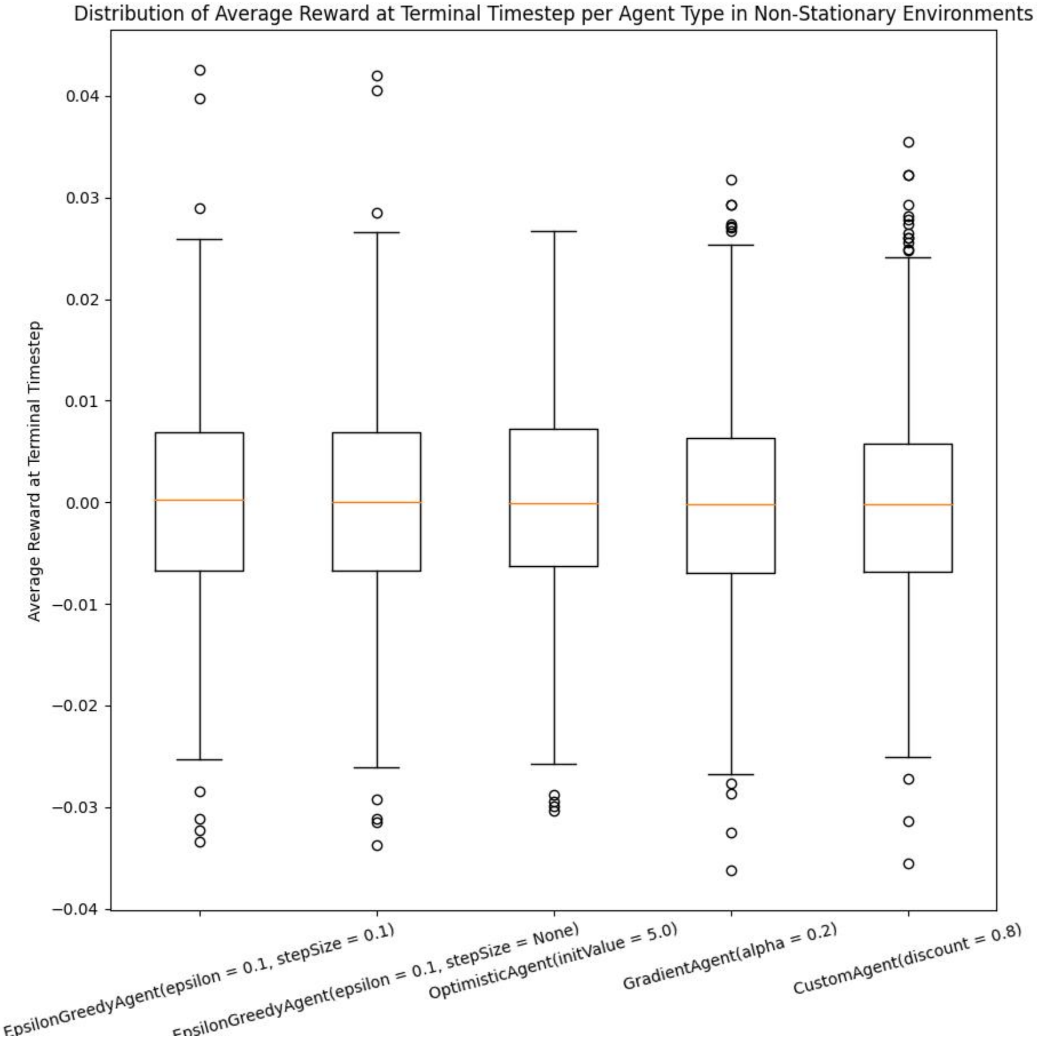

Figure 2: Distribution of average terminal rewards for Mean-Reverting environments.

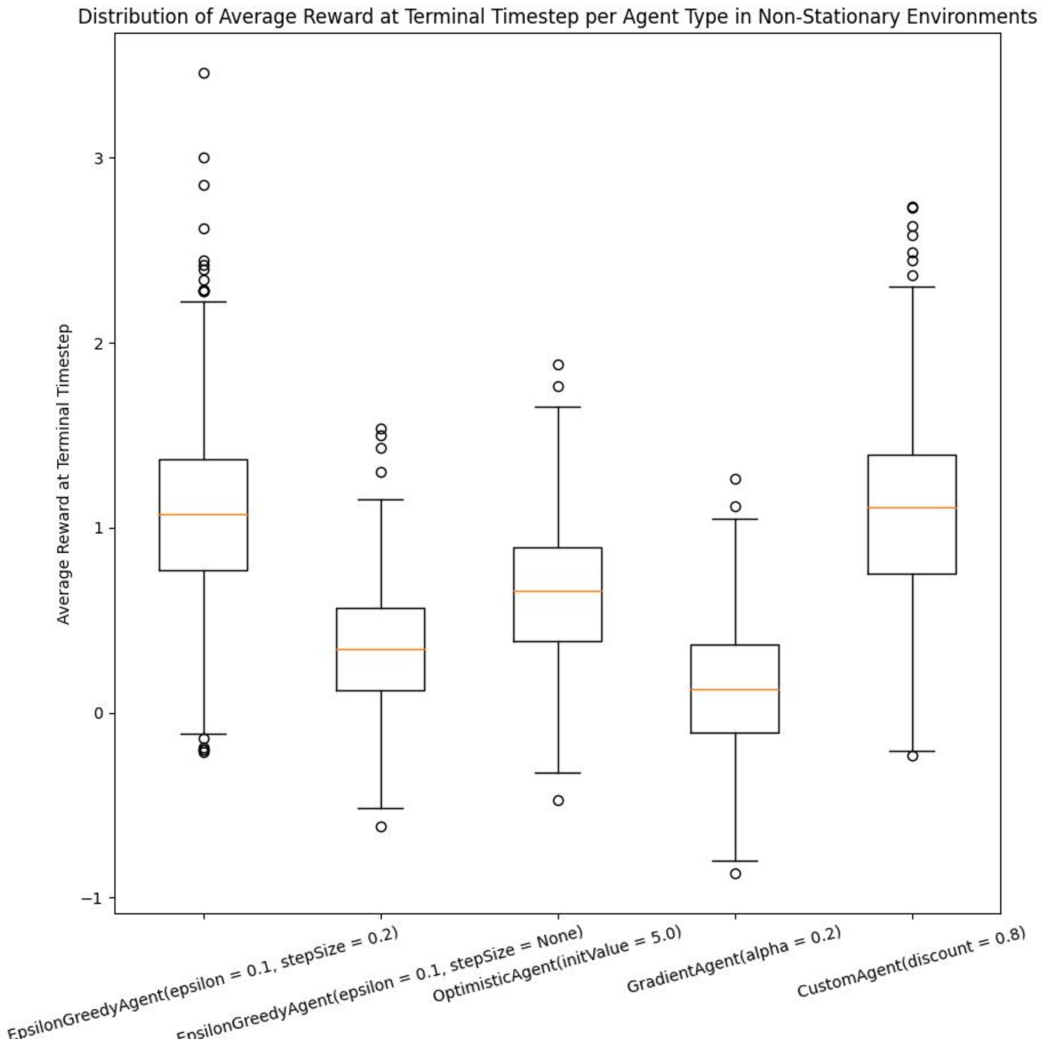

Figure 3: Distribution of average terminal rewards for Abruptly-Changing environments.

