
Is Prior-Free Black-Box Non-Stationary Reinforcement Learning Feasible?

Argyrios Gerogiannis
Grainger College of Engineering
ECE and CSL, UIUC
ag91@illinois.edu

Yu-Han Huang
Grainger College of Engineering
ECE and CSL, UIUC
yuhanhh2@illinois.edu

Venugopal V. Veeravalli
Grainger College of Engineering
ECE and CSL, UIUC
vvv@illinois.edu

Abstract

We study the problem of Non-Stationary Reinforcement Learning (NS-RL) without prior knowledge about the system’s non-stationarity. A state-of-the-art, black-box algorithm, known as MASTER, is considered, with a focus on identifying the conditions under which it can achieve its stated goals. Specifically, we prove that MASTER’s non-stationarity detection mechanism is not triggered for practical choices of horizon, leading to performance akin to a random restarting algorithm. Moreover, we show that the regret bound for MASTER, while being order optimal, stays above the worst-case linear regret until unreasonably large values of the horizon. To validate these observations, MASTER is tested for the special case of piecewise stationary multi-armed bandits, along with methods that employ random restarting, and others that use quickest change detection to restart. A simple, order optimal random restarting algorithm, that has prior knowledge of the non-stationarity is proposed as a baseline. The behavior of the MASTER algorithm is validated in simulations, and it is shown that methods employing quickest change detection are more robust and consistently outperform MASTER and other random restarting approaches.

1 INTRODUCTION

In the Reinforcement Learning (RL) problem an agent interacts with an environment with the goal of maxi-

mizing some notion of cumulative reward (Sutton and Barto, 2018). Standard RL algorithms assume that the environment is stationary in time. However, the stationarity assumption may not be accurate in many practical settings, and therefore the non-stationary version of the RL problem has received some attention in recent years.

The problem of Non-Stationary RL (NS-RL) has been studied across various settings, and different solutions have been proposed (Auer et al., 2019; Chen et al., 2019; Abbasi-Yadkori et al., 2023; Cheung et al., 2019; Russac et al., 2020; Faury et al., 2021; Besson et al., 2022; Mao et al., 2021; Chen and Luo, 2022; Touati and Vincent, 2020; Cheung et al., 2020a; Cheng et al., 2023; Wei and Luo, 2021). A key factor distinguishing solution methods is their approach to mitigating the underlying non-stationarity. Based on this characteristic, methods can be broadly classified into *adaptive* approaches and *restarting* approaches. The former involve designing algorithms that are able to run continuously without restarting, while adjusting their performance, whereas the latter employ a mechanism according to which the algorithm restarts the learning process at certain points in time.

In recent work, Peng and Papadimitriou (2024) have shown that the complexity of continuously adapting the learning process can be computationally prohibitive, suggesting that restarting the learning process in response to significant changes in the environment is preferable. Due to this observation, the focus of our work is on restarting based approaches. Under this paradigm, a critical task of the NS-RL problem is the development of an appropriate restarting mechanism for a given setting. The design of the decision rule that leads to the restart is intricate as both high accuracy and low computational complexity are essential. A well-designed restarting algorithm should reset after the non-stationarity has affected the environment significantly, while incurring as small a delay as possible. To decide when to restart, the algorithm can either

have prior information about the non-stationarity or be *prior-free*, i.e., it operates without any prior knowledge of the non-stationarity. The prior-free approach is more applicable in practice, and there have been a number of recent works on providing order optimal, prior-free solutions in different NS-RL settings (Wei and Luo, 2021; Chen et al., 2019; Auer et al., 2019; Abbasi-Yadkori et al., 2023; Besson et al., 2022).

In these works on prior-free solutions, various non-stationarity detection tests have been proposed (Auer et al., 2019; Abbasi-Yadkori et al., 2023; Wei and Luo, 2021). Works in Non-Stationary Multi-Armed Bandits (NS-MABs) have employed Quickest Change Detection (QCD) (Poor and Hadjiliadis, 2008; Veeravalli and Banerjee, 2014) techniques to restart in piecewise stationary settings. In fact, the QCD-based approach of Besson et al. (2022) in NS-MABs under piecewise stationarity, is the state-of-the-art method, which not only significantly outperforms the influential prior-free approach proposed in Auer et al. (2019) but also demonstrates better detection capabilities.

In the general NS-RL literature, the method introduced by Wei and Luo (2021) stands out. This prior-free method, termed as MASTER, is applicable across various NS-RL settings and achieves order optimal performance. This is the only known method with such generic black-box capabilities: as long as the input algorithm satisfies MASTER’s assumptions, it can be employed in a seamless manner. MASTER was applied to various RL settings¹ including: Multi-armed Bandits (Auer et al., 2002), Linear Bandits (Abbasi-Yadkori et al., 2011), Generalized Linear Bandits (GLBs) (Filippi et al., 2010), Finite-horizon Tabular Markov Decision Processes (Jin et al., 2018), Finite-horizon Linear Markov Decision Processes (FLMDPs) (Jin et al., 2020) and Contextual Bandits (Agarwal et al., 2014; Simchi-Levi and Xu, 2022). The paper by Wei and Luo (2021) lacks experimental validation. A few works have tested MASTER’s practical effectiveness, under the settings of GLBs (Wang et al., 2023) and FLMDPs (Zhou et al., 2022). The absence of empirical validation in other settings motivates the need to assess MASTER’s practical feasibility.

Our goal in this paper is to evaluate the performance bounds for MASTER given in Wei and Luo (2021), and assess its practical effectiveness through experimental validation. In summary, the primary contributions of this work are:

- We prove that MASTER’s non-stationarity de-

¹MASTER was also applied to infinite-horizon Markov Decision Processes, but it is not order optimal under prior-free assumptions. Thus, its analysis under this setting will not be studied in this work.

tection mechanism is not triggered for practical choices of horizon, leading to performance akin to a random restarting algorithm.

- We show that the regret bound for MASTER given in Wei and Luo (2021), while being order optimal, stays above the worst-case linear regret until unreasonably large values of the horizon.
- We run experiments for the special NS-RL problem of Piecewise Stationary MABs (PS-MABs), and validate the performance issues of MASTER.
- We propose an order optimal, non-prior-free, random restarting algorithm for PS-MABs, that can be used as a baseline for the problem.
- We show in simulations for PS-MABs that methods which employ QCD techniques are more robust and superior in performance compared to MASTER and the order optimal, random restarting approach.

2 PRELIMINARIES

Notation: For any non-negative integers $n, e, [n]$ denotes the set $\{1, 2, \dots, n\}$ and $[n, e]$ corresponds to $\{n, n+1, \dots, e\}$. The indicator function is denoted by $\mathbf{1}[\cdot]$. We write $h_1(x) = \tilde{\mathcal{O}}(h_2(x))$ with high probability, if $h_1(x) = \mathcal{O}(\text{poly}(\log(T/\delta))h_2(x))$ with probability $1 - \delta$ for some $\delta \in (0, 1)$. The base of log is the natural base, unless explicitly differentiated.

