Approaches to non-stationarity for multi-arm bandits

Nicholas Barnfield
Department of Mathematics and Statistics
McGill University
Montreal, QC H3A 0G4
nicholas.barnfield@mail.mcgill.ca

Kevin Zhao
Department of Mathematics and Statistics
McGill University
Montreal, QC H3A 0G4
tongke.zhao@mail.mcgill.ca

Mark-Anthony Moisescu-Pareja
Department of Mathematics and Statistics
McGill University
Montreal, QC H3A 0G4
mark-anthony.moisescu-pareja@mail.mcgill.ca

Abstract

In this project we compare the performance of various multi-arm bandit algorithms in different non-stationary settings. Specifically, we evaluate empirically the performance of the EXP3, ϵ-greedy, and Thompson-Sampling algorithms, as well as the discounted and standard variations of UCB. We consider non-stationary environments in the 3-arm bandit scenario whereby the reward distributions of each arm undergo stationary phases interrupted by sudden distribution-shifts. Through experiments, we discuss which forms of non-stationarity are favoured by the aforementioned algorithms. All team members contributed equally.

1 Introduction

The multi-arm bandit framework is the canonical example of the exploration versus exploitation trade-off in reinforcement learning. In the basic setting, an agent has a choice of "pulling" (sampling) one of a set of k "arms" for each time $t = 1, \dots, T$, where the action of pulling a given arm j provides the agent with a reward r sampled from a distributed f_j corresponding to the respective chosen arm. Due to its simple framework where no states are involved[1] and the reward distribution of the arms are taken to be independent of time, the multi-arm bandit problem has been widely studied in the theoretical literature, beginning in the early 1930s [Tho33] and re-surging in the 21^{st}-century.

Despite its simplistic framework amongst the growing popularity of ever-more complicated reinforcement learning architectures, multi-arm bandits have diverse applications in areas ranging from online advertising to clinical trials. For instance, a simple example of the latter can be seen in choosing which online advertisement to present a user visiting a website. Motivated by the less mathematically robust, but more realistic scenario at play in many real-world applications, we are interested in the performance of state of the art multi-arm bandit algorithms where the reward distributions of the arms change with time t. Returning to the former example, in reality a user's preferences ("reward distribution") will change and so in tune, the types of ads presented to the user should adapt as well.

Herein, we provide a comprehensive empirical evaluation of the EXP3, ϵ-greedy, and Thompson-Sampling algorithms, and the discounted and standard variations of UCB in a non-stationary setting of sudden distribution shifts; see Section 4 for precise details and Section 7 for pseudocode. While non-stationarity in the multi-arm bandit problem has been extensively studied, to our knowledge

[1] For a survey of the more complex contextual bandits scheme involving multiple states see [LPP10]

Preprint. Under review.

there is no survey comparison of such a wide-variety of algorithms for the sudden-shift regime in the literature.

2 Overview of Algorithms

A natural improvement to random selection, the easily-implemented ϵ-greedy algorithm [SB18, §2] is guaranteed to converge to sub-optimal solutions in stationary environment when ϵ is fixed. However, in a non-stationary regime, this fixed exploration rate may prove beneficial. The more sophisticated Thompson Sampling and UCB algorithms [RVRK18, ACBF02], which have sublinear regret bounds and converge to optimality in under some stationary environment, will almost assuredly lose these desired properties in the sudden-shift environments. Given Gaussian bandit setting (see Section 4) studied here, we take Gaussian distributions of fixed variance and mean zero for the priors in the Thompson Sampling algorithm and use a Bayesian framework to update the mean of the posterior distribution (see Section 7 for precise details). Not seen in the course, the EXP3 algorithm [ACBFS02] probabilistically selects actions using a categorical distribution that is weighted according to the observed rewards. The Discounted UCB algorithm represents an extension of the traditional UCB algorithm, incorporating a discount factor to manage exploration and exploitation in dynamic environments where rewards may undergo temporal changes. Similarly, the Sliding Window UCB algorithm, a modification of UCB, restricts the scope of historical data utilized for decision-making to a defined window size, facilitating adaptation to evolving environments while preserving computational efficiency. Conversely, the Bayes UCB algorithm [KCG12] capitalizes on Bayesian updating techniques to estimate the underlying reward distribution for each action, thereby facilitating more informed decision-making in multi-armed bandit scenarios through a nuanced balance of exploration and exploitation.

3 Related Work

In the works of Allesiardo et al. [AFM17] and Cavenaghi et al. [CSSZ21], the authors introduce a novel algorithm that dynamically adjusts exploration and exploitation strategies to effectively balance between exploiting current knowledge and exploring new options as the environment evolves. Non-stationarity is incorporated through mechanisms such as tracking changes in reward distributions and adapting exploration-exploitation trade-offs accordingly, ensuring adaptability to changing conditions.

4 Methodology

We consider 4 different non-stationary regimes of sudden distribution shifts characterized by the frequency and magnitude of the shift. For each regime, we further consider lower and higher variability settings to examine how the variance in a given arm's reward distribution affects the behaviour of the different algorithms. We characterize the frequency of changes by either being in the low frequency (LF) or high frequency regime (HF), in which the distribution of the arms will change every 100 or 200 time steps respectively. Letting $R_i \sim \mathcal{N}(a_i, \sigma)$ be the instantaneous reward distribution for arm $i \in \{1, 2, 3\}$, in the low magnitude (LM) regime, we begin each run with

$$a_1 = -a_3 = 5, \qquad a_2 = 0$$

whereas in the high magnitude (HM) regime, we use

$$a_1 = -a_3 = 10, \qquad a_2 = 0.$$

Namely, in each case, the first arm will yield the best expected reward for the agent with the expected difference in reward between the best and worst arm being twice as large in the HM regime. At each shift in the reward distribution, the mean reward of arm 1 and 3 swap while a_2 remains constant. By symmetry of the means a_1 and a_3, given the swap $a_1 \rightleftarrows a_3$ occurs an odd number of times, any *fixed* randomized strategy will yield an expected reward of 0. This provides a good baseline of comparison to a "blind" strategy, namely one in which no learning takes place such as when the sampled rewards are not observable. We further consider two variance configurations for the arms: a high variance (HV) setting where $\sigma = 15$ and a low variance (LV) setting with $\sigma = 5$. This yields a total of 8 non-stationary regimes.

In our incorporation of non-stationarity by swapping the mean for the reward distribution of the best and worst arms, it is plausible for "intelligent" algorithms — i.e. one's that leverage observed rewards — to perform worst than a blind agent. For instance, an algorithm that is slow to converge, but that is persistent in it's greedy choice, may only settle on the best arm after a swap has occured — leading to suboptimal choices during consecutive stationary phases. This leads to an interesting setting for contrasting the robustness of the multi-arm bandit strategies as well as easily interpretable results.

For each environment and each of the 6 algorithms discussed in Section 2, we first perform hyper-parameter tuning[2] –maximizing the mean cumulative reward taken over 100 runs of 2000 time steps. Then, for each non-stationary regime, we compare the cumulative reward of the best configuration of each algorithm using the same Monte Carlo procedure.

5 Experiments and Analysis

As explained above, we will categorize our results into 8 different regimes. The results are shown in figure 1. It is easy to observe that in general, all variations of the UCB algorithms performed significantly better in general both in the sense of total reward and linearity despite of the non-stationary environment. Interestingly, when comparing the EXP3, ϵ-greedy and Thompson-Sampling algorithms, we observe that the Thompson-Sampling algorithm initially performs the best, however, after reaching a certain time step, we have that the ϵ-greedy algorithm performs the best.

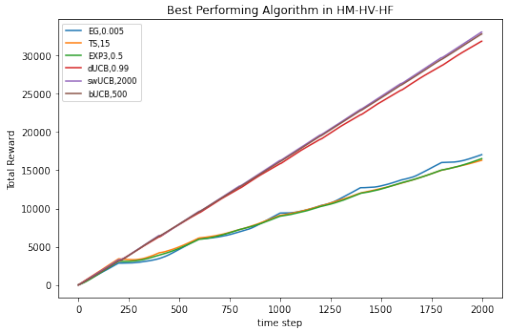
In the HM-HV-HF regime, figure 1(a) shows that the Sliding Window UCB algorithm with a window size of 2000 and $c = 20$ (parameter which determines the degree of exploration) performed the best. In addition, we see that the Thompson-Sampling algorithm with $\sigma = 15$ performed the worst. In the HM-HV-LF regime, figure 1(b) shows that the Sliding Window UCB algorithm with a window size of 2000 and $c = 15$ performed the best. Furthermore, we see that the Thompson-Sampling algorithm with $\sigma = 15$ performed the worst. In the HM-LV-HF regime, figure 1(c) shows that the Sliding Window UCB algorithm with a window size of 2000 and $c = 25$ performed the best. In addition, we see that the EXP3 algorithm with $\epsilon = 0.01$ performed the worst. In the HM-LV-LF regime, figure 1(d) shows that Bayes UCB algorithm with num of initial observations $= 300$ and $c = 10$ performed the best. Furthermore, we see that the Thompson-Sampling algorithm with $\sigma = 10$ performed the worst. In the LM-HV-HF regime, figure 1(e) shows that the Sliding Window UCB algorithm with a window size of 2000 and $c = 20$ performed the best. In addition, we see that the Thompson-Sampling algorithm with $\sigma = 15$ performed the worst. In the LM-HV-LF regime, figure 1(f) shows that the Sliding Window UCB algorithm with a window size of 2000 and $c = 20$ performed the best. Furthermore, we see that the Thompson-Sampling algorithm with $\sigma = 15$ performed the worst. In the LM-LV-HF regime, figure 1(g) shows that the Sliding Window UCB algorithm with a window size of 2000 and $c = 12$ performed the best. In addition, we see that the EXP3 algorithm with $\gamma = 0.5$ performed the worst. In the LM-LV-LF regime, figure 1(h) shows that the Bayes UCB algorithm with num of initial observations $= 500$ and $c = 5$ performed the best. Furthermore, we see that the Thompson-Sampling algorithm with $\sigma = 10$ performed the worst. Interestingly, for the Discounted UCB algorithm, Sliding Window UCB algorithm and Bayes UCB algorithm, for values of c larger than c_{optimal}, we have that the total reward starts to decrease.

Moreover, the three variation of the UCB algorithms are able to achieve a linear cumulative reward to certain extent despite the non-linear environment which shows how well they reacts in the non-stationary environment that was set up. On the other side, the other three algorithms(Epsilon-Greedy, Thompson Sampling and EXP3) responded poorly in comparison. They are unable to achieve linearity at all and the change in environment has a relatively big effect on the performance. It is interesting to note that the two algorithms (Thompson Sampling and Epsilon Greedy) that performed well in the stationary beta distribution bandit that we did in A1 performed badly.

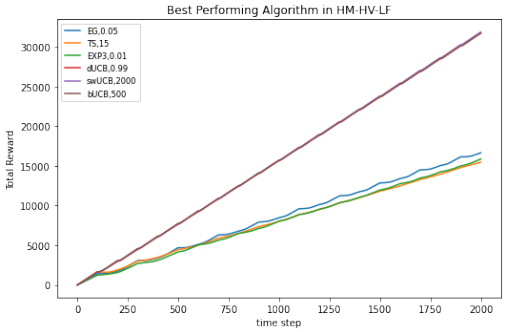
6 Conclusion

We provided a comparison of a wide-array of algorithms in the setting of periodic, sudden shifts in the reward distribution of Gaussian multi-arm bandits. While more contrived examples of non-stationarity exist, the concrete interpretation of non-stationarity used herein lends easily to a theoretical study in

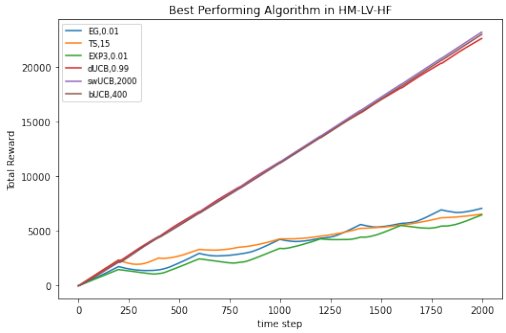
[2]See the figures in Section 7 for an extensive hyper-parameter evaluation of each algorithm and regime.