Contrary to our previous assumption, the optimistic agent continued to outperform most alternatives in both drifting and mean reverting environments. While this is not immediately apparent from the box plots above, the statistics in Appendix A are clear: the optimistic agent outperforms all other agents across all metrics in drifting environments and acquires the greatest AAR in mean-reverting environments. We hypothesize this consistent performance is likely due to a poor choice of initialization in drifting and mean-reverting environments. Increasing $\epsilon$ in drifting environments would make prediction more difficult. This would likely disperse the distribution of terminal rewards in our box plots and lower the rewards of the optimistic agent as initial approximations will no longer be accurate over time. Similarly, increasing $\kappa$ in mean-reverting environments - making prediction simpler - would likely increase the spread of terminal rewards, showing an increase in the performance of the other agents as they learn to adapt to the mean over time and leave the optimistic agent behind.

Unlike the other two non-stationary bandit environments, the abruptly changing box plots clearly demonstrate an advantage for epsilon greedy strategies with fixed step size as well as our proposed algorithm. We posit that this is due to their consistent exploration over time, whereas the other agents tend to favour exploitation increasingly over time.

In environments with abrupt changes, the Epsilon-Greedy Agent with fixed step size achieved the highest POA (40.089%) while our proposed method led in AAR ($10,891.695$) and ANAR (0.778). We introduced ANAR to better represent performance, showing that the proposed strategy is a strong competitor.

In mean-reverting environments, all agents performed similarly, with ANAR values of 0.500. This is further evidence that increasing the mean reversion coefficient $\kappa$ could improve the veracity of our results.

Overall, the Optimistic Agent led performance across all environments; however, our proposed method consistently ranked among the best, demonstrating its robustness and adaptability. Its finite hyperparameter space offers practical advantages over the infinite space of the optimistic agent.

A more complete set of statistics, tables, and figures can be found in Appendices A-C.

## 6 Discussion

These results indicate that the optimistic agent consistently outperforms most agents in both stationary and non-stationary environments, particularly in drifting settings. The proposed probabilistic method also performed strongly, demonstrating its adaptability and robustness across different bandit types.

The superior performance of the optimistic agent in finite timestep scenarios may be attributed to its inherent bias towards exploration in the early timesteps of the simulation due to optimistic initial values, which ensures sufficient exploration before convergence. However, this strategy will not likely scale well with longer durations or more complex environments.

The proposed method's reliance on probability calculations provides a more grounded approach to balancing exploration and exploitation *over time*, resulting in competitive performance. Additionally, its finite hyperparameter space makes it a potentially more practical alternative to infinite hyperparameter space agents like optimistic algorithms.

### 6.1 Limitations and Future Work

One limitation of the current study is the fixed nature of hyperparameters across different environments. Future work will explore adaptive hyperparameter tuning methods to enhance performance further. Additionally, testing the proposed method in more complex and real-world bandit problems, such as those with high-dimensional action spaces or contextual information, would provide deeper insights into its applicability and scalability.

Another area for future research is the integration of contextual information into the probabilistic framework, potentially extending the method to contextual bandit problems. This extension could significantly enhance the method's applicability in domains where additional information about the environment or actions is available.

Finally, perhaps the greatest limitation in our proposed solution is its inherent assumption of normally distributed rewards. While the Central Limit Theorem does assure the normal distribution of sample means regardless of the distribution of the original variables, evaluating performance in non-normally distributed environments is imperative.

## 7 Conclusion

This study introduced a novel probabilistic strategy for the k-armed bandit problem based on reward probability calculations. The proposed method demonstrated competitive performance in both stationary and non-stationary environments, making it a viable alternative to traditional strategies. By leveraging probability computations derived from observed rewards, the method effectively balances exploration and exploitation, leading to robust performance across different settings.

The findings suggest that the proposed method can serve as a strong competitor to existing strategies, particularly in environments with dynamic reward distributions. Its computational simplicity and finite hyperparameter space offer practical advantages, making it suitable for real-world applications.

Future research will focus on further optimization and adaptation to more complex bandit settings, including those with high-dimensional action spaces and contextual information. The integration of adaptive hyperparameter tuning methods and the extension to contextual bandit problems represent promising directions for enhancing the method's applicability and impact.

## References

[1] Vashist Avadhanula, Riccardo Colini Baldeschi, Stefano Leonardi, Karthik Abinav Sankararaman, and Okke Schrijvers. Stochastic bandits for multi-platform budget optimization in online advertising. In *Proceedings of the Web Conference 2021*, pages 2805–2817, 2021.

[2] Deepayan Chakrabarti, Ravi Kumar, Filip Radlinski, and Eli Upfal. Mortal multi-armed bandits. *Advances in neural information processing systems*, 21, 2008.

[3] Gerardo Duran-Martin, Aleyna Kara, and Kevin Murphy. Efficient online bayesian inference for neural bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6002–6021. PMLR, 2022.

[4] Jian Li. The k-armed bandit problem with multiple priors. *Journal of Mathematical Economics*, 80:22–38, 2019.

[5] Herbert Robbins. Some aspects of the sequential design of experiments. 1952.

[6] Sofía S Villar, Jack Bowden, and James Wason. Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 30(2):199, 2015.