2.1 Problem Formulation And Definitions

We formally define the NS-RL problem as follows.

Definition 1 Assume an agent that interacts with an environment through a policy set Π over a horizon T , that is, a fixed number of time steps. The environment has a collection of T reward functions $f_1, \dots, f_T : \Pi \rightarrow [0, 1]$, unknown to the agent, that model the environment’s feedback. In each time step $t = 1, \dots, T$, the agent selects a policy $\pi_t \in \Pi$ and receives the feedback in the form of a random reward $R_t \in [0, 1]$ with mean $f_t(\pi_t) = \mathbb{E}[R_t | \pi_t]$. The goal is to minimize the dynamic regret, $\text{Reg}_D(T) = \mathbb{E}\left[\sum_{t=1}^T (f_t^* - f_t(\pi_t))\right]$, where $f_t^* = \max_{\pi \in \Pi} f_t(\pi)$ is the best expected reward at time t .

To measure the underlying non-stationarity, a general measure is provided as follows.

Definition 2 (Following Wei and Luo (2021).) $\Delta : [T] \rightarrow \mathbb{R}$ is a non-stationarity measure, if it satisfies $\Delta(t) \geq \max_{\pi \in \Pi} |f_t(\pi) - f_{t+1}(\pi)|$, for all t . For interval $[s, e]$, the cumulative non-stationarity is given as

$$\Delta_{[s,e]} = \sum_{\tau=s}^{e-1} \Delta(\tau), \text{ with } \Delta_{[s,s]} = 0, \text{ and the number of changes (+1) as } L_{[s,e]} = 1 + \sum_{\tau=s}^{e-1} \mathbf{1}[\Delta(\tau) \neq 0].$$

For a given horizon T , the total cumulative non-stationarity and the number of changes are denoted as $\Delta = \Delta_{[1,T]}$ and $L = L_{[1,T]}$, respectively. In every setting except infinite-horizon MDPs, the dynamic regret minimax bound in $\tilde{\mathcal{O}}(\cdot)$ is given by $\min\{\text{Reg}_L^*, \text{Reg}_\Delta^*\}$, where $\text{Reg}_L^* = \sqrt{LT}$ and $\text{Reg}_\Delta^* = \Delta^{1/3}T^{2/3} + \sqrt{T}$. These bounds are known to be optimal even when L and Δ are known (Wei and Luo, 2021). Thus, the goal in prior-free NS-RL is to achieve the optimal bound when L and Δ are unknown, i.e., an algorithm that is both prior-free and order optimal.

2.2 The MASTER Algorithm

MASTER is a black-box reduction that schedules and runs multiple instances of an input algorithm, denoted as ALG, and uses two tests for non-stationarity detection. More specifically, MASTER is scheduled for blocks of length 2^n in the horizon, where $n \in \mathbb{N}$. The scheduling of multiple instances is done by first partitioning the interval of length 2^n equally into 2^{n-m} sub-intervals for every $m = 0, \dots, n$. Then, for each sub-interval of length 2^m , an instance of ALG with length 2^m is scheduled with probability $\frac{\rho(2^n)}{\rho(2^m)} \leq 1$ for that specific sub-interval. Since $\frac{\rho(2^n)}{\rho(2^m)} = 1$, there always exists a length- 2^n instance, and, depending on the random scheduling, possibly multiple length- 2^m instances.

After the scheduling and the initialization, the instance with the shortest length that covers the current time step is made active, while all the other instances remain inactive. MASTER follows the policy of the active instance and updates it accordingly, whereas all inactive instances are paused. Finally, MASTER performs the two non-stationarity tests, termed as Test 1 and Test 2, at the end of each time step and if either of their conditions is satisfied, MASTER restarts the described procedure with $n = n + 1$. We provide the analytic algorithm of MASTER in Appendix A.1.

MASTER, as a black-box method, has specific requirements for ALG, presented below.

Assumption 1 (Following Wei and Luo (2021).) *ALG outputs $\tilde{f}_t \in [0, 1]$ at the beginning of each time step t . Suppose there exists a non-increasing function $\rho : [T] \rightarrow \mathbb{R}$ such that $\rho(t) \geq 1/\sqrt{t}$ and $C(t) = t\rho(t)$ is non-decreasing. Additionally, assume $\Delta(t)$ to be a non-stationary measure. Then, for any $t \in [T]$ such that $\Delta_{[1,t]} \leq \rho(t)$, the following holds with probability at least $1 - \delta/T$:*

1. $\tilde{f}_t \geq \min_{\tau \in [1,t]} f_\tau^* - \Delta_{[1,t]}, \text{ and}$

2. $\frac{1}{t} \sum_{\tau=1}^t (\tilde{f}_\tau - R_\tau) \leq \rho(t) + \Delta_{[1,t]}.$

This assumption implies that if there exists an interval where the non-stationarity is weak, ALG should be able to achieve an sub-linear regret bound and provide an optimistic estimate of the optimal reward. Since the active instance of ALG at each time step may be different, the notation \tilde{g}_t is used to denote the scalar output \tilde{f}_t of the current active instance of MASTER. MASTER's approach is able to ensure an equivalence of the guarantees of Assumption 1 through a procedure termed as MALG. MALG is essentially the random scheduling scheme, i.e., MASTER without the non-stationarity detection tests. MALG is designed in such a way that its performance is equivalent to a multi-scale version of the Assumption 1, as shown in Lemma 3 from Wei and Luo (2021) which we present below.

Lemma 1 (Adapted from Wei and Luo (2021).) *Let $\hat{n} = \log_2 T + 1$, $\hat{\rho}(t) = 6\hat{n}\log(T/\delta)\rho(t)$ and $t' = t - \text{alg}.s + 1$. Suppose Assumption 1 holds for ALG, and that $n \leq \log_2 T$. Then, for any instance alg, with start at $\text{alg}.s$ and finish at $\text{alg}.e$, that is maintained by MALG and any $t \in [\text{alg}.s, \text{alg}.e]$ that satisfies $\Delta_{[\text{alg}.s,t]} \leq \rho(t')$, MALG satisfies with probability at least $1 - \delta/T$:*

1. $\tilde{g}_t \geq \min_{\tau \in [\text{alg}.s,t]} f_\tau^* - \Delta_{[\text{alg}.s,t]}, \text{ and}$
2. $\frac{1}{t'} \sum_{\tau=\text{alg}.s}^t (\tilde{g}_\tau - R_\tau) \leq \hat{\rho}(t') + \hat{n}\Delta_{[\text{alg}.s,t]}.$

Thus, it is evident that for an instance alg of MASTER, if there exists $t \in [\text{alg}.s, \text{alg}.e]$ such that $\Delta_{[\text{alg}.s,t]} \leq \rho(t')$, then that instance can ensure the sub-linear regret for the interval $[\text{alg}.s, t]$. However, due to the existence of non-stationarity, this condition may be violated, and therefore MASTER needs to restart. In a prior-free setting, the exact identification of this condition is not possible, and thus MASTER's detection scheme uses the consequences of Lemma 1 to identify the violation of the condition.