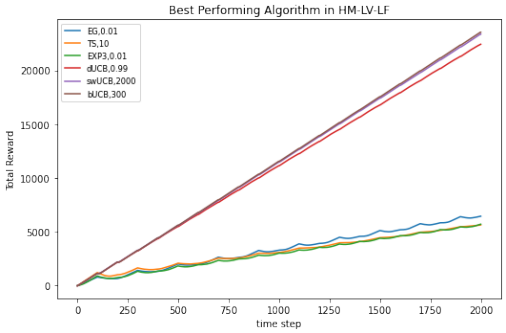
(a) HM-HV-HF



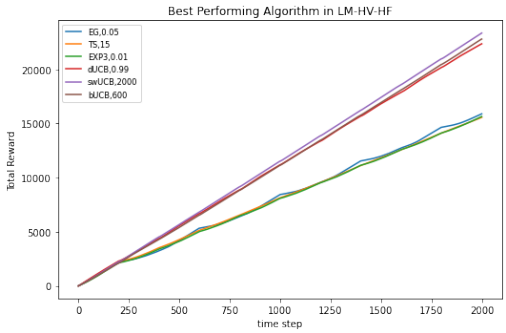
(b) HM-HV-LF



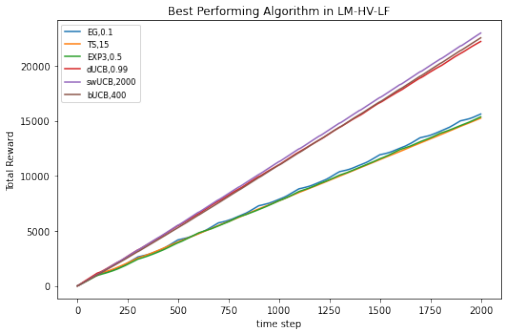
(c) HM-LV-HF



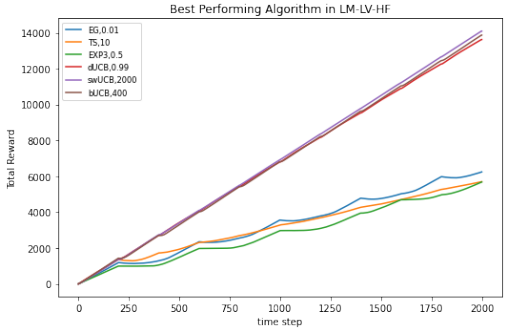
(d) HM-LV-LF



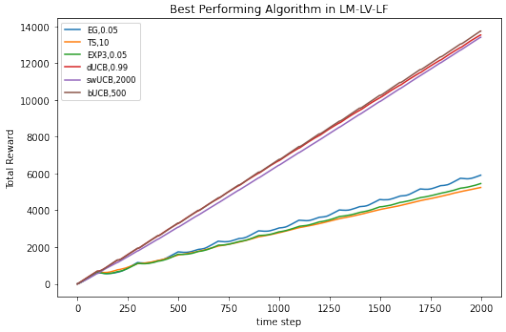
(e) LM-HV-HF



(f) LM-HV-LF



(g) LM-LV-HF



(h) LM-LV-LF

Figure 1: Results for different regimes.

explicating the empirical results — leading to interesting new research directions in the mathematical underpinnings of multi-arm bandits.

Through performing extensive simulations including hyper-parameter tuning, we find that the three variations of the UCBs is the best in almost all of the regime of changes in reward distribution, whereas, the other three algorithms performed poorly in comparison. This led some thoughts on possible future works, including testing or developing new algorithms and constructing more intense environments to better mimic real-life situations. More specifically, research can be done on

- Enhancing the adaptability of algorithms in rapidly changing environments, future work could explore the development of self-tuning algorithms that can automatically adjust their hyper parameters in real-time
- The evaluation of these algorithms across a wider array of non-stationary environments, particularly those that closely simulate complex real-world dynamics, such as financial markets, autonomous vehicle control systems, or adaptive network traffic routing

7 Supplementary Material

In this section, we will also be providing the performance in each regime when the hyper-parameters are not tuned. For example, for the variation of UCB algorithms, we used $c = 2$ as a baseline experiment as it performed very well for the stationary settings in Assignment 1. Based on these tuning plots below, we can see the non-stationarity can be seen in the curves, where for the average instantaneous reward received over time. some of the algorithms performs similarly to a random action selection and it's very sensitive to the change of environment. Similarly in the average cumulative reward received over time, we are able to see a wave like curve where it clearly shows that the algorithms are reacting to the non-stationary environment. But first, let start off with the pseudo-codes for the algorithms implemented.

Algorithm 1 ϵ-Greedy Algorithm

Initialize Q-values and action counts: $Q(a) \leftarrow 0$, $N(a) \leftarrow 0$ for all actions a
for $t = 1, 2, \ldots, T$ **do**
    With probability ϵ, choose $A_t \sim$ Uniform$(1, K)$
    Otherwise, choose $A_t \leftarrow \arg\max_a Q(a)$
    Take action A_t and observe reward R_t
    $N(A_t) \leftarrow N(A_t) + 1$
    $Q(A_t) \leftarrow Q(A_t) + \frac{1}{N(A_t)}(R_t - Q(A_t))$
end for

Algorithm 2 EXP3 Algorithm (with explore parameter $\gamma > 0$)

Initialize weight vector: $w_t(a) \leftarrow 1$ for all arms a
Initialize learning rate: $\eta > 0$
for $t = 1, 2, \ldots, T$ **do**
    $p_t(a) = (1 - \gamma)\frac{w_t(a)}{\sum_{b=1}^{K} w_t(b)} + \frac{\gamma}{K}$, for all arms a
    Sample $A_t \sim$ Categorical(p_t)
    Take action A_t and observe reward R_t
    $w_{t+1}(A_t) \leftarrow w_t(A_t) \times \exp(\gamma \cdot \frac{R_t}{K \cdot p_t(A_t)})$
end for

Algorithm 3 Thompson Sampling Algorithm (prior variance σ)

Initialize $N(a) = 1$ and $\mu(a) = 0$ for all arms a
for $t = 1, 2, \ldots, T$ **do**
    for each arm a **do**
        Sample from the posterior distribution: $\theta_{t,i} \sim \mathcal{N}(\mu(a), \sigma)$
    end for
    $A_t \leftarrow \arg\max_a \theta_{t,a}$
    Take action A_t and observe reward R_t
    $N(A_t) \leftarrow N(A_t) + 1$
    $\mu(A_t) \leftarrow \mu(A_t) \cdot \frac{N(A_t)-1}{N(A_t)} + \frac{R_t}{N(A_t)}$
end for

Algorithm 4 Discounted UCB Algorithm (with discount factor γ)

Initialize $N_i = 0$, $Q_i = 0$ for all arms i and a confidence parameter c
for each round $t = 1, 2, \ldots, T$ **do**
    for each arm i **do**
        if $N_i > 0$ **then**
            Compute $UCB_i = Q_i + c\sqrt{\frac{\log t}{N_i}}$
        else
            Set UCB_i to a value that ensures arm i is selected
        end if
    end for
    Select arm $I_t = \arg\max_i UCB_i$
    Observe reward r_t from arm I_t
    Update $N_{I_t} = \gamma N_{I_t} + 1$
    Update $Q_{I_t} = Q_{I_t} + \frac{1}{N_{I_t}}(r_t - Q_{I_t})$
end for

Algorithm 5 Sliding Window UCB Algorithm

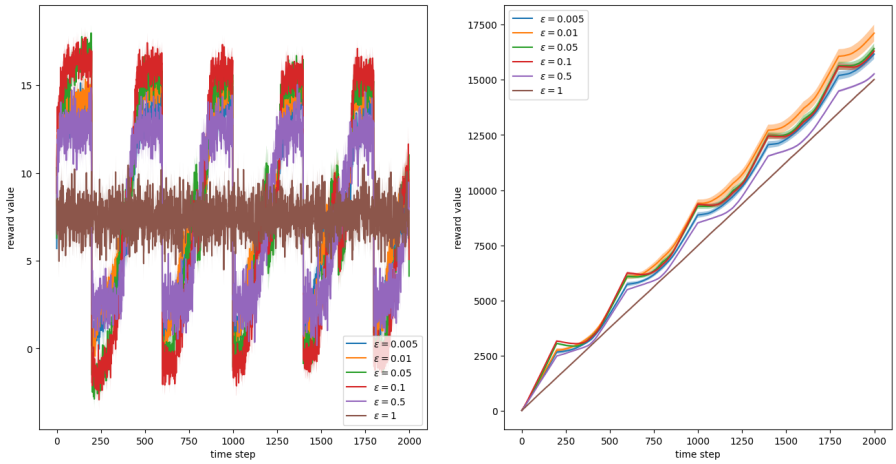
Initialize window size W and confidence parameter c
Initialize arrays $N_i(0) = 0$, $Q_i(0) = 0$ for all arms i
for each round $t = 1, 2, \ldots, T$ **do**
    for each arm i **do**
        Calculate $N_i(t)$ as the number of selections of arm i within the last W rounds
        Calculate $Q_i(t)$ as the average reward of arm i within the last W rounds
        if $N_i(t) > 0$ **then**
            Compute $UCB_i(t) = Q_i(t) + c\sqrt{\frac{2\log t}{N_i(t)}}$
        else
            Set $UCB_i(t)$ to a value that ensures arm i is selected
        end if
    end for
    Select arm $I_t = \arg\max_i UCB_i(t)$
    Observe reward r_t from arm I_t
    Update the reward history for arm I_t
end for

Algorithm 6 Bayes UCB Algorithm

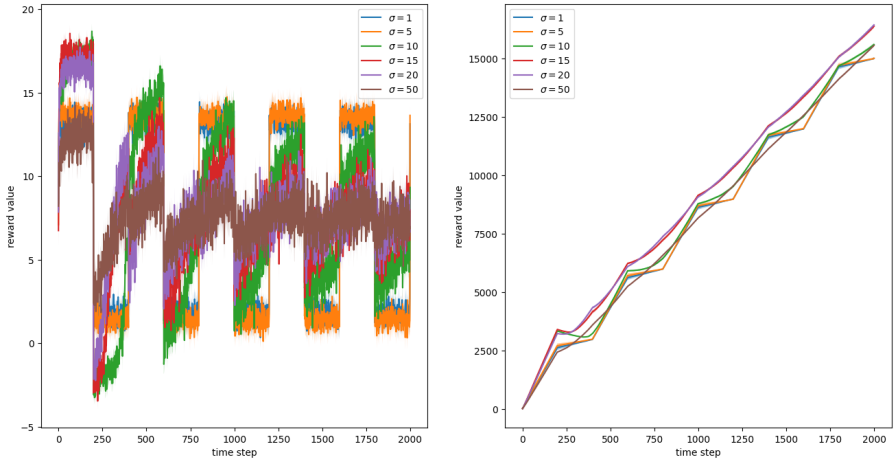
Initialize n (horizon), $\prod^0$ (initial prior on θ), c (parameters of the quantile)
for $t = 1, 2, \ldots, n$ **do**
    for each arm $j = 1, 2, \ldots, K$ **do**
        compute

$$q_j(t) = Q\left(1 - \frac{1}{t(\log n)^c}, \lambda_j^{t-1}\right)$$

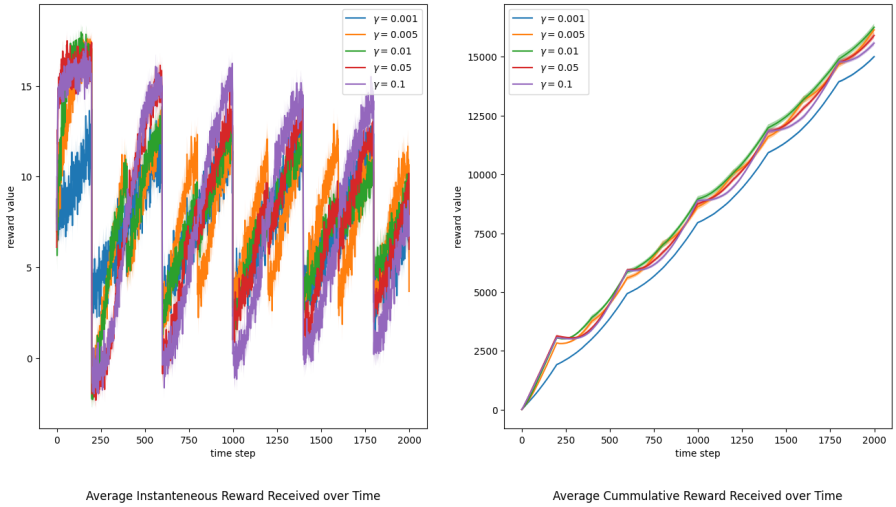
    end for
    draw arm $I_t = \arg\max_{j=1,\ldots,K} q_j(t)$
    get reward $X_t = Y_{I_t,t}$ and update $\prod^t$ according to (3)
end for



(a) ϵ-greedy. Optimal: $\epsilon = 0.01$.