# Appendices

## A Non-Stationary Tables

| **Agent** | **AAR** | **POA** | **ANAR** |
|---|---|---|---|
| Epsilon-Greedy ($\epsilon = 0.01, \text{stepSize} = 0.2$) | 14681.354 | 80.210% | 0.863 |
| Epsilon-Greedy ($\epsilon = 0.01$) | 14131.516 | 71.046% | 0.852 |
| **Optimistic (initValue** $= 2.0$**)** | **15107.843** | **84.983%** | **0.871** |
| Gradient ($\alpha = 0.5$) | 14625.242 | 76.057% | 0.860 |
| Proposed (discount $= 0.9$) | 14854.770 | 78.942% | 0.866 |

Table 2: Performance on Non-Stationary Bandits (Drifting). This table summarizes the performance metrics for different agents in a drifting mean environment. The proposed method shows strong performance.

| **Agent** | **AAR** | **POA** | **ANAR** |
|---|---|---|---|
| Epsilon-Greedy ($\epsilon = 0.1, \text{stepSize} = 0.1$) | 1.816 | 10.015% | 0.500 |
| **Epsilon-Greedy** ($\epsilon = 0.1$) | **1.736** | **10.763%** | **0.500** |
| **Optimistic (initValue** $= 5.0$**)** | **2.218** | **9.742%** | **0.500** |
| Gradient ($\alpha = 0.2$) | -1.106 | 10.085% | 0.500 |
| Proposed (discount $= 0.8$) | -1.894 | 9.984% | 0.500 |

Table 3: Performance on Non-Stationary Bandits (Mean Reverting). This table highlights the challenges faced by all agents in adapting to rapidly mean-reverting environments with ANAR values reflecting the difficulty of maintaining any advantage.

| **Agent** | **AAR** | **POA** | **ANAR** |
|---|---|---|---|
| **Epsilon-Greedy** ($\epsilon = 0.1,$ **stepSize** $= 0.2$) | **10789.521** | **48.089%** | **0.776** |
| Epsilon-Greedy ($\epsilon = 0.1$) | 3464.568 | 17.395% | 0.591 |
| Optimistic (initValue $= 5.0$) | 6429.296 | 27.274% | 0.668 |
| Gradient ($\alpha = 0.2$) | 1275.978 | 13.371% | 0.533 |
| **Proposed (discount** $= 0.8$**)** | **10891.695** | **45.225%** | **0.778** |

Table 4: Performance on Non-Stationary Bandits (Abrupt Changes). This table indicates that the proposed method performs particularly well in environments with abrupt changes, reflecting its adaptability to sudden shifts in reward distributions.

## B Stationary Figures

The dashed red line in each figure represents the mean across all simulations and timesteps.

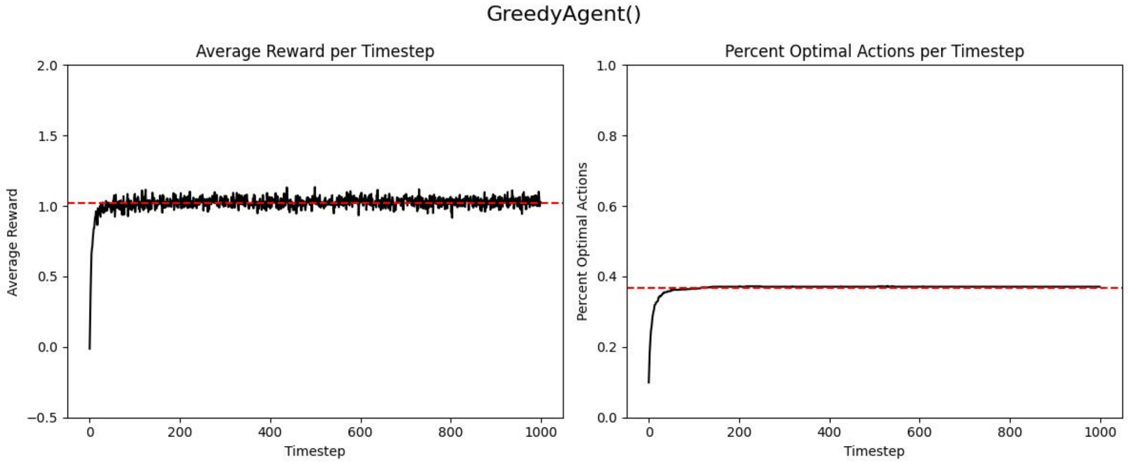

Figure 4: Performance of Greedy Agent over Timesteps in Stationary Environment. The Greedy Agent shows rapid convergence but to a suboptimal action due to lack of exploration.

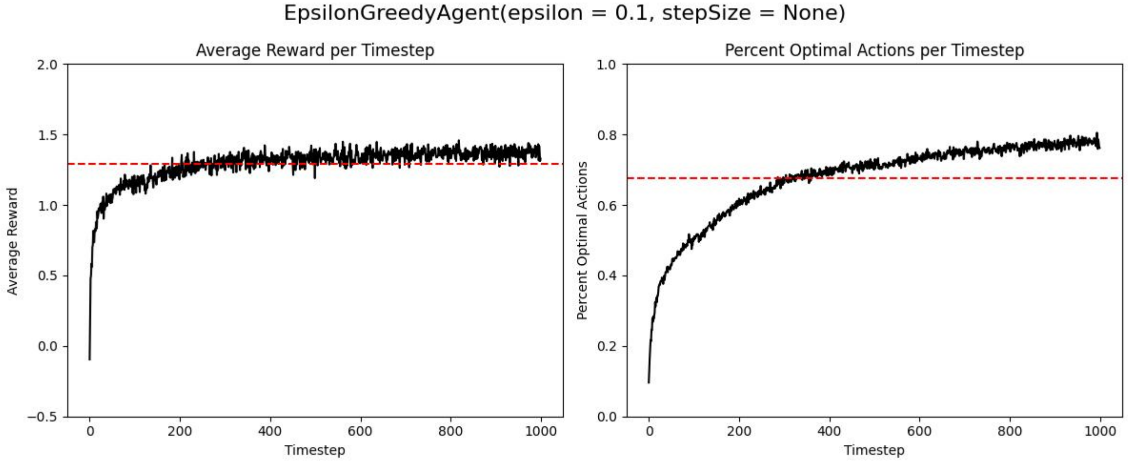

Figure 5: Performance of Epsilon-Greedy Agent ($\epsilon = 0.1$) over Timesteps in Stationary Environment. The Epsilon-Greedy Agent balances exploration and exploitation, resulting in higher overall rewards compared to the Greedy Agent.

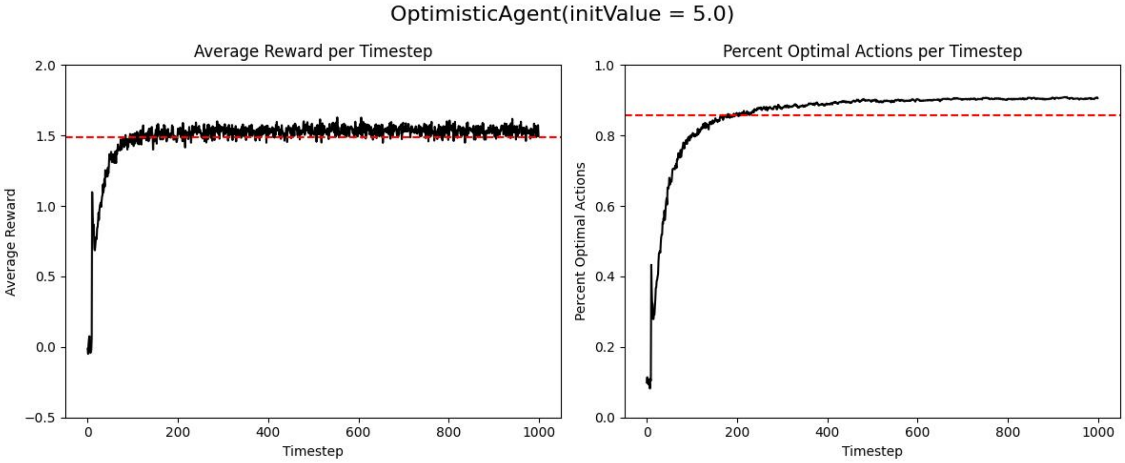

Figure 6: Performance of Optimistic Agent (initValue = 5.0) over Timesteps in Stationary Environment. The Optimistic Agent shows robust performance due to initial high reward estimates promoting exploration.