Also, there is an assumption about the parameter δ .

Assumption 2 (Following Wei and Luo (2021).) *The probability parameter $\delta \in (0, 1)$ is assumed to have an order of $1/\text{poly}(T)$.*

3 THEORETICAL ANALYSIS

3.1 MASTER's Non-Stationarity Detection

Regarding MASTER's analysis, it is necessary to ensure its non-stationarity detection scheme works effectively. MASTER employs two detection tests. The

first, Test 1, leads to a restart when for any 2^m -length instance alg , both $t = alg.e$ and the following hold:

$$\frac{1}{2^m} \sum_{\tau=alg.s}^{alg.e} R_\tau - \min_{\tau \in [t_n, t]} \tilde{g}_\tau \geq 9\hat{\rho}(2^m).$$

The second test, Test 2, is applied at each time step t of block 2^n , with the following restarting condition:

$$\frac{1}{t - t_n + 1} \sum_{\tau=t_n}^t (\tilde{g}_\tau - R_\tau) \geq 3\hat{\rho}(t - t_n + 1)$$

where t_n is the start of the current 2^n block and $t \in [t_n, t_n + 2^n - 1]$.

To detect significant non-stationarity using these tests, the statistics on the LHS need to surpass the threshold on the RHS. However, due to MASTER's assumptions, the LHS has an upper bound. Since $\rho(t)$ is non-increasing and $\rho(t) \geq 1/\sqrt{t}$, this enforces a constraint on δ , irrespective of the form of $\rho(t)$. On the other hand, fixing a specific order dependence for δ sets a bound on T for which the tests can be triggered.

Theorem 1 *If Assumption 1 is satisfied, the thresholds of Test 1 and Test 2 can be crossed only if $\delta \geq T \exp\left(-\frac{\sqrt{T}}{54(\log_2 T+1)}\right)$, and since $\delta < 1$, T must be at least 1.24 billion.*

The proof of this theorem is given in Appendix B.3. To prove this theorem, we also present the following auxiliary lemma, which provides additional insights into the non-stationarity detection tests.

Lemma 2 *Test 1 will not be triggered for any value of the horizon that satisfies $\rho(T) > \frac{1}{54(\log_2 T+1)\log(T/\delta)}$ and Test 2 will also not be triggered if $\rho(T) > \frac{1}{18(\log_2 T+1)\log(T/\delta)}$.*

Theorem 1 sets additional dependency for δ with T on top of Assumption 2, and it shows that the non-stationarity detection scheme cannot work for a significant range of possible horizons, highlighting possible practical issues. These results are derived assuming that the statistics of the tests on the LHS of the test inequalities are maximized and the threshold on the RHS is minimized, to identify the best conditions under which the thresholds are crossed. Fixing a dependency of $1/\text{poly}(T)$ for δ , can lead to a higher lower bound for the horizon, as can be seen in the inequality of Theorem 1. In addition, since the range of the statistics used for detection may be lower than when the LHS of the test inequalities is maximized, the horizon can be much more constrained. For example, if a case where the statistics of the tests range in $[0, 0.1]$ the minimum horizon value to detect the non-stationarity would be around at least 313 billion.

Remark 1 *The lower bound on T from Theorem 1 may be even larger as it depends on the form of the non-increasing function $\rho(t)$ and the dependence of δ with T . For instance, using the proposed functions in Wei and Luo (2021) for MABs, in the simplest case of a two-armed bandit, T needs to be at least 143 billion.*

Regardless of horizon length, the two non-stationarity detection tests also impose two different conditions on the size of the main 2^n -length instance as well as the length of the 2^m -length instances. Thus, we have the following corollary.

Corollary 1 *Any 2^m -length instance, such that $\rho(2^m) > \frac{1}{54(\log_2 T+1)\log(T/\delta)}$, will not trigger Test 1, and any 2^n -block such that $\rho(2^n) > \frac{1}{18(\log_2 T+1)\log(T/\delta)}$, will not trigger Test 2 as well.*

The proof of this corollary follows immediately from the proof of Lemma 2. Since $\rho(t)$ is non-increasing, this Corollary sets a lower bound on the value of n , such that the non-stationarity detection tests can be triggered. Selecting an n that is not large enough to satisfy Corollary 1 leads to every instance not being able to identify any non-stationarity that may exist. Moreover, even if a proper n is chosen, since m ranges from 0 to n there will exist a set of 2^m -instances that will not contribute in the non-stationarity detection process of Test 1 at all.

3.2 MASTER's Performance Analysis

MASTER's performance is strongly tied to its non-stationarity detection scheme. In particular, the regret bounds given in Theorem 2 of Wei and Luo (2021) are derived through the guarantees provided by Test 1 and Test 2.

To obtain the regret bound of MASTER, Wei and Luo (2021) first derive a regret bound for any 2^n block by dividing the dynamic regret into two terms, and then, using Test 1 and Test 2 to bound the terms appropriately, as presented in Lemma 16 of Appendix C from Wei and Luo (2021).

As seen in Definition 1, the reward is bounded in $[0, 1]$, and thus T is a trivial upper bound for the regret over a horizon of length T . Therefore, any non-trivial bound on the regret should be smaller than T . As we have shown in Section 3.1, the thresholds of the tests are highly conservative, which in turn affects the regret bounds. This is presented in the following Corollary.

Corollary 2 *If Assumption 1 holds, then the $\text{Reg}_D(T)$ bound of MASTER is at least*

$$B_D(T) = 24(\log_2(T) + 1)\log(T)\sqrt{T}(1 + 15\log(T)).$$

Therefore, whenever

$$T \leq 4 \times 10^{14}, \quad T < B_D(T),$$

i.e., the $\text{Reg}_D(T)$ bound of MASTER is trivial.

The proof of the corollary is given in Appendix B.4. Note that similar to the statement of Remark 1, having a more constrained form for $\rho(t)$ will lead to a higher upper bound for Corollary 2. Based on this result, even though the analysis of MASTER ensures that it is order optimal, its performance may not be guaranteed for the majority of practical applications, as its performance bounds are not useful for horizons of practical interest (horizons larger than 4×10^{14} are hardly expected in practical settings).

MASTER may be order optimal in the $\tilde{\mathcal{O}}(\cdot)$ notation, however, it is evident that when the logarithmic terms and the constants are not omitted, the regret bound becomes trivial for an unreasonably large range of horizons. Deriving tight performance bounds is crucial, as a state-of-the-art method should achieve non-trivial regret performance for reasonable horizon values.

3.3 The Choice Of Piecewise Stationary Multi-Armed Bandits

MASTER is a black-box method for NS-RL settings that satisfy its assumptions, and its guarantees must hold across these settings. To verify the performance of MASTER according to our results, it suffices to apply the algorithm to the simplest case considered in Wei and Luo (2021), which corresponds to NS-MABs. The MAB problem is a special case of the RL problem, which can be considered as having only a single, known state (Sutton and Barto, 2018), while the policy set Π corresponds to set of the arms of the bandit. If MASTER performs poorly in NS-MAB settings, its usefulness in more complex scenarios is even more doubtful.