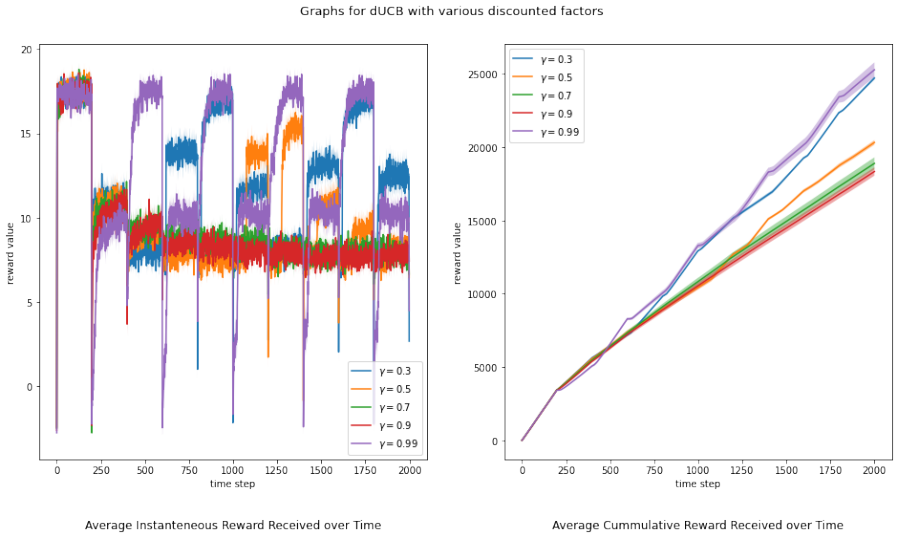


(b) Thompson Sampling. Optimal: $\sigma = 20$.

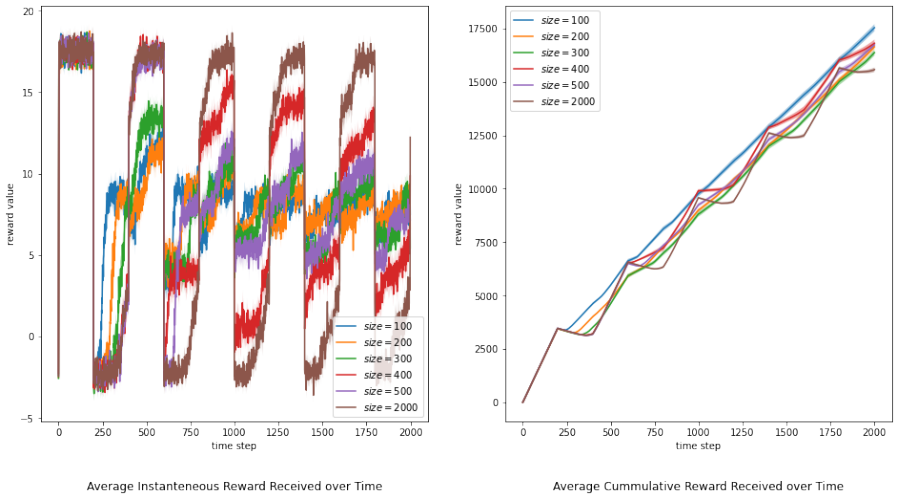


(c) EXP3. Optimal: $\gamma = 0.01$.

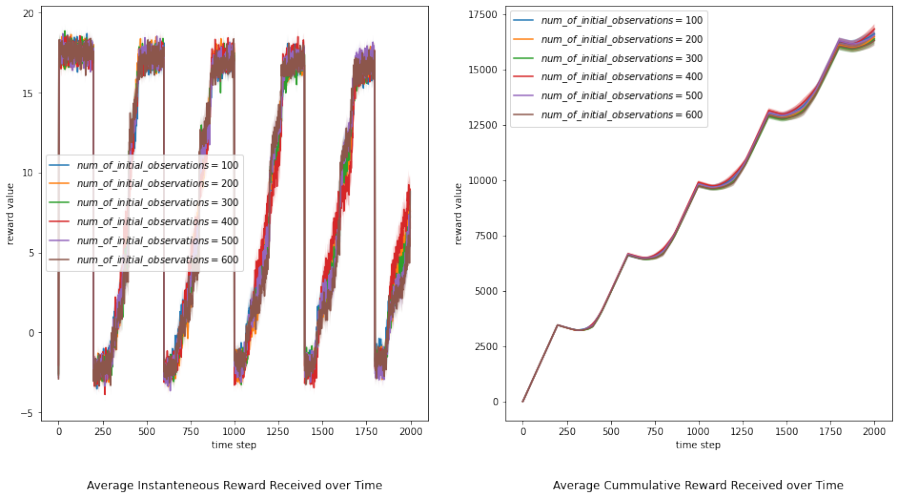
Figure 2: HM-HV-HF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.99$; c = 2

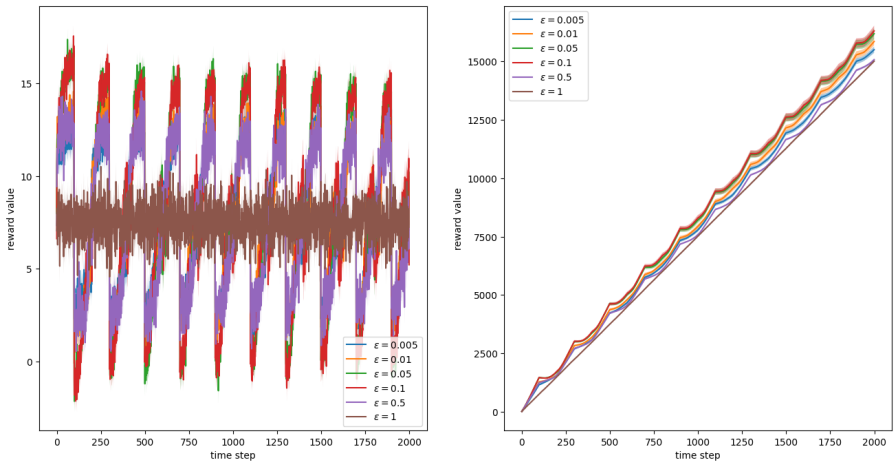


(b) swUCB. Optimal: window size = 100; c = 2

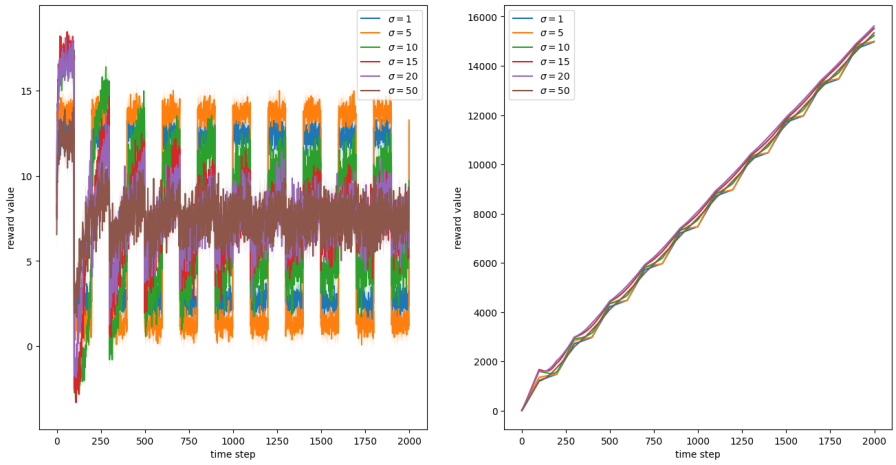


(c) bUCB. Optimal: $num\_initial\_obs$ = 400; c = 2

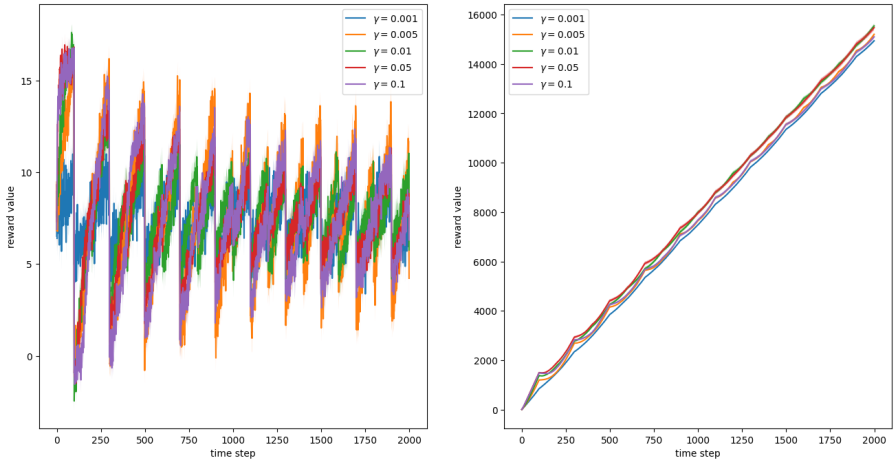
Figure 3: HM-HV-HF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.1$.

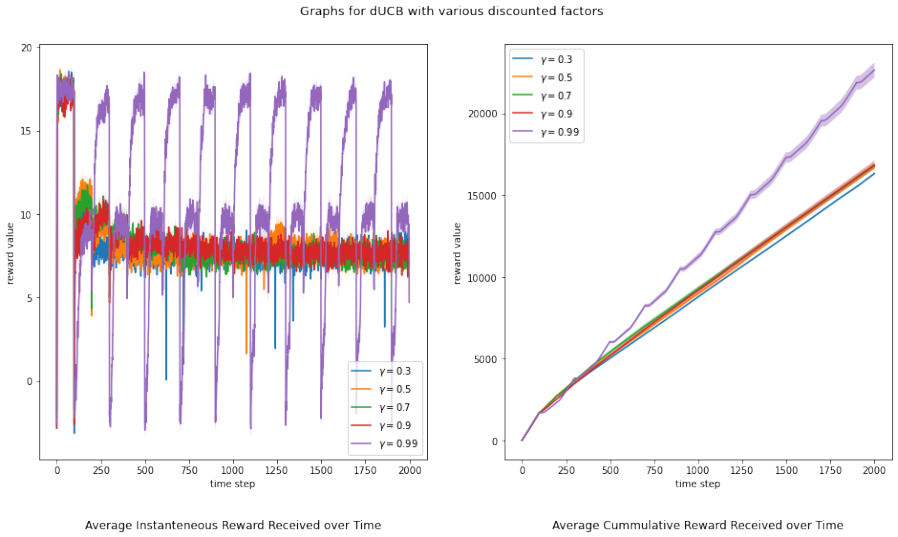


(b) Thompson Sampling. Optimal: $\sigma = 20$.