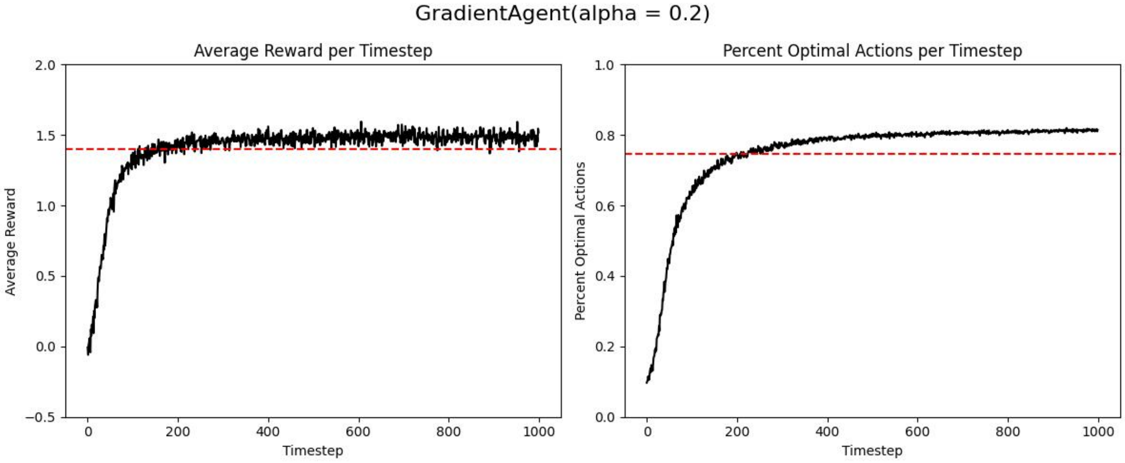

Figure 7: Performance of Gradient Agent ($\alpha = 0.2$) over Timesteps in Stationary Environment. The Gradient Agent uses preference-based learning to achieve competitive performance.

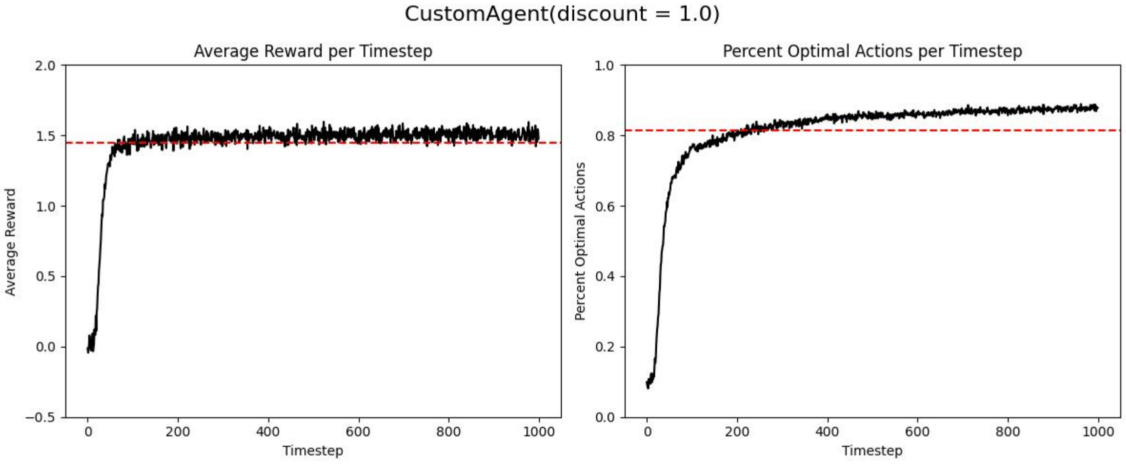

Figure 8: Performance of Proposed Agent (discount = 1.0) over Timesteps in Stationary Environment. The Proposed Agent demonstrates effective exploration-exploitation balance, quickly achieving high rewards and consistently improving over time.

# C Non-Stationary Figures

The dashed red line in each figure represents the mean across all simulations and timesteps.

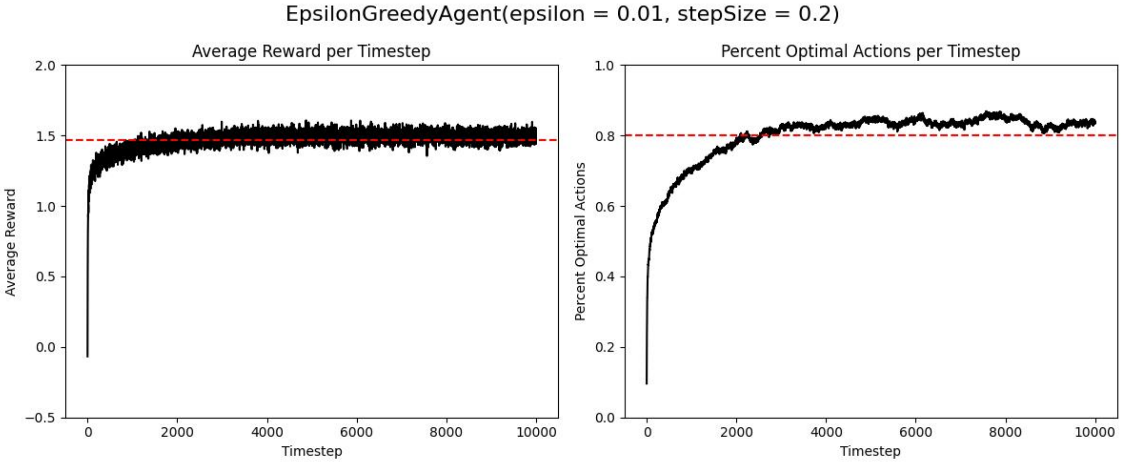

Figure 9: Performance of Epsilon-Greedy Agent ($\epsilon = 0.01, \text{stepSize} = 0.2$) over Timesteps in Drifting Environment. This shows effective adaptation to slowly changing reward distributions.