As shown in Wei and Luo (2021), most applications of MASTER focus on bandit settings, except for non-stationary finite-horizon episodic MDPs, while order optimality is not achieved in the infinite-horizon case. Beyond NS-MABs, MASTER has been applied to NS variants of Linear Bandits, Generalized Linear Bandits, and Contextual Bandits, all of which fall under the broader framework of non-stationary sequential stochastic optimization. Since MASTER’s input and output are consistent across bandit algorithms (action and reward), differences in base algorithms do not affect its function as a black-box. However, MASTER’s change-detection mechanism does not generalize well. The form of $\rho(t)$ becomes increasingly restrictive in more complex NS-RL settings (Wei and Luo, 2021). Since all bandit settings share the same fundamental structure, studying them is equivalent to analyzing

harder versions of NS-MABs. Thus, issues in NS-MABs will persist, if not worsen, in more complex settings like MDPs, where MASTER’s change-detection mechanism is expected to fail for even broader horizon ranges. Ultimately, if MASTER’s guarantees break for an important part of NS-RL settings, its claim as a black-box method is weakened.

The difficulty of the NS-MAB problem is associated with the way the non-stationarity evolves. We will consider the case of piecewise stationarity, i.e., when the changes happen at specific points and between consecutive changes the system is stationary. The piecewise stationary assumption is an important case of practical interest, as many real-world problems correspond to scenarios at which the environment remains constant between changes (see e.g. Seznec et al. (2020); Besson et al. (2022)).

We should note that the choice of piecewise stationarity assists MASTER’s assumptions for the existence of intervals that do not have significant non-stationarity, ensuring that $\Delta_{[\text{alg.} s, t]} \leq \rho(t - \text{alg.} s + 1)$, as shown in Lemma 1.

Definition 3 In a piecewise stationary environment, there exist some change-points $\{\nu^{(k)} \in \{1, \dots, T\} : k \in \{1, \dots, N_T\}\}$, where $\nu^{(k)}$ denotes the k^{th} change-point and N_T denotes the number of change-points over the finite horizon T , with $N_T = \sup\{k \in \mathbb{N} : \nu^{(k)} \leq T\}$. Then, for any time steps between consecutive change-points, the reward functions in a piecewise stationary environment remain identical, i.e., for all t, t' such that $t, t' \in \{\nu^{(k-1)}, \dots, \nu^{(k)} - 1\}$ for some $k \in \{1, \dots, N_T\}$, $f_t = f_{t'}$.

Piecewise stationarity allows for easier modeling and adaptation compared to more complex forms of non-stationarity, where changes are continuous. In the case of gradual continuous changes for which $\Delta(t)$ remains bounded away from 0 for all t , $\Delta = \mathcal{O}(T)$, and the optimal regret will clearly be linear. Therefore, under gradual continuous changes, an NS-RL algorithm can attain sub-linear regret only if $\Delta(t)$ converges to 0 at a sufficiently fast rate, i.e., the environment converges to some stationary state. For the piecewise stationary setting we do not require the convergence of $\Delta(t)$ to zero, as long as intervals between change-points grow sufficiently fast with the horizon.

Non-stationarity detection in the piecewise stationary setting is a well-studied problem, with principled algorithmic solutions. The non-stationary detection problem can be formulated as a quickest change detection (QCD) problem, for which methods have been developed that are optimal under various criteria (see e.g. Poor and Hadjiliadis (2008)).

An important aspect of the piecewise stationary setting, is the nature of the change-point process. That is, the change-points can either be deterministic or random. In many works, an environment with deterministic change-points is considered (Liu et al., 2018; Cao et al., 2019; Auer et al., 2019; Besson et al., 2022). In the context of this work, we consider an alternative, random change-point model, which allows us to conduct experiments more effectively in order to assess MASTER’s performance. Thus, we provide the following geometric change-point model.

Definition 4 *The change-points are said to be i.i.d. geometric with parameter $\eta > 0$ if the intervals between consecutive change-points are i.i.d. geometrically distributed with parameter η , i.e., $(\nu^{(k)} - \nu^{(k-1)}) \stackrel{\text{i.i.d.}}{\sim} \text{Geo}(\eta)$.*

The geometric change-point model is less general than the deterministic model, as it assumes a prior on change patterns, whereas the deterministic model, set by an oblivious adversary, makes no such assumptions. While deterministic placements allow for worst-case scenarios, they can vary significantly across methods, making fair comparisons challenging. In contrast, the geometric model introduces variability, enabling broader evaluation and assessing performance across scenarios. Specifically, given a fixed number of change-points, each placement is selected with equal probability. Therefore, the worst-case change-point placement for each algorithm is selected with equal probability, making the geometric model fair for all algorithms. Finally, the geometric change-point model is straightforward to implement, making it a practical tool for generating diverse experimental scenarios and testing robustness under varying conditions. In contrast, the deterministic change-point model may lack clear guidelines for change-point placement, potentially introducing ambiguity.

Note that under geometric change-points, $\Delta(t)$ does not converge to 0 as long as $\eta = \Theta(T^{-\xi})$ for some $\xi \in (0, 1)$. In the context of PS-MABs, the state-of-the-art work of Besson et al. (2022) has combined QCD methods with stationary bandit algorithms, achieving order optimal performance. It is important to note that this method relies on *forced exploration*, i.e., the algorithm requires samples that are not obtained by the bandit procedure itself, in order to ensure that changes are detected effectively.

3.4 Random Restarting Algorithms

Since MASTER’s non-stationarity detection scheme fails for a critical range of values of the horizon T , MASTER effectively reduces to the MALG procedure

from Wei and Luo (2021) described in Section 2.2. MALG can be viewed as a random restarting algorithm with memory. The random restarts arise from random scheduling and switching when shorter instances exist, while the memory component is due to the possible reactivation without a history reset.

Seminal works on the NS-MAB problem and more generally on non-stationary stochastic optimization problem have devised restarting procedures that use the knowledge of the degree of the non-stationarity to attain order optimal performance (Besbes et al., 2014, 2015). Due to the inclusion of the geometric change-point model, we propose a simple random-restarting based procedure, termed as *Random Restarts* (RR) that is order optimal, but *not* prior-free, as a baseline for the setting of PS-MABs.

In the RR procedure, a bandit algorithm that is designed for the stationary setting is used as the base algorithm. A Bernoulli process with a suitably chosen success probability is used to decide when to reset and restart the base algorithm; therefore the times between restarts are i.i.d. geometric random variables.

Theorem 2 *Let the geometric change-point parameter be $\eta = \Theta(T^{-\xi})$, for some $\xi \in (0, 1)$. Set the intervals between random restarts of the RR algorithm to be i.i.d. $\text{Geo}(\eta_R)$, for some $\eta_R \in (0, 1)$ with $\eta_R > \eta$ (order-wise). If the regret of base algorithm is $\mathcal{O}(\text{poly}(\log(T)))$, and $\eta_R = \sqrt{\eta/\text{poly}(\log(T))}$, the RR algorithm is order optimal for PS-MABs.*

The proof of the theorem is included in Appendix A.3, and the detailed description of RR is given in Appendix A.3. Even though RR is designed for geometric change-points, it can be appropriated to the setting of deterministic change-points, by considering a fixed instance of the change-point process.