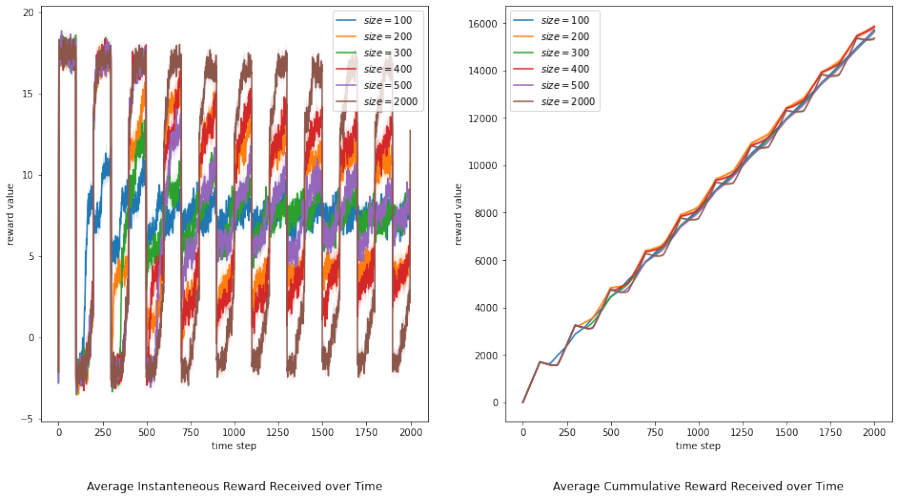


(c) EXP3. Optimal: $\gamma = 0.01$.

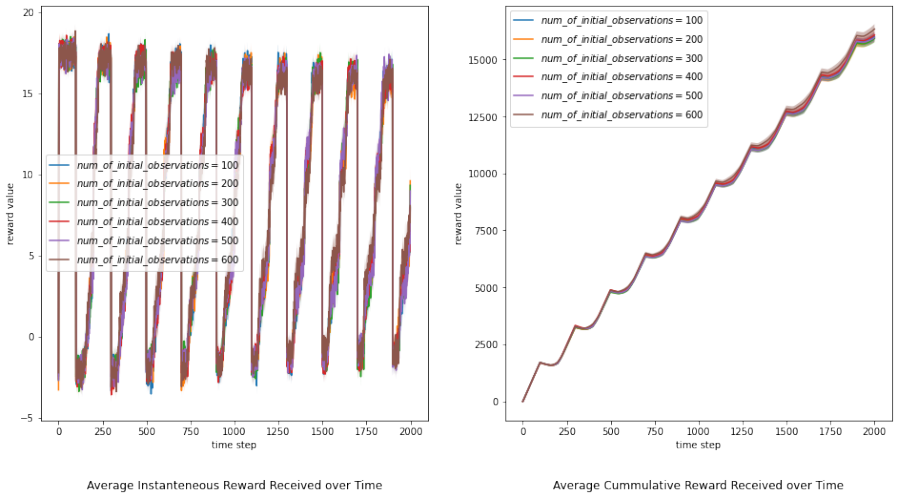
Figure 4: HM-HV-LF tunning plots (Part1)



(a) dUCB. Optimal: $\gamma = 0.99$; c = 2

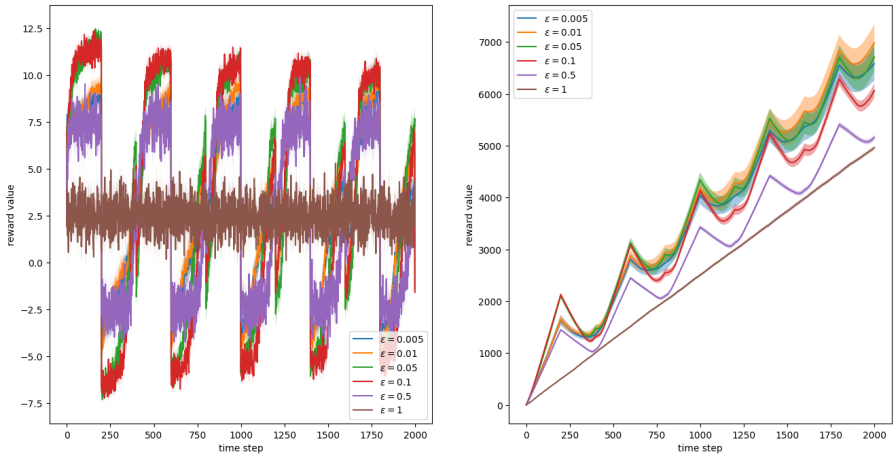


(b) swUCB. Optimal: window size = 400; c = 2

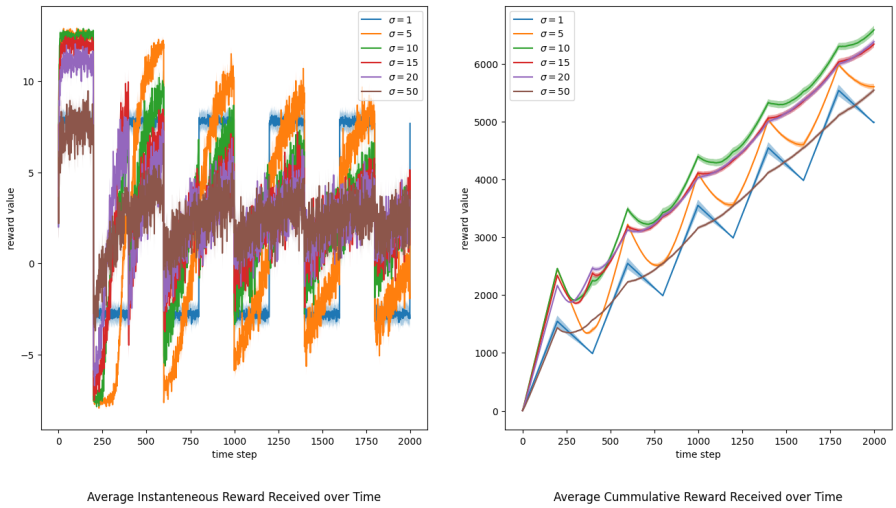


(c) bUCB. Optimal: $num\_initial\_obs$ = 600; c = 2

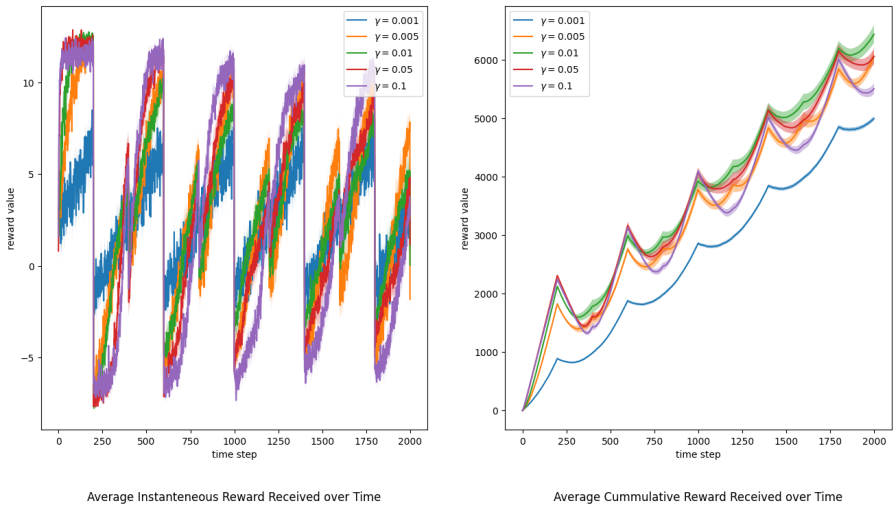
Figure 5: HM-HV-HF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.01$.

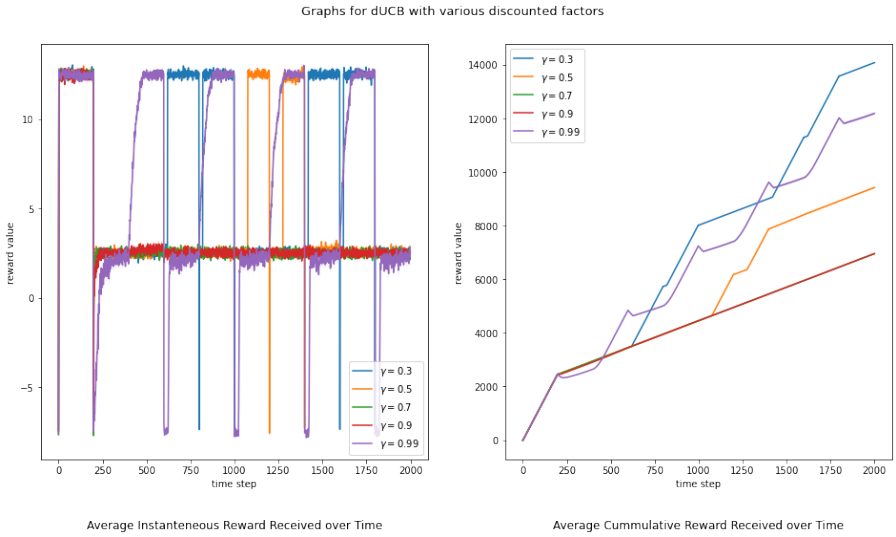


(b) Thompson Sampling. Optimal: $\sigma = 10$.

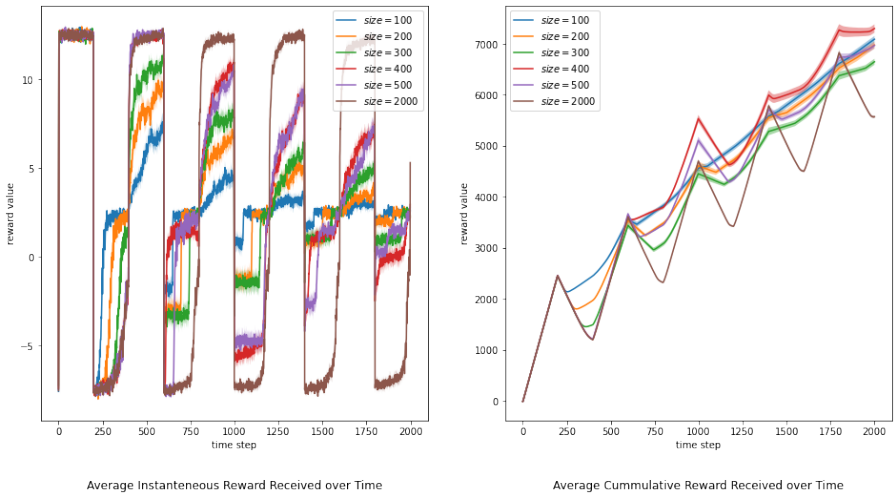


(c) EXP3. Optimal: $\gamma = 0.01$.