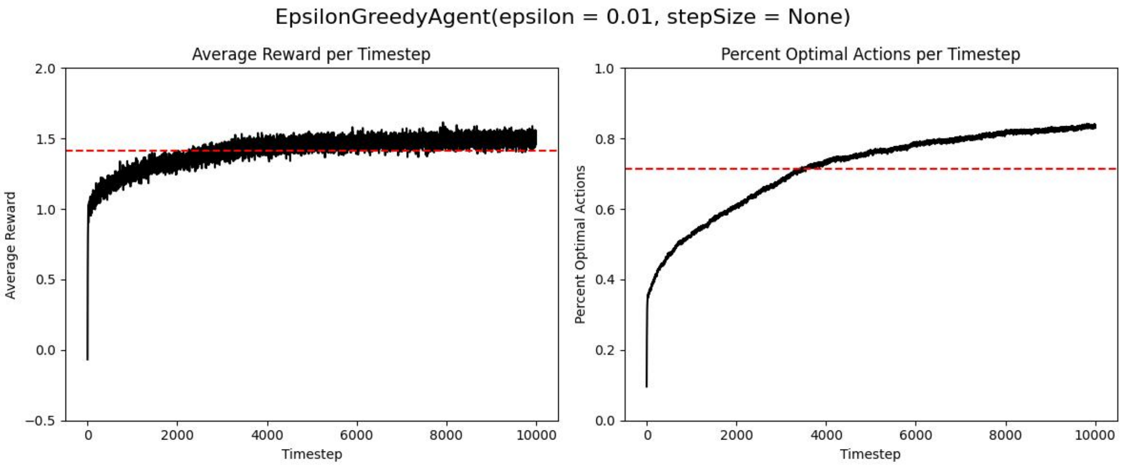

Figure 10: Performance of Epsilon-Greedy Agent ($\epsilon = 0.01$) over Timesteps in Drifting Environment. This illustrates the impact of step size on adaptation speed.

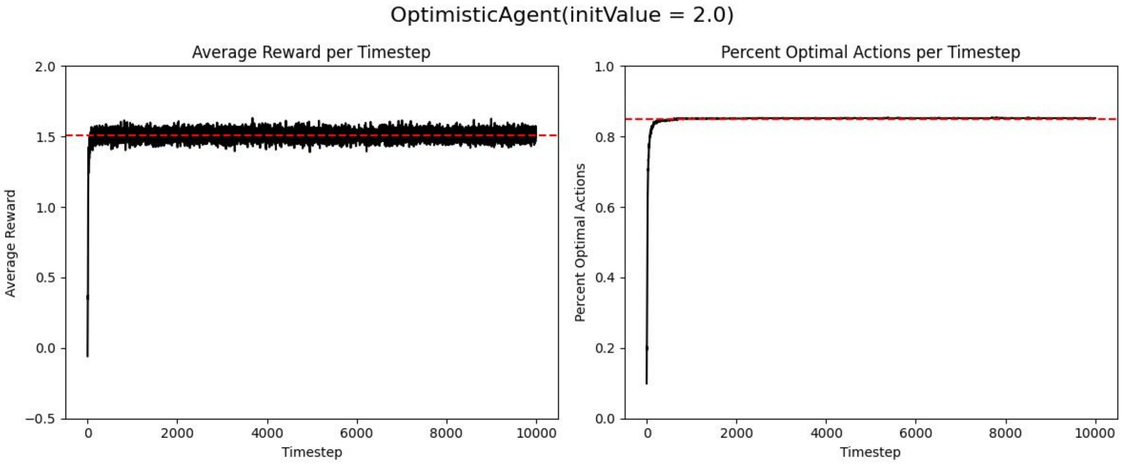

Figure 11: Performance of Optimistic Agent (initValue = 2.0) over Timesteps in Drifting Environment. The optimistic initial values promote exploration, which induces fast convergence.

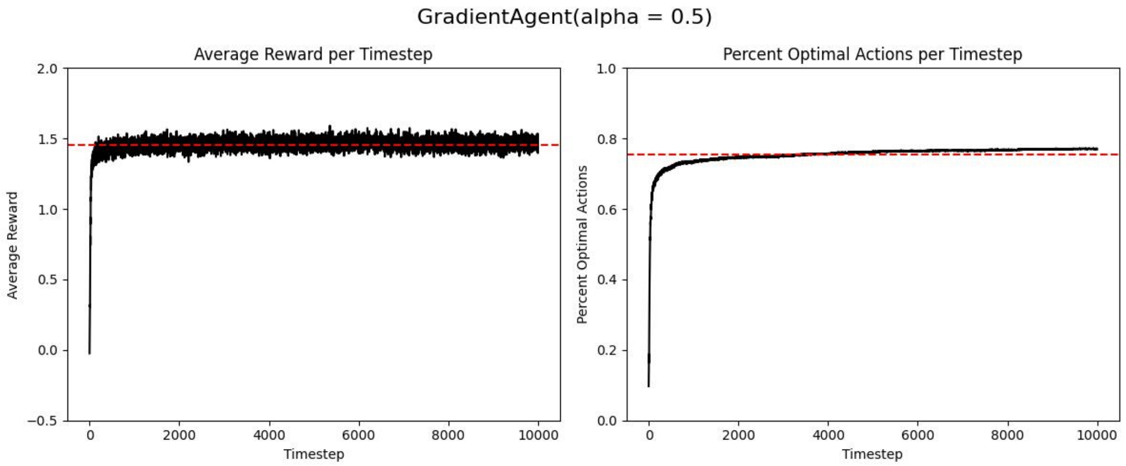

Figure 12: Performance of Gradient Agent ($\alpha = 0.5$) over Timesteps in Drifting Environment. This demonstrates preference-based learning in a non-stationary setting.

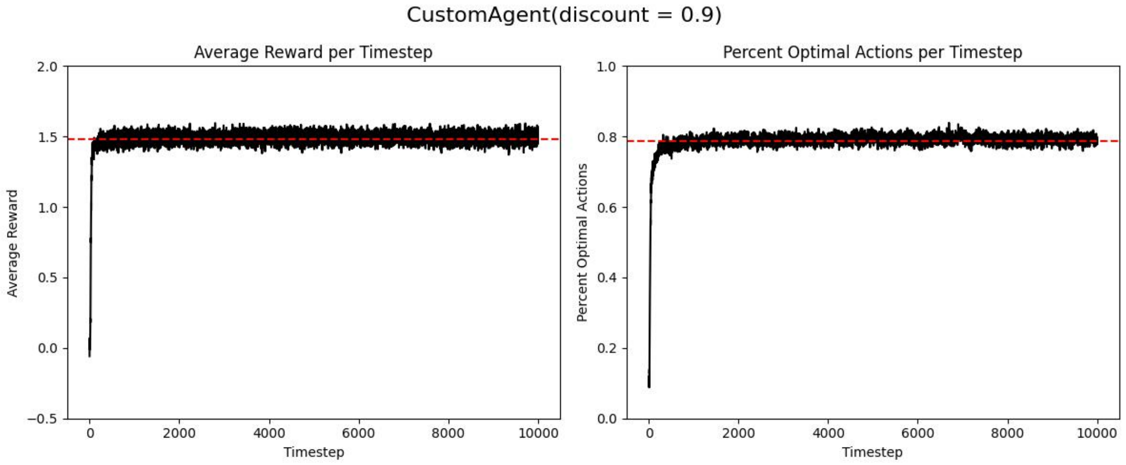

Figure 13: Performance of Proposed Agent (discount = 0.9) over Timesteps in Drifting Environment. This shows strong adaptability to drifting reward means.