4 EXPERIMENTAL STUDY

4.1 Setup And Algorithms

The experiments were conducted in four different settings of piecewise stationarity, while keeping the same MAB structure. More specifically, we consider a PS-MAB problem with 5 arms and rewards that are bounded in $[0, 1]$ in each step, for different values of the horizon T . In the simulations, we vary in the way the change-points are generated and in the effect of the changes that occur. Thus, we define the settings:

Deterministic Change-Points. For a given horizon T and a number of change-points $N_C < T$, the change-points are (approximately) evenly spaced.

Geometric Change-Points. The change-points are

geometric with parameter $\eta = T^{-\xi}, \xi \in (0, 1)$.

Uniform Problem. To simulate a realistic scenario, in each change-point a uniformly selected number of arms have their means changed, and the magnitude of each change is uniformly selected in $[0.1, 0.4]$.

Worst-Case Problem. For a worst-case scenario in terms of change detection, at each change, the arm with the lowest mean switches to the highest, while the other arms remain unchanged. Means are initialized at 0.3 plus a maximum difference of 0.005. At each change-point, the lowest mean changes to the highest mean plus a gap uniformly drawn from $[0.005, 0.05]$.

In every problem, if a mean exceeds 1, the means are reset based on the initialization procedure of the problem. More details about the experimental setup are provided in Appendix C.2.

Algorithms. We include the order optimal RR algorithm we propose, which is the only algorithm that has information about the non-stationarity. Since MASTER can be viewed as a random restarting procedure, as it does not have a functioning detection scheme, we employ an algorithm that randomly restarts the klUCB bandit algorithm (Garivier and Cappé, 2011) in each time step with probability $p = 0.05$, termed as $\text{RR}_{p,05}$. With this comparison we aim to show that for practical horizons, MASTER is equivalent to an arbitrarily tuned random-restarting algorithm, without guaranteeing order-optimal performance. The optimal random-restarting probability is given by the RR algorithm. Regarding the algorithms that employ QCD, we first include the state-of-the-art approach of Besson et al. (2022), GLRklUCB, with the Bernoulli GLR change-detector. Finally, to demonstrate the robustness of the QCD-based methods, taking inspiration from GLRklUCB, we combine the Bernoulli GLR change-detector with klUCB and UCB (Auer et al., 2002), due to its lower complexity compared to klUCB. However, unlike GLRklUCB, the latter algorithms do not use forced exploration, and we denote them as QCD+klUCB and QCD+UCB. More details on our QCD-based algorithms are given in Appendix A.4.

Knowledge Of T . Although every method that is considered, except MASTER, implicitly assumes knowledge of the horizon T , they can easily be extended to unknown horizons through the use of the doubling trick (Besson and Kaufmann, 2018). MASTER essentially incorporates the doubling trick in the outer loop of the algorithm, by doubling the block-length every time n is updated. However, MASTER does not necessarily wait until the end of the block to update n if Test 1 or Test 2 are triggered. Thus to ensure a fair and easy comparison with the other methods, we fix $n = \log_2 T$ throughout the operation

of MASTER and do not update n and simply restart MALG if Test 1 or Test 2 are triggered.

4.2 Experimental Results

The experiments were conducted for horizons ranging from $T = 1000$ to $T = 100000$, with increments of approximately 5000, averaged over 4000 independent trials. Due to limited space, we present results for $T = 100000$. For geometric change-points, we use $\xi \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, and for deterministic change-points, $N_C \in \{3163, 1000, 317, 101, 32, 10\}$. The omitted results are available in Appendix C.4.

4.2.1 Dynamic Regret

We compare the dynamic regret of the methods across time steps in the environment. As shown in Figure 1a, for the Uniform problem, MASTER is consistently outperformed by RR. Additionally, the performance of $\text{RR}_{p,05}$ closely mirrors that of MASTER, reinforcing the observation about random restarting with memory. Regarding the QCD-based methods, GLRklUCB achieves order optimal performance, outperforming RR in nearly every instance except one, despite RR’s knowledge of the non-stationarity. Moreover, GLRklUCB significantly surpasses both MASTER and $\text{RR}_{p,05}$. Surprisingly, QCD+ not only outperforms MASTER but also exceeds the performance of RR and GLRklUCB. A similar trend is evident in the Worst-Case problem (Figure 1b), where MASTER is outperformed by every method, including $\text{RR}_{p,05}$ at higher levels of non-stationarity. Across both problems, the QCD-based methods consistently demonstrate the best performance.

To assess the robustness of the methods, we plot the final dynamic regret ($\text{Reg}_D(T)$) as the non-stationarity decreases. A robust method should exhibit lower final dynamic regret with decreasing non-stationarity, while maintaining small variance. As shown in Figure 2, MASTER’s robustness is comparable to that of $\text{RR}_{p,05}$, and it is significantly worse than that of RR. In contrast, the QCD-based methods demonstrate superior robustness compared to MASTER and the random restarting algorithms. The final regret of the QCD-based methods display lower variance than that of MASTER and $\text{RR}_{p,05}$, and in certain cases, lower than RR. Thus, the QCD-based methods show greater robustness, achieving better regret performance and lower variance as the non-stationarity decreases. Among them, QCD+UCB achieves the lowest variance.

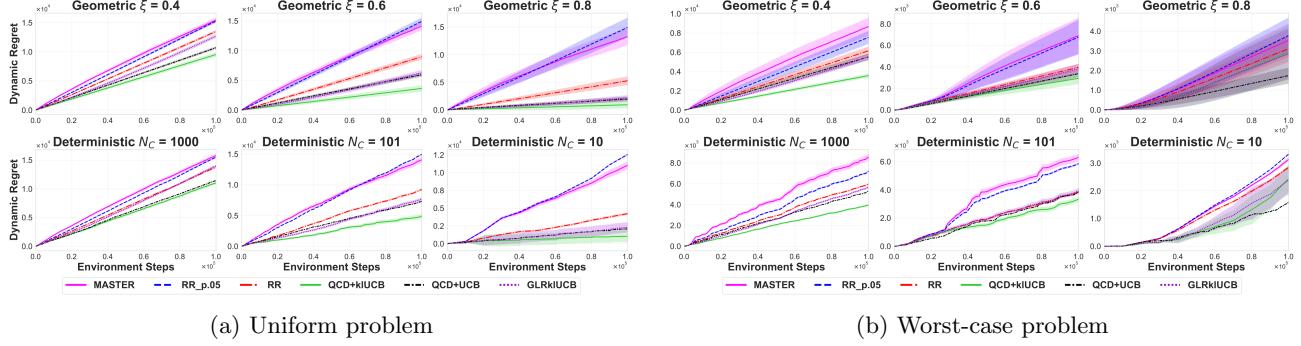


Figure 1: Dynamic regret plots versus the time steps for $T = 100000$, averaged over 4000 independent runs. The case of the geometric change-points is on the top row for $\xi = 0.4, 0.6, 0.8$ and the case of the deterministic change-points on the bottom row for $N_C = 1000, 101, 10$. Left: Uniform problem. Right: Worst-case problem.