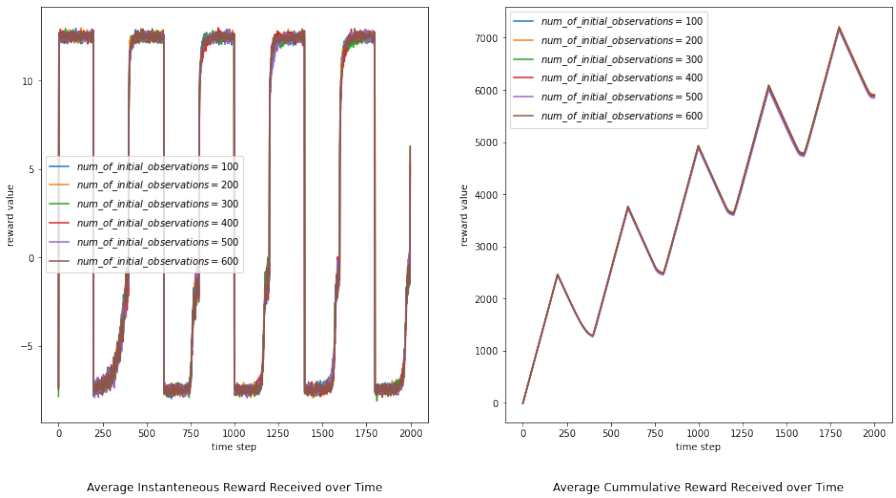
Figure 6: HM-LV-HF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.3$; c = 2

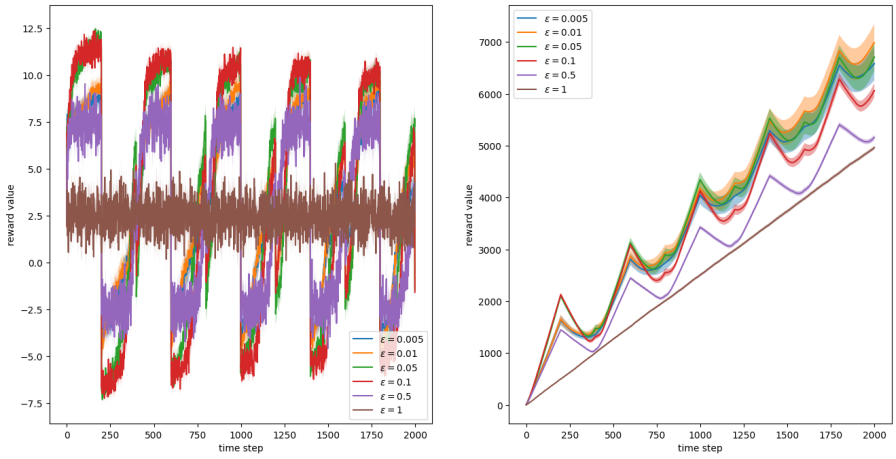


(b) swUCB. Optimal: window size = 400; c = 2

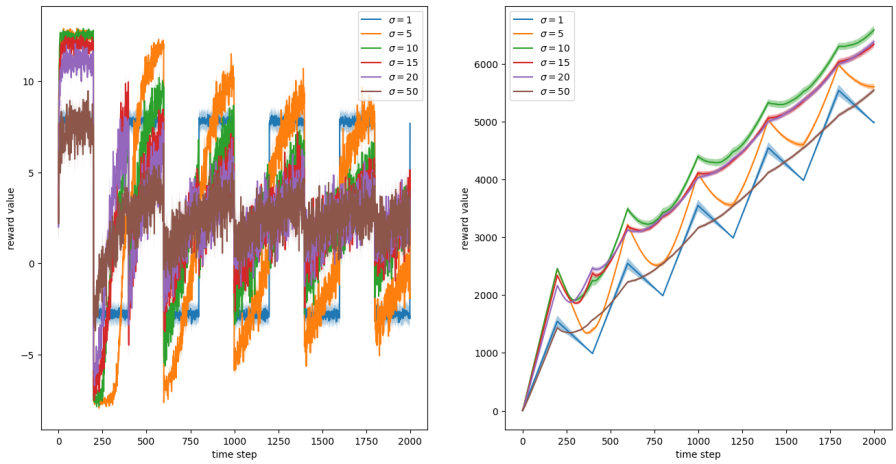


(c) bUCB. Optimal: $num\_initial\_obs$ = 400; c = 2

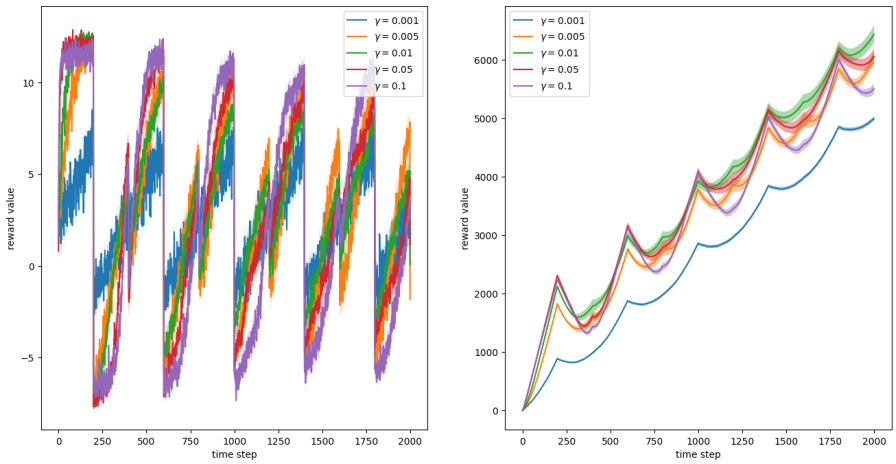
Figure 7: HM-LV-HF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.01$.

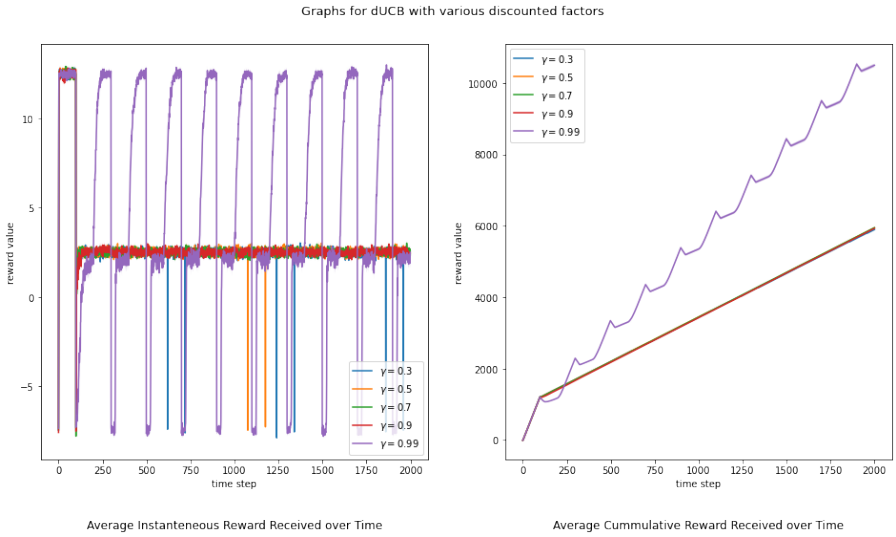


(b) Thompson Sampling. Optimal: $\sigma = 10$.

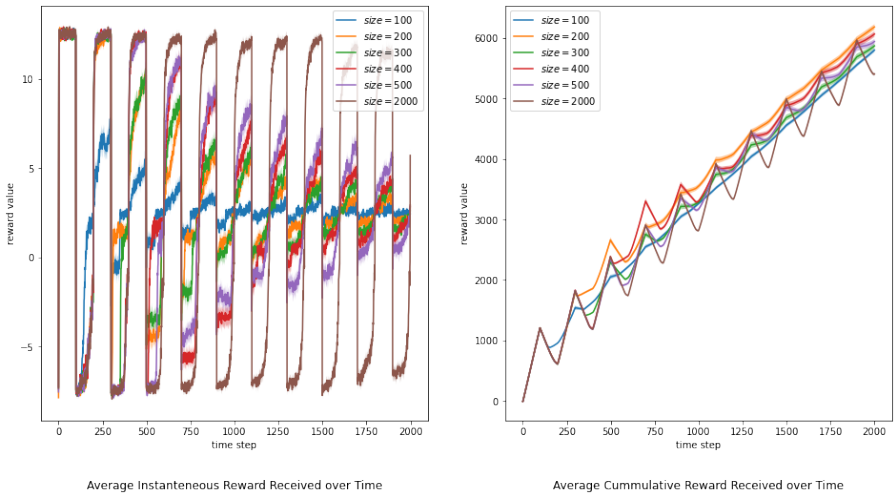


(c) EXP3. Optimal: $\gamma = 0.01$.

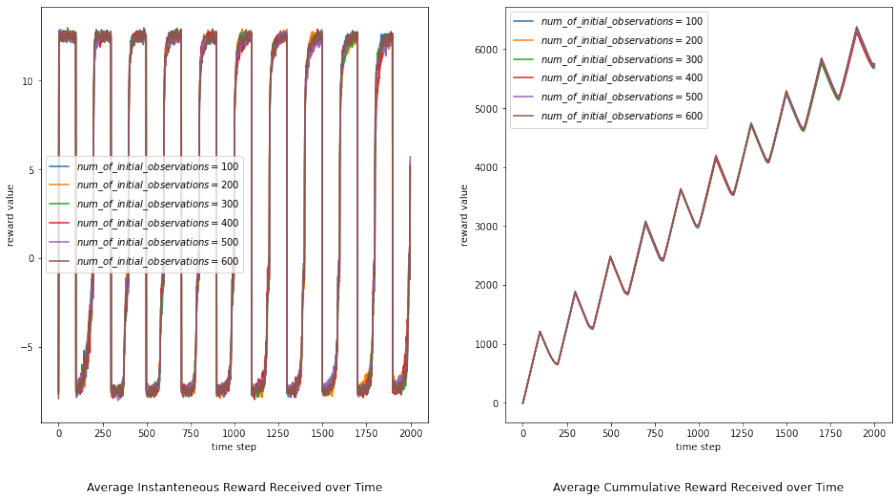
Figure 8: HM-LV-LF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.99$; c = 2