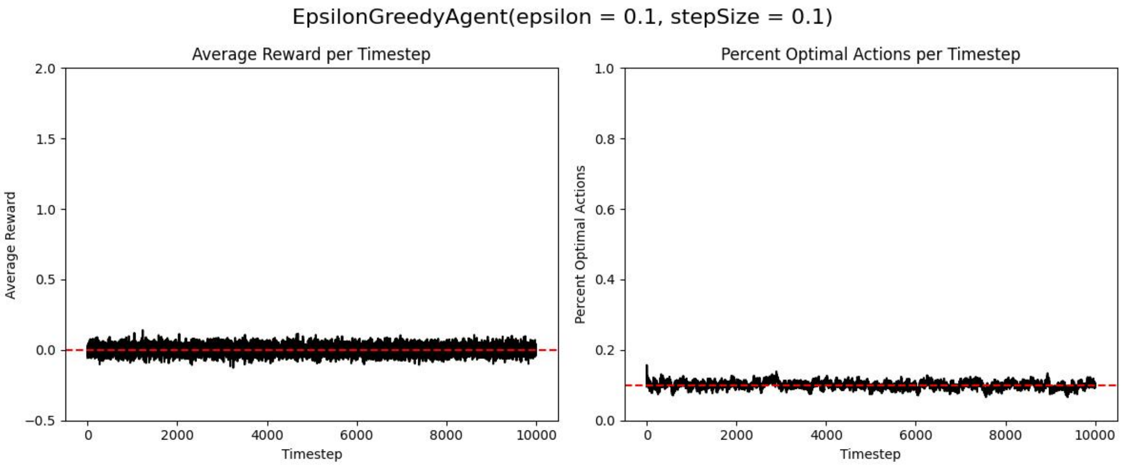

Figure 14: Performance of Epsilon-Greedy Agent ($\epsilon = 0.1, \text{stepSize} = 0.1$) over Timesteps in Mean-Reverting Environment. This shows the agent's difficulty in adapting to rapidly reverting means.

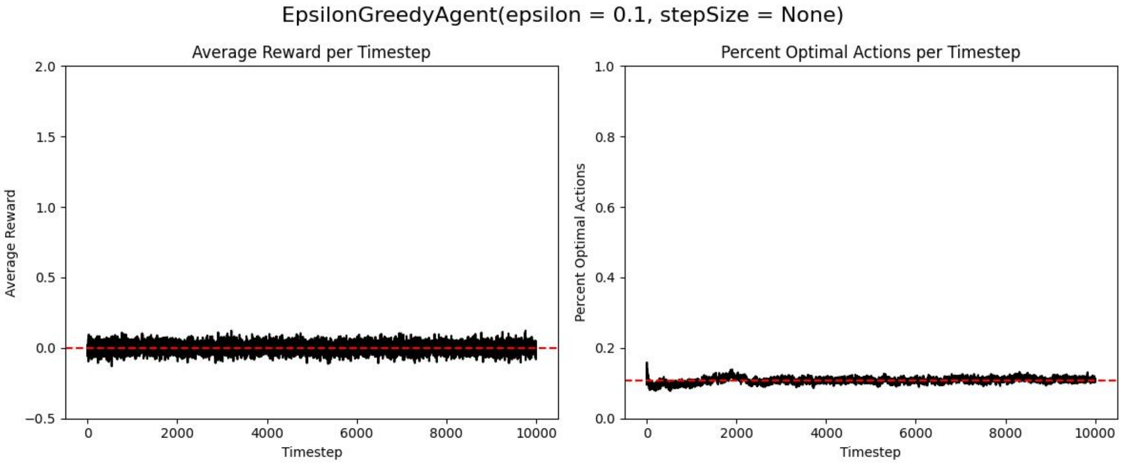

Figure 15: Performance of Epsilon-Greedy Agent ($\epsilon = 0.1$) over Timesteps in Mean-Reverting Environment. This highlights the need for frequent exploration in rapidly changing settings.

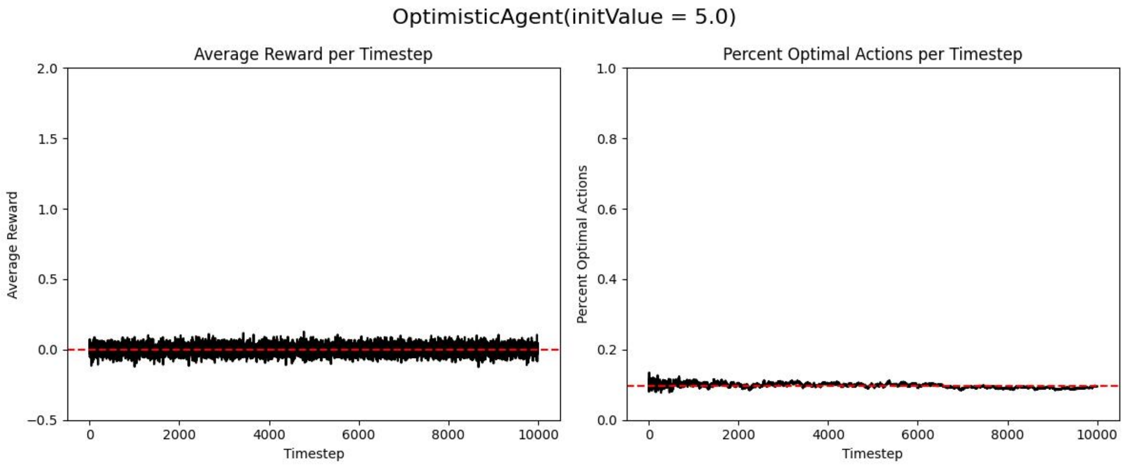

Figure 16: Performance of Optimistic Agent (initValue = 5.0) over Timesteps in Mean-Reverting Environment. The high initial values do not appear to aid convergence in mean-reverting environments.