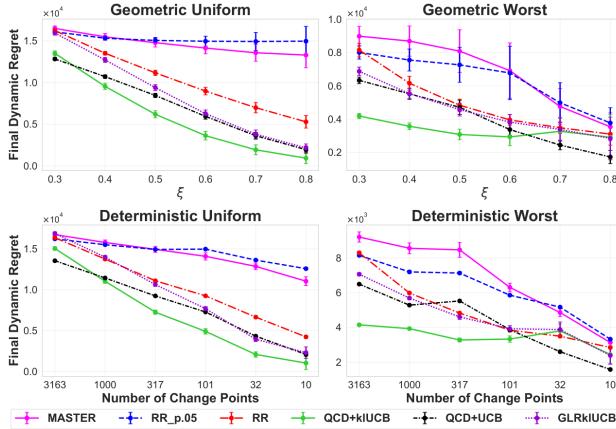


Figure 2: Final dynamic regret versus the decrease in non-stationarity, for $T = 100000$, 4000 runs.

4.2.2 Detection And Time Efficiency

To verify our theory, we examine the non-stationarity detection mechanism of MASTER. As seen in Table 1, MASTER indeed does not declare any changes for any of the problems, confirming the result in Theorem 1. Conversely, the QCD-based approaches are able to declare a significant number of the change-points. The strong performance of QCD-based approaches likely stems from their ability to restart swiftly after change-points, a guarantee that MASTER’s non-stationarity detection mechanism lacks.

Lastly, we evaluate the computational efficiency of the algorithms based on the time required per run. As shown in Table 2, MASTER is less efficient compared to the QCD-based methods. As provided in Appendix A.1, MASTER’s inefficiency is largely due to its scheduling process, which becomes even more demanding as the horizon increases. Due to its similar performance with MASTER, we include $RR_{p.05}$, which

is 2 to 3 times more efficient than the former. Notably, using the klUCB algorithm increases the complexity of both the random-restarting and the QCD-based algorithms, resulting in longer run times. In contrast, opting for the UCB algorithm significantly reduces the runtime, offering speed improvements over MASTER by a factor of 5 to 37 as non-stationarity increases.

Table 1: Average Number of Changes Declared By Algorithms For Different Change-Points, $T = 100000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	129.43	137.21	99.21	54.05	23.78	8.88
GLRklUCB	Worst	147.03	79.08	45.28	20.73	6.07	1.46
QCD+UCB	Unif	121.62	139.51	104.03	58.01	25.35	9.23
QCD+UCB	Worst	150.80	110.35	79.12	36.51	11.77	3.77
QCD+klUCB	Unif	134.84	131.53	91.32	48.76	21.53	8.22
QCD+klUCB	Worst	131.94	64.20	31.33	12.93	4.59	1.27

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	3163	1000	317	101	32	10
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	98.45	136.94	119.82	69.49	27.38	8.31
GLRklUCB	Worst	148.35	83.78	51.12	21.52	6.74	1.19
QCD+UCB	Unif	84.00	149.00	122.00	83.00	28.00	9.00
QCD+UCB	Worst	141.00	95.00	102.00	36.00	13.00	4.00
QCD+klUCB	Unif	104.45	134.02	113.89	62.68	25.81	7.95
QCD+klUCB	Worst	130.42	71.08	34.60	14.37	4.38	1.11

5 SUMMARY AND OUTLOOK

In this work, we examined the feasibility of prior-free black-box NS-RL in practice, focusing on the only existing method that is order optimal: MASTER. We showed that there is a large span of horizon values (up to at least 1.24 billion) where MASTER’s non-stationarity detection mechanism is ineffective, reducing its performance to that of a random restarting approach (with some memory). Furthermore, we demonstrated that the performance bounds for MASTER

Table 2: Average Time Per Run (sec) For Different Change-Points, $T=100000$, 4000 Runs.

		(a) Random Change-Points					
Algorithm	PB	$\xi=0.3$	$\xi=0.4$	$\xi=0.5$	$\xi=0.6$	$\xi=0.7$	$\xi=0.8$
MASTER	Unif	3.33 ± 0.03	3.34 ± 0.02	3.34 ± 0.04	3.33 ± 0.03	3.36 ± 0.04	3.35 ± 0.20
MASTER	Worst	3.32 ± 0.03	3.32 ± 0.01	3.33 ± 0.02	3.33 ± 0.05	3.35 ± 0.08	3.36 ± 0.15
RR_p.05	Unif	1.13 ± 0.02	1.12 ± 0.01	1.12 ± 0.02	1.12 ± 0.03	1.12 ± 0.05	1.12 ± 0.07
RR_p.05	Worst	0.96 ± 0.01	0.94 ± 0.02	0.94 ± 0.03	0.96 ± 0.05	1.03 ± 0.09	1.23 ± 0.05
GLRklUCB	Unif	1.38 ± 0.01	1.37 ± 0.01	1.45 ± 0.02	1.65 ± 0.05	2.18 ± 0.20	3.76 ± 1.02
GLRklUCB	Worst	1.33 ± 0.03	1.49 ± 0.04	1.68 ± 0.11	2.14 ± 0.37	3.27 ± 0.53	6.08 ± 2.06
QCD+UCB	Unif	0.09 ± 0.01	0.08 ± 0.01	0.11 ± 0.01	0.24 ± 0.04	0.72 ± 0.20	2.24 ± 1.04
QCD+UCB	Worst	0.08 ± 0.01	0.10 ± 0.01	0.16 ± 0.02	0.33 ± 0.07	0.68 ± 0.19	1.65 ± 0.71
QCD+klUCB	Unif	1.68 ± 0.01	1.68 ± 0.01	1.73 ± 0.01	1.90 ± 0.04	2.44 ± 0.22	4.12 ± 1.09
QCD+klUCB	Worst	1.67 ± 0.01	1.80 ± 0.06	2.06 ± 0.28	2.71 ± 0.84	3.60 ± 0.55	6.42 ± 2.13