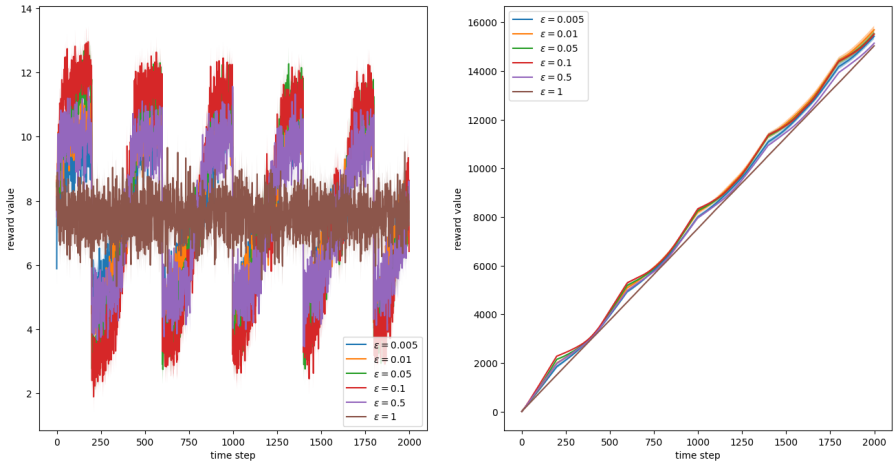


(b) swUCB. Optimal: window size = 200; c = 2

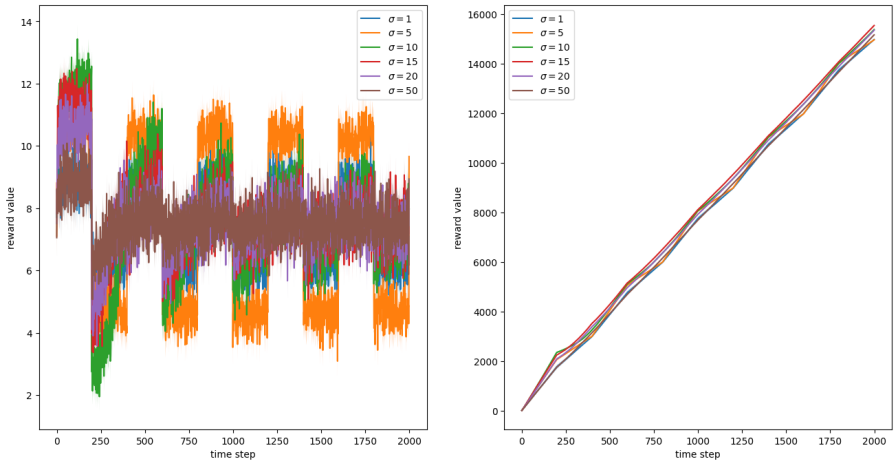


(c) bUCB. Optimal: $num\_initial\_obs$ = 400; c = 2

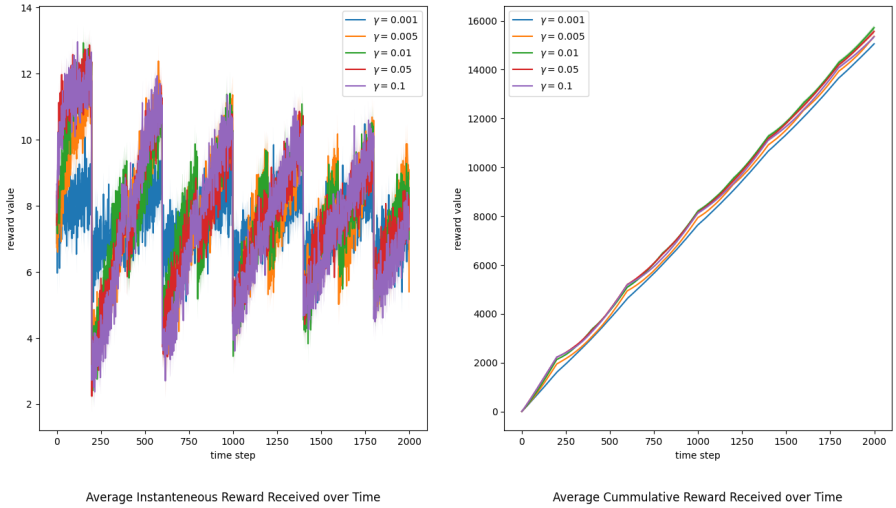
Figure 9: HM-LV-LF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.01$.

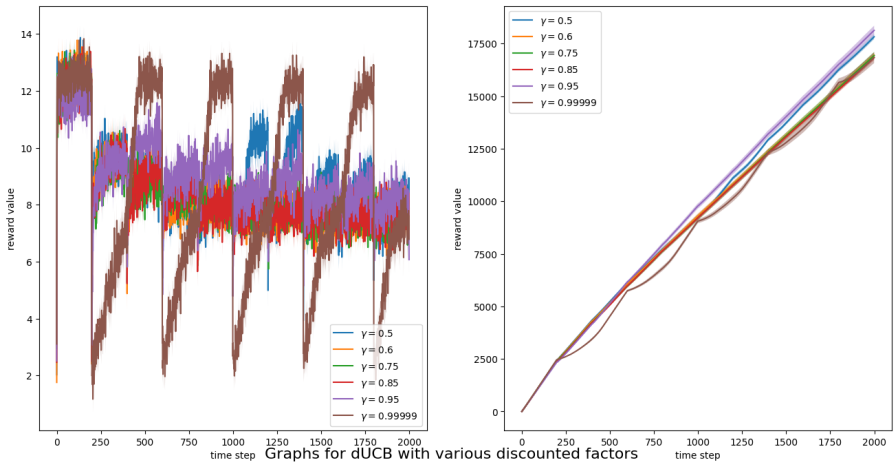


(b) Thompson Sampling. Optimal: $\sigma = 15$.

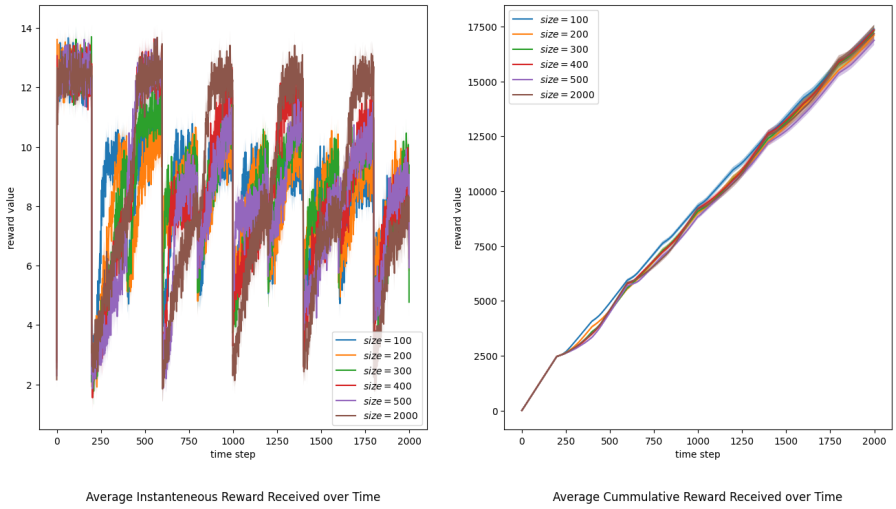


(c) EXP3. Optimal: $\gamma = 0.01$.

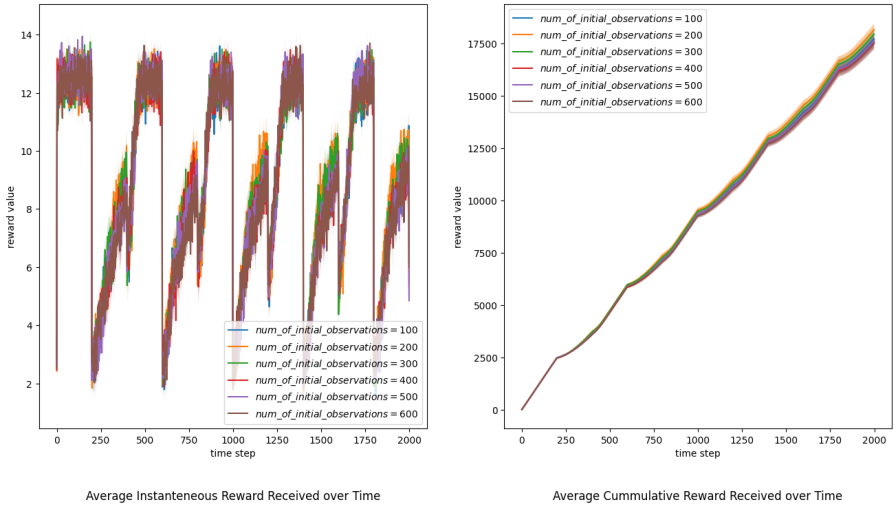
Figure 10: LM-HV-HF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.95$; c = 2

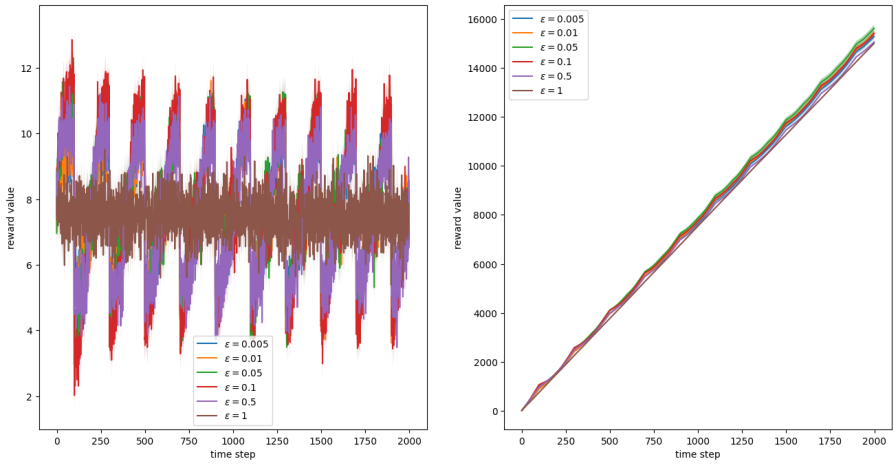


(b) swUCB. Optimal: window size = 2000; c = 2

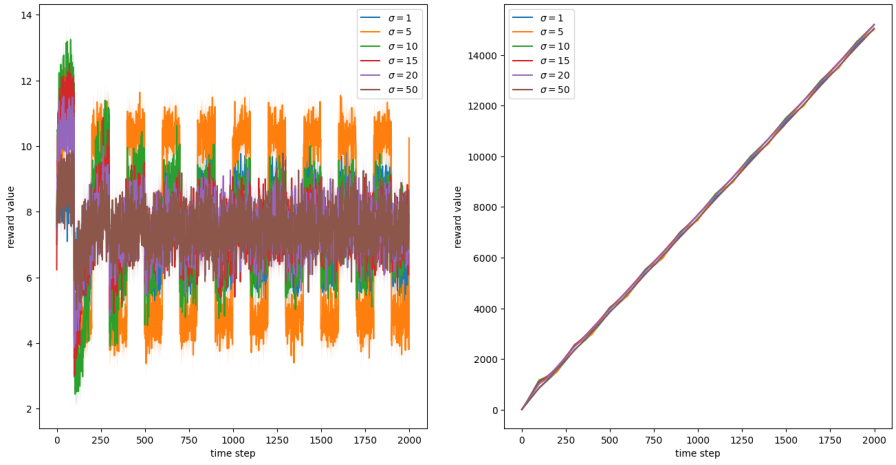


(c) bUCB. Optimal: $num\_initial\_obs$ = 200; c = 2

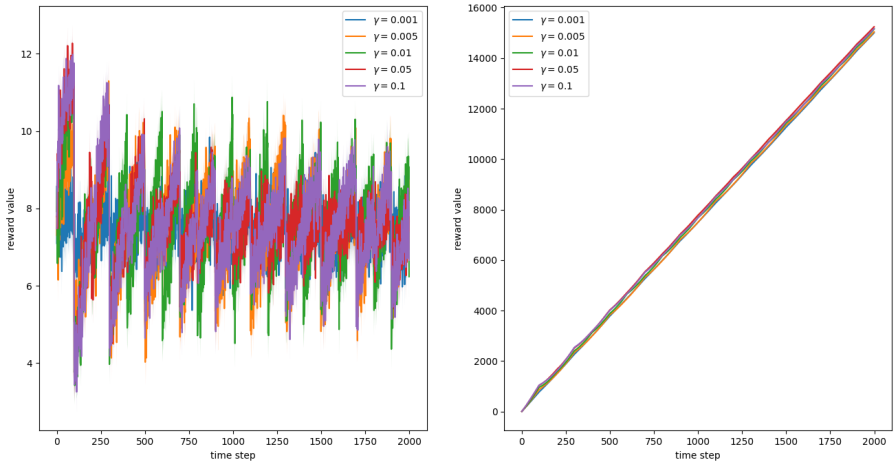
Figure 11: LM-HV-HF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.05$.