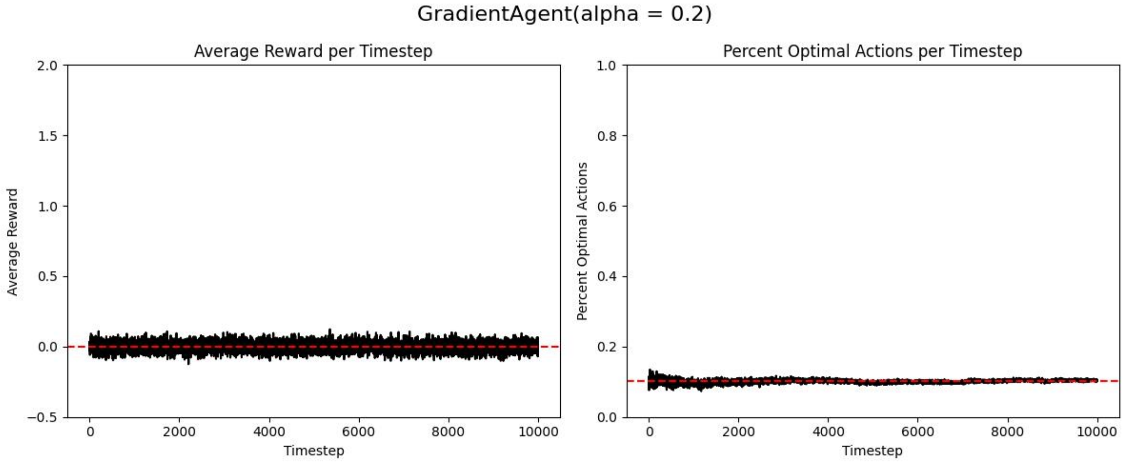

Figure 17: Performance of Gradient Agent ($\alpha = 0.2$) over Timesteps in Mean-Reverting Environment.

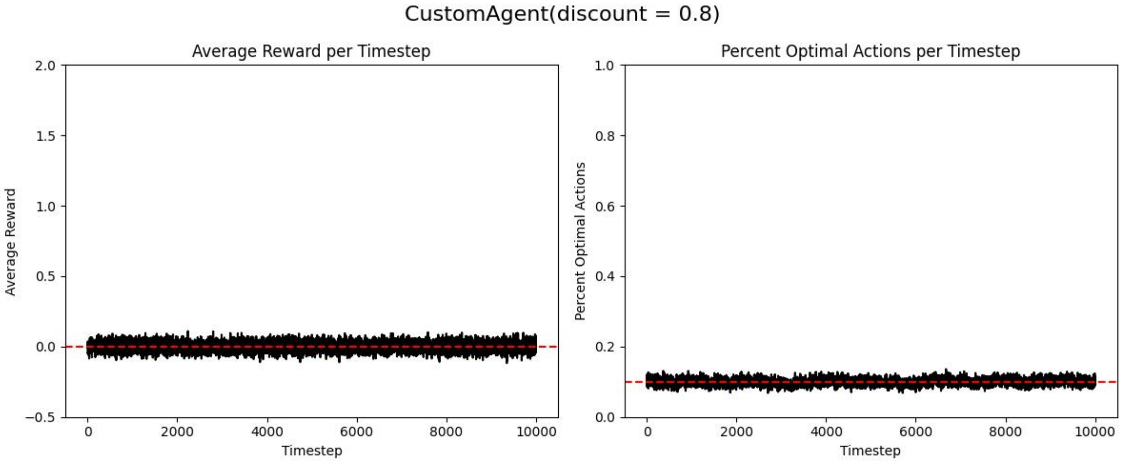

Figure 18: Performance of Proposed Agent (discount = 0.8) over Timesteps in Mean-Reverting Environment. No superior ability to adapt in mean-reverting environments is demonstrated.

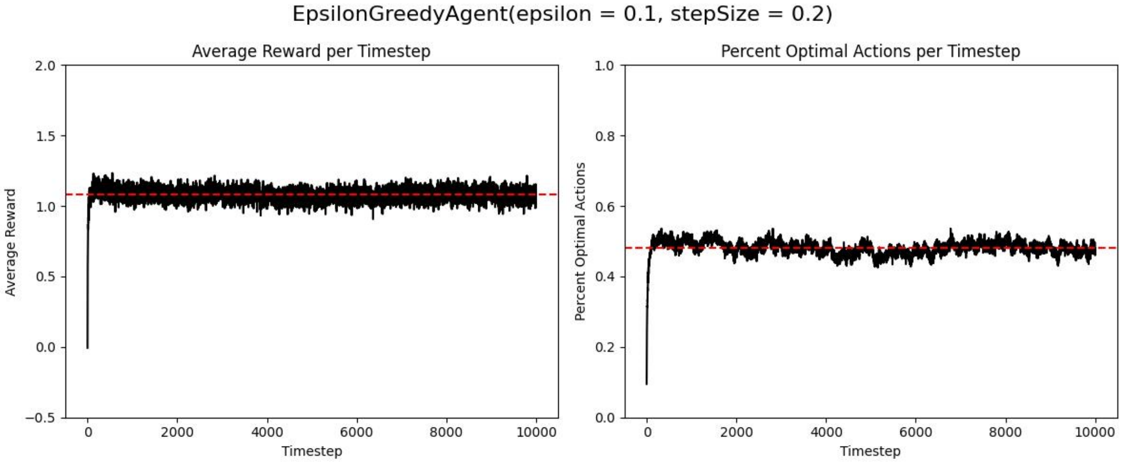

Figure 19: Performance of Epsilon-Greedy Agent ($\epsilon = 0.1, \text{stepSize} = 0.2$) over Timesteps in Abrupt Change Environment. This shows how exploration helps in environments with sudden changes.

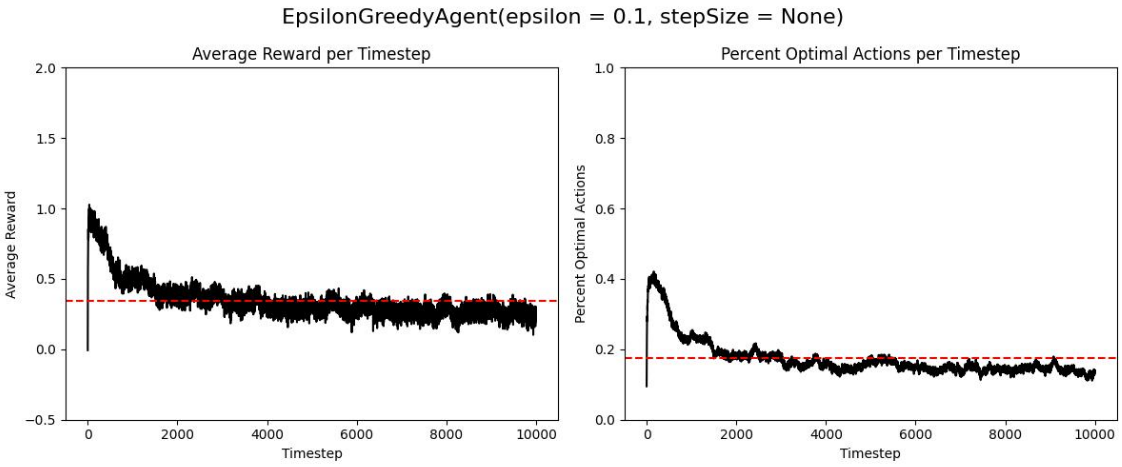

Figure 20: Performance of Epsilon-Greedy Agent ($\epsilon = 0.1$) over Timesteps in Abrupt Change Environment. This illustrates the need for constant stepsizes in abruptly changing environments.