		(b) Deterministic Change-Points					
Algorithm	PB	$N_C=3163$	$N_C=1000$	$N_C=317$	$N_C=101$	$N_C=32$	$N_C=10$
MASTER	Unif	3.35 ± 0.42	3.32 ± 0.04	3.35 ± 0.22	3.32 ± 0.01	3.32 ± 0.15	3.32 ± 0.02
MASTER	Worst	3.38 ± 0.09	3.38 ± 0.10	3.40 ± 0.08	3.38 ± 0.06	3.37 ± 0.03	3.35 ± 0.07
RR_p.05	Unif	1.15 ± 0.26	1.14 ± 0.04	1.16 ± 0.12	1.12 ± 0.01	1.10 ± 0.04	1.08 ± 0.11
RR_p.05	Worst	0.97 ± 0.05	0.95 ± 0.05	0.94 ± 0.02	1.00 ± 0.01	1.04 ± 0.01	1.24 ± 0.01
GLRklUCB	Unif	1.46 ± 0.28	1.38 ± 0.03	1.43 ± 0.17	1.52 ± 0.01	1.83 ± 0.06	2.67 ± 0.20
GLRklUCB	Worst	1.33 ± 0.06	1.48 ± 0.07	1.66 ± 0.06	2.04 ± 0.11	2.97 ± 0.17	5.06 ± 0.94
QCD+UCB	Unif	0.12 ± 0.08	0.07 ± 0.01	0.13 ± 0.05	0.13 ± 0.00	0.38 ± 0.04	1.09 ± 0.14
QCD+UCB	Worst	0.09 ± 0.02	0.11 ± 0.03	0.16 ± 0.02	0.41 ± 0.01	0.73 ± 0.00	1.05 ± 0.03
QCD+klUCB	Unif	1.71 ± 0.29	1.69 ± 0.03	1.73 ± 0.17	1.78 ± 0.01	2.02 ± 0.07	2.88 ± 0.17
QCD+klUCB	Worst	1.68 ± 0.06	1.77 ± 0.07	1.95 ± 0.07	2.35 ± 0.16	3.24 ± 0.22	5.09 ± 0.77

given in Wei and Luo (2021) are not useful for an unreasonably large range of horizon values (up to a horizon of at least 4×10^{14}). To validate these findings, we considered the special NS-RL setting of PS-MABs, incorporating both random and deterministic change-points. To establish a suitable random restarting baseline for our experiments, we proposed an order optimal random restarting algorithm tailored for PS-MABs. Our experiments confirmed MASTER’s limitations, while showing that prior-free methods utilizing QCD-based mechanisms effectively detect changes, yielding superior performance and robustness.

These observations highlight a gap in prior-free, black-box NS-RL: the need for an order optimal algorithm that is not only theoretically sound but also practically feasible. The theoretical analysis of MASTER provided in Wei and Luo (2021) is a valuable starting point, and future work could focus on developing approaches that mimic MASTER’s multi-scale guarantees. On the other hand, the insight that effective non-stationarity detection is key to good performance, paves the way for developing principled change detection algorithms, based on QCD theory, that can be generalized to broader NS-RL settings.

Acknowledgments

The authors thank Subhonmesh Bose and the anonymous reviewers for the insightful discussions and comments. This work was supported by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196, and by a grant from the C3.ai Digital Transformation Institute, through the University of Illinois at Urbana-Champaign.

References

- Abbasi-Yadkori, Y., György, A., and Lazić, N. (2023). A new look at dynamic regret for non-stationary stochastic bandits. *Journal of Machine Learning Research*, 24(288):1–37.
- Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.
- Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. (2014). Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*, pages 1638–1646. PMLR.
- Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256.
- Auer, P., Gajane, P., and Ortner, R. (2019). Adaptively tracking the best bandit arm with an unknown number of distribution changes. In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 138–158. PMLR.
- Besbes, O., Gur, Y., and Zeevi, A. (2014). Stochastic multi-armed-bandit problem with non-stationary rewards. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.
- Besbes, O., Gur, Y., and Zeevi, A. (2015). Non-

- stationary stochastic optimization. *Operations research*, 63(5):1227–1244.
- Besson, L. and Kaufmann, E. (2018). What doubling tricks can and can’t do for multi-armed bandits.
- Besson, L., Kaufmann, E., Maillard, O.-A., and Seznec, J. (2022). Efficient change-point detection for tackling piecewise-stationary bandits. *Journal of Machine Learning Research*, 23(77):1–40.
- Cao, Y., Wen, Z., Kveton, B., and Xie, Y. (2019). Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 418–427. PMLR.
- Chen, L. and Luo, H. (2022). Near-optimal goal-oriented reinforcement learning in non-stationary environments. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 33973–33984. Curran Associates, Inc.
- Chen, Y., Lee, C.-W., Luo, H., and Wei, C.-Y. (2019). A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 696–726. PMLR.
- Cheng, Y., Yang, J., and Liang, Y. (2023). Provably efficient algorithm for nonstationary low-rank mdps. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 6330–6372. Curran Associates, Inc.
- Cheung, W. C., Simchi-Levi, D., and Zhu, R. (2019). Learning to optimize under non-stationarity. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1079–1087. PMLR.
- Cheung, W. C., Simchi-Levi, D., and Zhu, R. (2020a). Reinforcement learning for non-stationary Markov decision processes: The blessing of (More) optimism. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1843–1854. PMLR.
- Cheung, W. C., Simchi-Levi, D., and Zhu, R. (2020b). Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism. In *International Conference on Machine Learning*, pages 1843–1854. PMLR.
- Faury, L., Russac, Y., Abeille, M., and Calauzènes, C. (2021). Regret bounds for generalized linear bandits under parameter drift. *CoRR*, abs/2103.05750.
- Filippi, S., Cappe, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits: The generalized linear case. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.
- Garivier, A. and Cappé, O. (2011). The kl-ucb algorithm for bounded stochastic bandits and beyond. In Kakade, S. M. and von Luxburg, U., editors, *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pages 359–376, Budapest, Hungary. PMLR.
- Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. (2018). Is q-learning provably efficient? In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on learning theory*, pages 2137–2143. PMLR.
- Liu, F., Lee, J., and Shroff, N. (2018). A change-detection based framework for piecewise-stationary multi-armed bandit problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).
- Mao, W., Zhang, K., Zhu, R., Simchi-Levi, D., and Basar, T. (2021). Near-optimal model-free reinforcement learning in non-stationary episodic mdps. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7447–7458. PMLR.
- Peng, B. and Papadimitriou, C. (2024). The complexity of non-stationary reinforcement learning. In Vernade, C. and Hsu, D., editors, *Proceedings of The 35th International Conference on Algorithmic Learning Theory*, volume 237 of *Proceedings of Machine Learning Research*, pages 972–996. PMLR.
- Poor, H. V. and Hadjiliadis, O. (2008). *Quickest Detection*. Cambridge University Press.
- Russac, Y., Cappé, O., and Garivier, A. (2020). Algorithms for non-stationary generalized linear bandits. *CoRR*, abs/2003.10113.
- Seznec, J., Menard, P., Lazaric, A., and Valko, M. (2020). A single algorithm for both restless and rested rotting bandits. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third*

- International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3784–3794. PMLR.
- Simchi-Levi, D. and Xu, Y. (2022). Bypassing the monster: A faster and simpler optimal algorithm for contextual bandits under realizability. *Mathematics of Operations Research*, 47(3):1904–1931.
- Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Touati, A. and Vincent, P. (2020). Efficient learning in non-stationary linear markov decision processes. *CoRR*, abs/2010.12870.
- Veeravalli, V. V. and Banerjee, T. (2014). Chapter 6 - quickest change detection. In Zoubir, A. M., Viberg, M., Chellappa, R., and Theodoridis, S., editors, *Academic Press Library in Signal Processing: Volume 3*, volume 3 of *Academic Press Library in Signal Processing*, pages 209–255. Elsevier.
- Wang, J., Zhao, P., and Zhou, Z.-H. (2023). Revisiting weighted strategy for non-stationary parametric bandits. In Ruiz, F., Dy, J., and van de Meent, J.-W., editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 7913–7942. PMLR.
- Wei, C.-Y. and Luo, H. (2021). Non-stationary reinforcement learning without prior knowledge: an optimal black-box approach. In Belkin, M. and Kpotufe, S., editors, *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4300–4354. PMLR.
- Zhou, H., Chen, J., Varshney, L. R., and Jagmohan, A. (2022). Nonstationary reinforcement learning with linear function approximation. *Transactions on Machine Learning Research*.