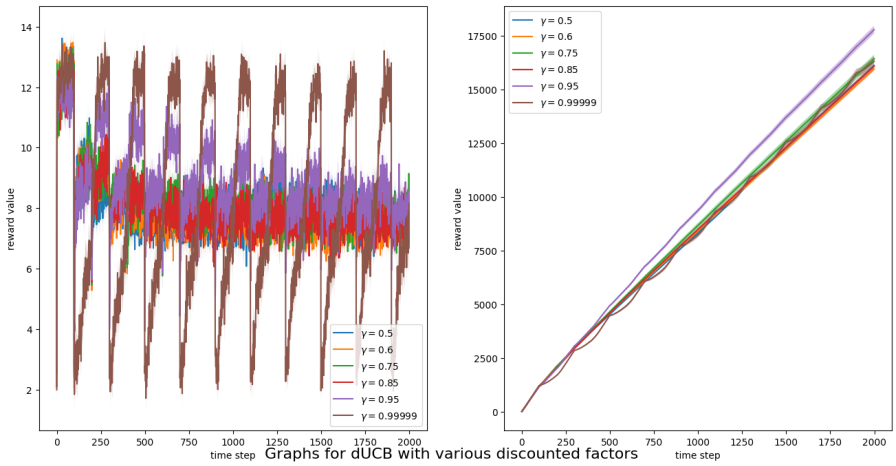


(b) Thompson Sampling. Optimal: $\sigma = 20$.

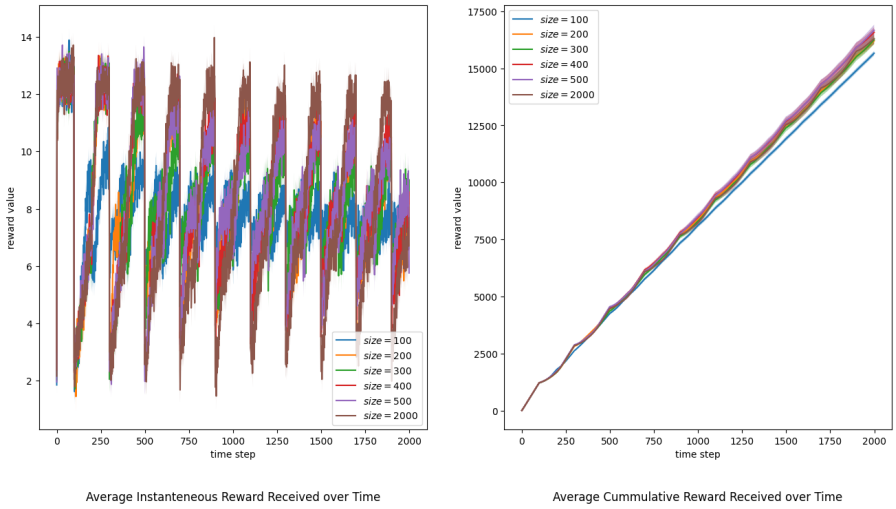


(c) EXP3. Optimal: $\gamma = 0.05$.

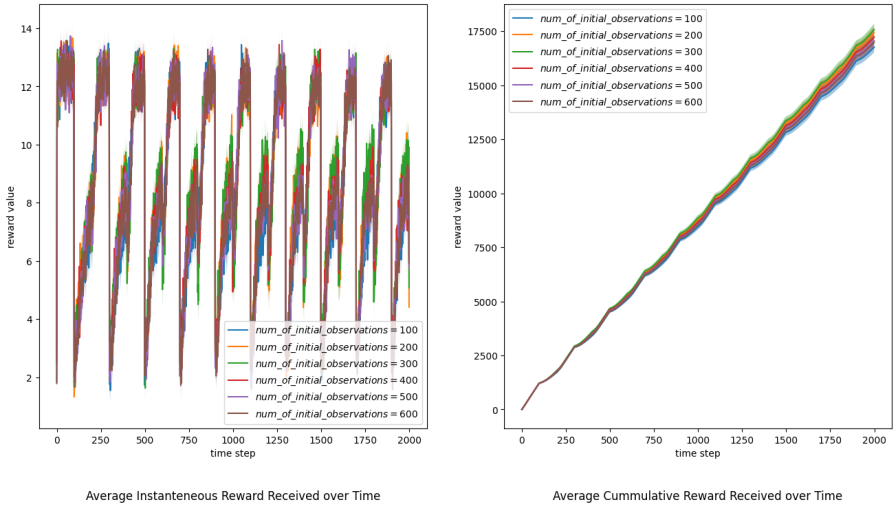
Figure 12: LM-HV-LF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.95$; c = 2

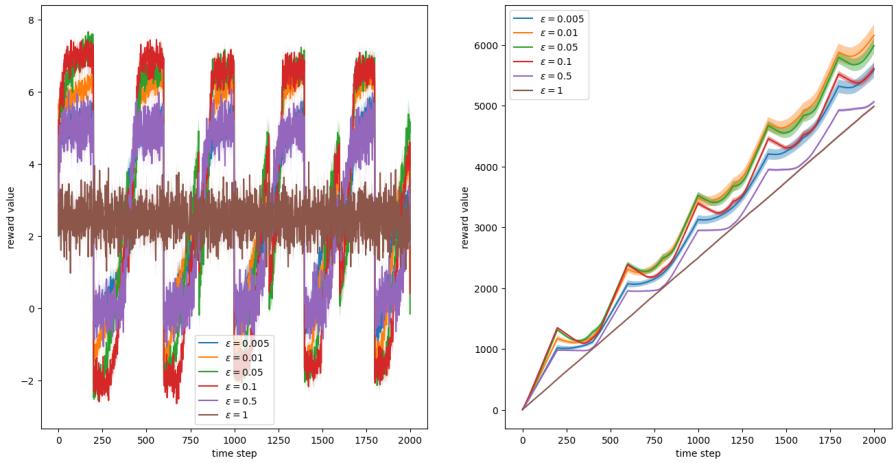


(b) swUCB. Optimal: window size = 500; c = 2

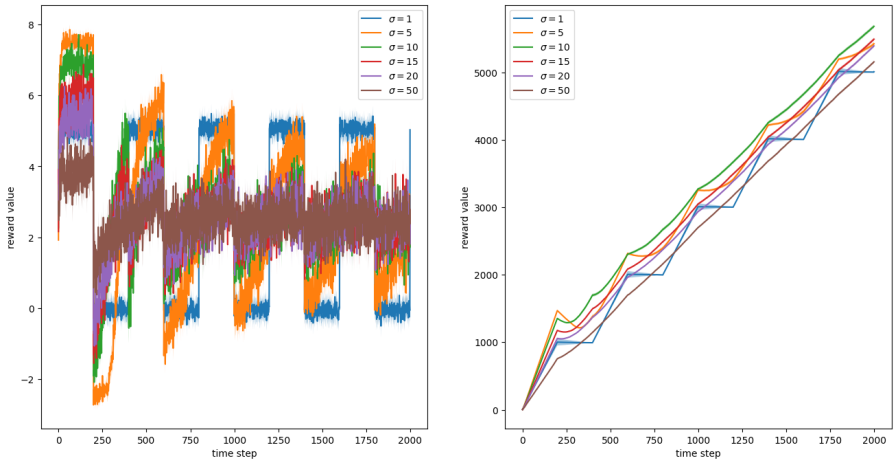


(c) bUCB. Optimal: $num\_initial\_obs$ = 300; c = 2

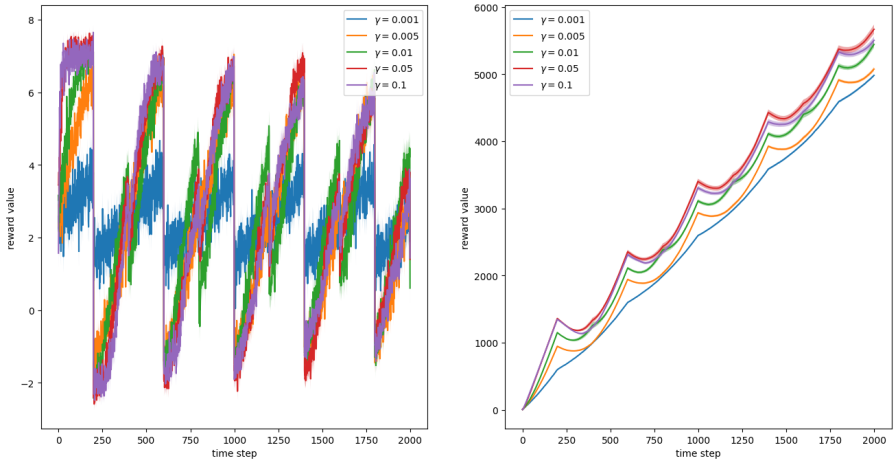
Figure 13: LM-HV-LF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.01$.

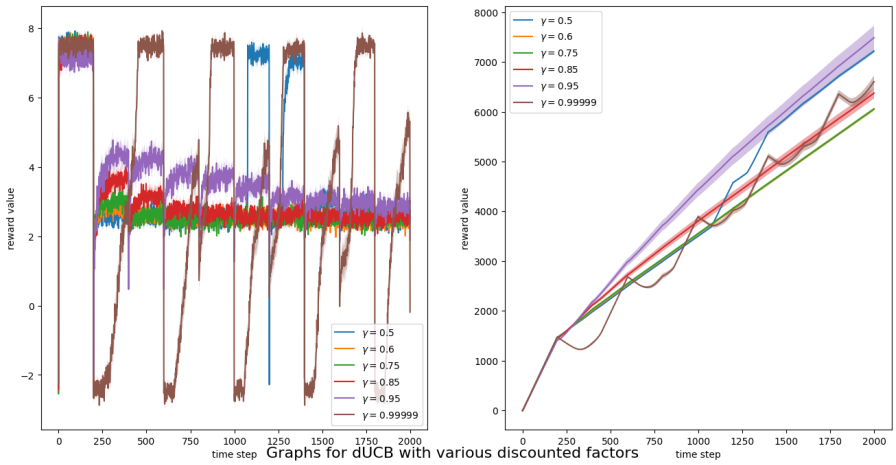


(b) Thompson Sampling. Optimal: $\sigma = 10$.

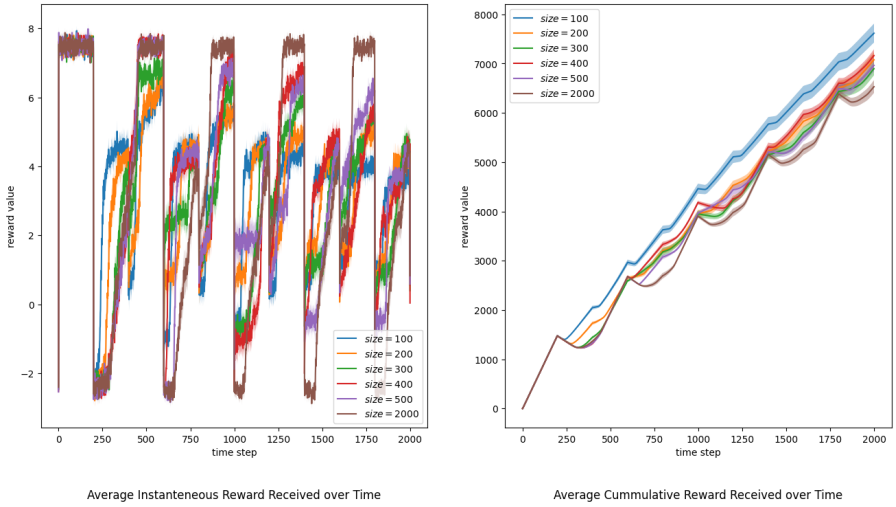


(c) EXP3. Optimal: $\gamma = 0.05$.