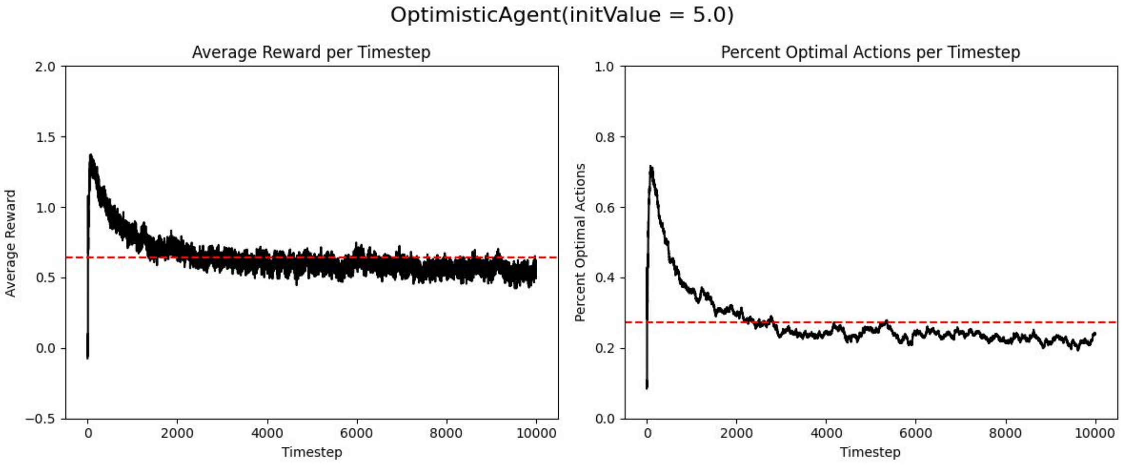

Figure 21: Performance of Optimistic Agent (initValue = 5.0) over Timesteps in Abrupt Change Environment. This demonstrates the risk of initial convergence.

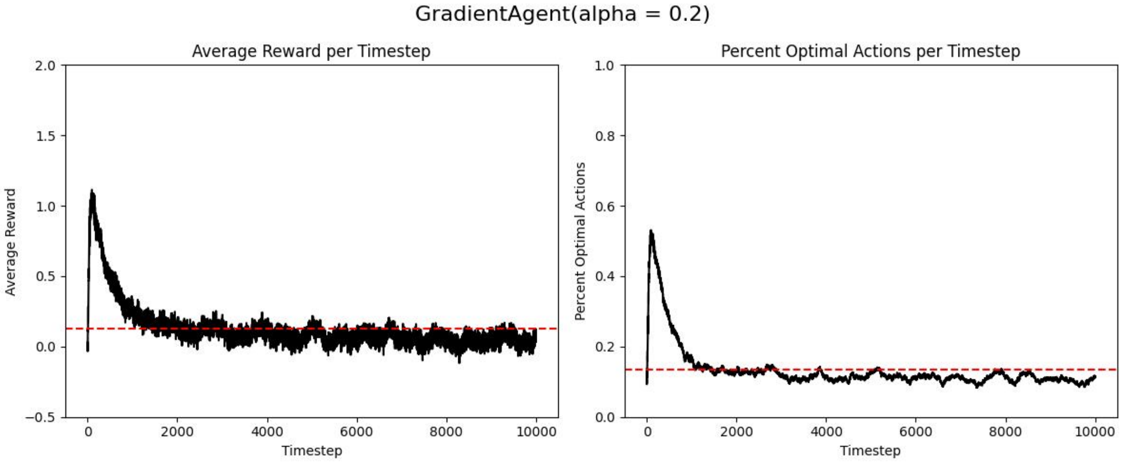

Figure 22: Performance of Gradient Agent ($\alpha = 0.2$) over Timesteps in Abrupt Change Environment. This demonstrates how preference-based algorithms fail to adapt to abrupt changes, similar to optimistic agents.

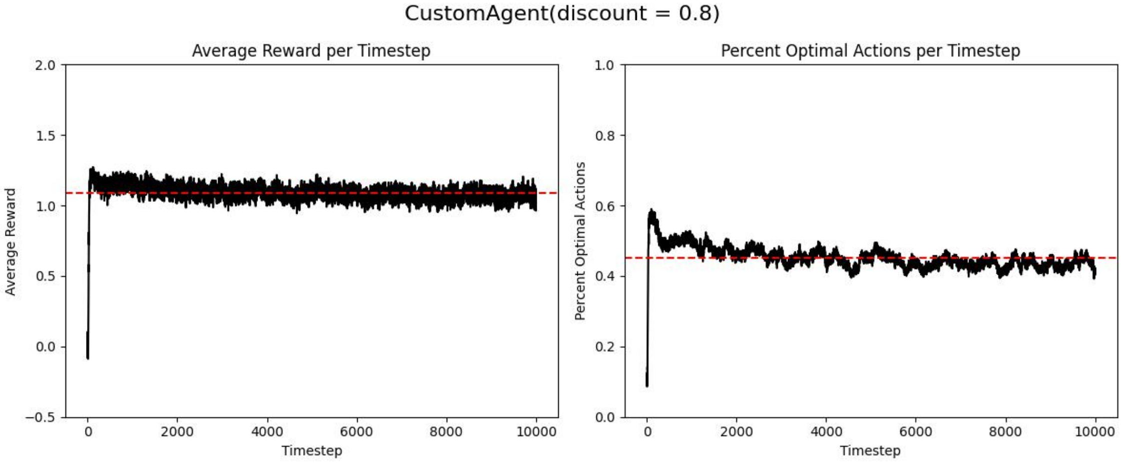

Figure 23: Performance of Proposed Agent (discount = 0.8) over Timesteps in Abrupt Change Environment. This shows great adaptability of the proposed method in environments with sudden changes.