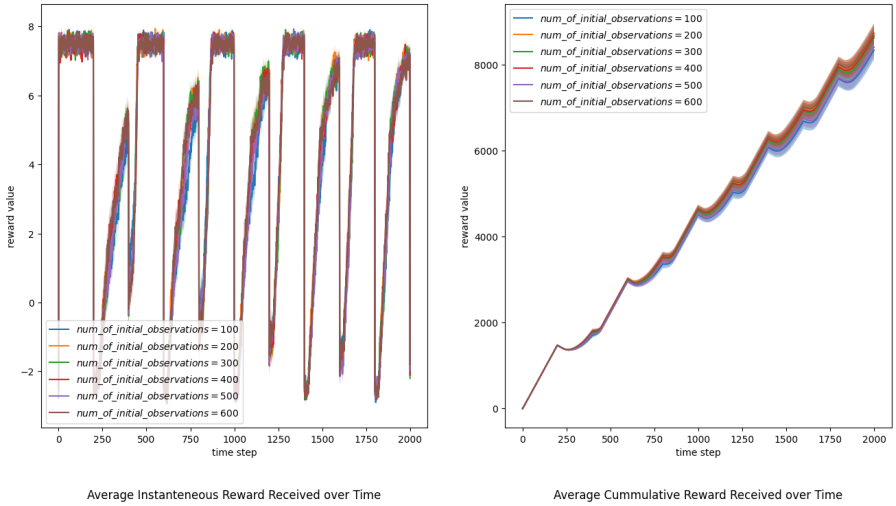
Figure 14: LM-LV-HF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.95$; c = 2

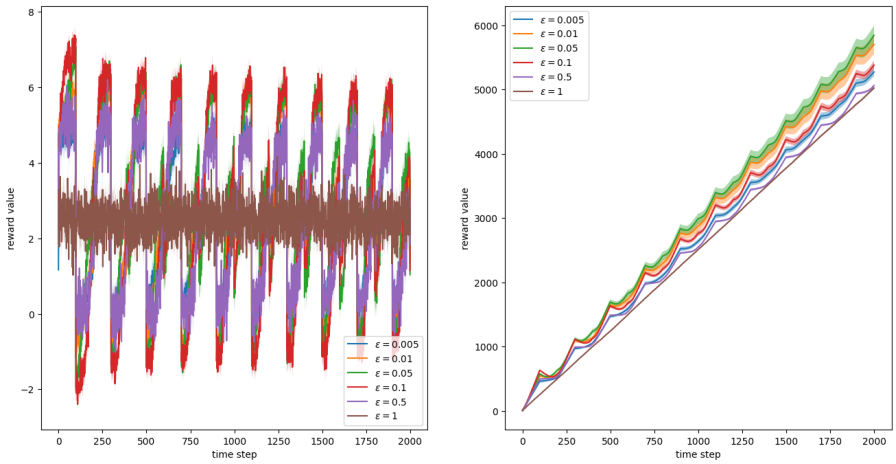


(b) swUCB. Optimal: window size = 100; c = 2

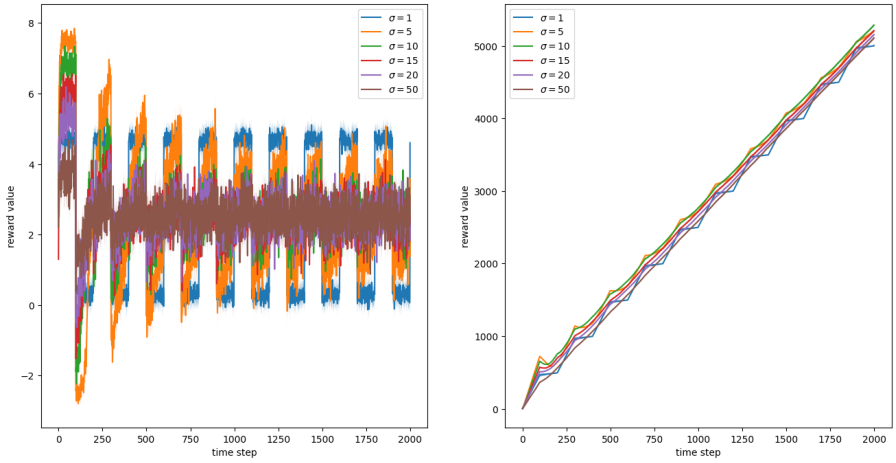


(c) bUCB. Optimal: $num\_initial\_obs$ = 600; c = 2

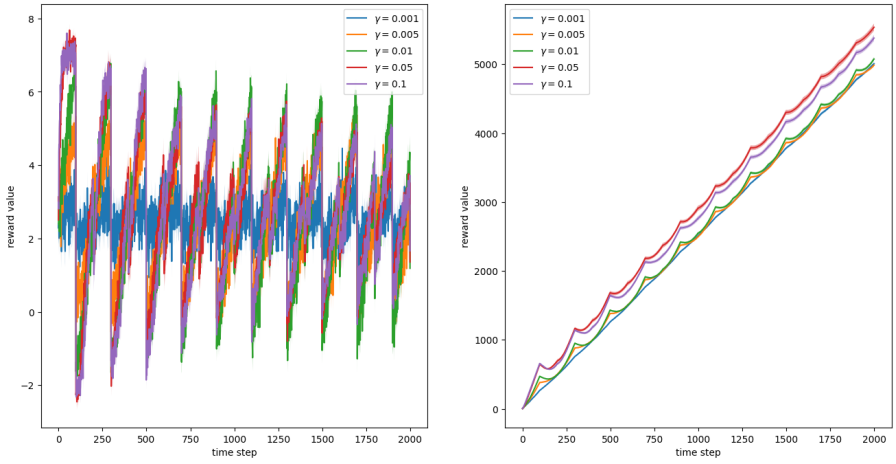
Figure 15: LM-LV-HF tunning plots (Part 2)



(a) ϵ-greedy. Optimal: $\epsilon = 0.05$.

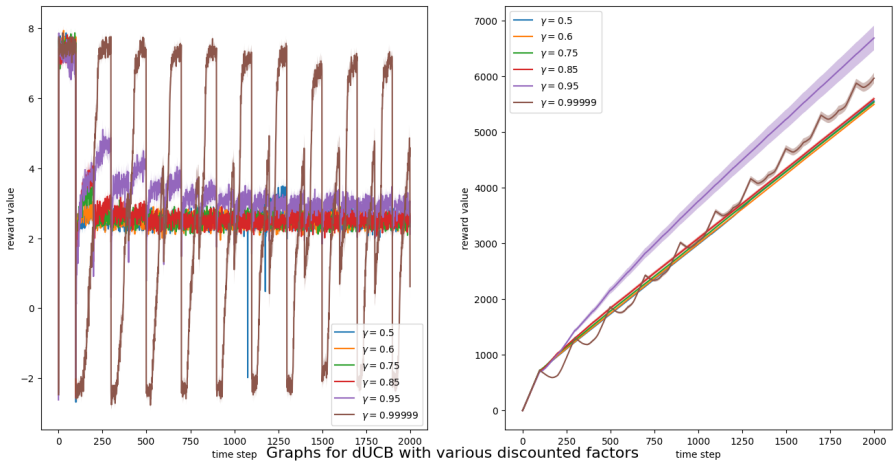


(b) Thompson Sampling. Optimal: $\sigma = 10$.

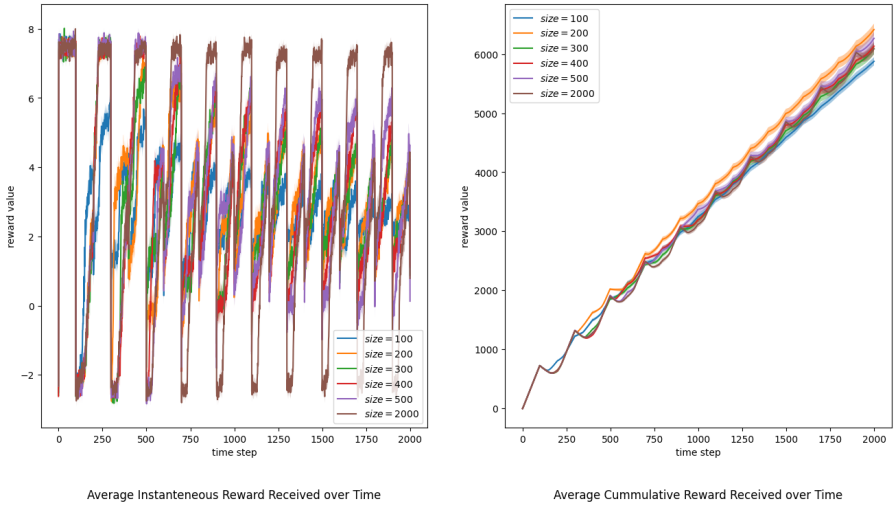


(c) EXP3. Optimal: $\gamma = 0.05$.

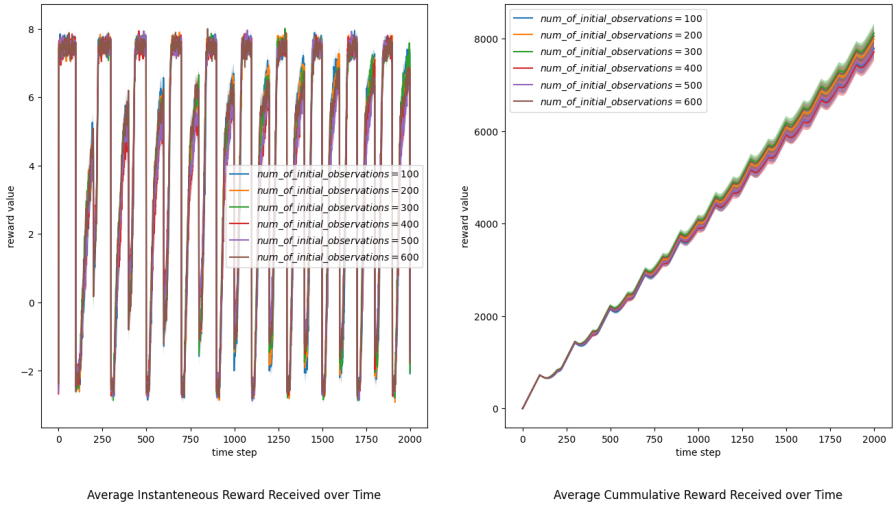
Figure 16: LM-LV-LF tunning plots (Part 1)



(a) dUCB. Optimal: $\gamma = 0.95$; c = 2



(b) swUCB. Optimal: window size = 200; c = 2



(c) bUCB. Optimal: $num\_initial\_obs$ = 300; c = 2

Figure 17: LM-LV-LF tunning plots (Part 2)

References

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer, *Finite-time analysis of the multi-armed bandit problem*, Machine Learning **47** (2002), no. 2-3, 235–256.

[ACBFS02] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire, *Non-stochastic bandit problems and exp3*, Proceedings of the thirteenth annual conference on Computational learning theory, Springer, 2002, pp. 23–37.

[AFM17] Robin Allesiardo, Raphaël Féraud, and Odalric-Ambrym Maillard, *The non-stationary stochastic multi-armed bandit problem*, International Journal of Data Science and Analytics **3** (2017), no. 4, 267–283.

[CSSZ21] E Cavenaghi, G Sottocornola, F Stella, and M Zanker, *Non stationary multi-armed bandit: Empirical evaluation of a new concept drift-aware algorithm*, Entropy (Basel) **23** (2021), no. 3, 380.

[KCG12] Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier, *On bayesian upper confidence bounds for bandit problems*, Artificial intelligence and statistics, PMLR, 2012, pp. 592–600.

[LPP10] Tyler Lu, Dávid Pál, and Martin Pál, *Contextual multi-armed bandits*, Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 485–492.

[RVRK18] Daniel J. Russo, Benjamin Van Roy, and Abbas Kazerouni, *A tutorial on thompson sampling*, Foundations and Trends® in Machine Learning **11** (2018), no. 1, 1–96.

[SB18] Richard S. Sutton and Andrew G. Barto, *Reinforcement learning: An introduction*, MIT Press, 2018.

[Tho33] William R Thompson, *On the likelihood that one unknown probability exceeds another in view of the evidence of two samples*, Biometrika **25** (1933), no. 3-4, 285–294.