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Please refer to Sections 2 and 3 for the mathematical setting, assumptions, algorithms and models. Also, refer to Section 4 and the Appendix for information on the algorithms used in the experiments.
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Please refer to Sections 3 and 4 and the Appendix.
- (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Please refer to the Appendix and the link to the code.
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. [Yes] Please refer to Sections 2 and 3 and the Appendix.
 - (b) Complete proofs of all theoretical results. [Yes] Please refer to the Appendix.
 - (c) Clear explanations of any assumptions. [Yes] Please refer to Sections 2 and 3 and the Appendix.
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] Please refer to the Appendix and the link to the code.
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Please refer to Section 4 and the Appendix.
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Please refer to Section 4 and the Appendix.
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Please refer to the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. [Not Applicable]
 - (b) The license information of the assets, if applicable. [Not Applicable]
 - (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
 - (d) Information about consent from data providers/curators. [Not Applicable]
 - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:

Is Prior-Free Black-Box Non-Stationary Reinforcement Learning Feasible?

- (a) The full text of instructions given to participants and screenshots. [Not Applicable]
- (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
- (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A ALGORITHMIC IMPLEMENTATIONS

A.1 The MASTER Algorithm

Algorithm 1 Multi-scale Algorithm with Stationary TEsts and Restarts

Input: $\rho(\cdot)$, input algorithm ALG.

Initialization: $t \leftarrow 1$.

```

1: for  $n = 0, 1, \dots$  do
2:    $t_n \leftarrow t$ .
3:   for  $m = 0, 1, \dots, n$  do
4:     for  $k = 0, 1, \dots, 2^{n-m} - 1$  do
5:        $I_k \leftarrow [t_n + k2^m, \dots, t_n + (k+1)2^m - 1]$ 
6:       With probability  $\frac{\rho(2^n)}{\rho(2^m)}$ , schedule an instance of ALG,  $alg$ , to  $I_k$ 
        with start  $alg.s = t_n + k2^m$  and end  $alg.e = t_n + (k+1)2^m - 1$ .
7:     end for
8:   end for
9:   for  $t = t_n, t_n + 1, \dots, t_n + 2^n - 1$  do
10:    if  $t > T$  then
11:      End the algorithm.
12:    end if
13:    Let  $alg$  be the instance assigned to the shortest  $I_k$ , for all  $k$ , that contains  $t$ .
14:    Receive  $\tilde{f}_t$  and  $\pi_t$  from  $alg$ .
15:    Execute  $\pi_t$  and receive the reward  $R_t$ .
16:     $\tilde{g}_t \leftarrow \tilde{f}_t$ .
17:    if (Test 1)  $t = alg'.e$  for some  $2^m$ -length instance  $alg'$  and  $\frac{1}{2^m} \sum_{\tau=alg'.s}^{alg'.e} R_\tau - \min_{\tau \in [t_n, t]} \tilde{g}_\tau \geq 54(\log_2 T + 1) \log(T/\delta)\rho(2^m)$  then
18:      Non-stationarity detected. Restart from Line 1.
19:    end if
20:    if (Test 2)  $\frac{1}{t-t_n+1} \sum_{\tau=t_n}^t (\tilde{g}_\tau - R_\tau) \geq 18(\log_2 T + 1) \log(T/\delta)\rho(t - t_n + 1)$  then
21:      Non-stationarity detected. Restart from Line 1.
22:    end if
23:  end for
24: end for

```

From Line 2 to Line 8, MASTER first partitions the interval of length 2^n into 2^{n-m} sub-intervals of length 2^m , which are denoted by I_k . Then, each sub-interval is assigned an instance of ALG with probability $\frac{\rho(2^n)}{\rho(2^m)}$ independently. Note that there is always an instance of ALG assigned to the original interval of length 2^n since $\frac{\rho(2^n)}{\rho(2^n)} = 1$. At Line 13, MASTER selects the instance with the shortest interval to run and pauses all the other instances. Therefore, whenever there is a shorter instance compared to the current active one at the next time-step, MASTER makes the current instance inactive and the shorter one active. If any previously active instance becomes the one with the smallest length, it resumes and runs with the data prior to the pause. Thus, the MASTER algorithm is a restarting scheme with memory. Moreover, MASTER restarts if it reaches the end of the 2^n block, and then n is incremented by 1 for the next block. MASTER can also restart through Test 1 and 2, which are used to detect non-stationarity. For a more detailed discussion regarding the MASTER algorithm, we refer the reader to Wei and Luo (2021).

A.2 Multi-Armed Bandit Preliminaries

In order to present the different algorithms for Piecewise Stationary Multi-Armed Bandits (PS-MABs), we first provide some notation that will be used in the context of MABs.

We assume that a (stationary) bandit algorithm, \mathcal{B} , maintains a history list for its arms, which includes the number of pulls for each arm, along with the rewards that are obtained from each pull. We denote this history by $\mathcal{H}_{\mathcal{B}}$. Based on Definition 1, we have that the policy set Π corresponds to the set of arms, which is given by

$\{1, \dots, A\}$ when we have a total of A arms. Finally, we use the notation $\mathcal{H}_{\mathcal{B},a}$ for any arm $a \in \{1, \dots, A\}$ to denote the bandit history that corresponds to arm a .

A.3 The RR Algorithm

In this section, we present the Random Restarts (RR) algorithm for PS-MABs, which is given as follows.

Algorithm 2 Random Restarts

Input: base bandit algorithm \mathcal{B} with $\text{poly}(\log(T))$ regret, number of arms A , geometric distribution parameter η , horizon T

Initialization: set random restart intervals as i.i.d. $\text{Geo}(\sqrt{\eta/\text{poly}(\log(T))})$, history list $\mathcal{H}_{\mathcal{B}} \leftarrow \emptyset$.

```

1: for  $t = 1, 2, \dots, T$  do
2:   if  $t$  is a random restart point then
3:      $\mathcal{H}_{\mathcal{B}} \leftarrow \emptyset$ .
4:   end if
5:    $A_t \leftarrow \mathcal{B}(\mathcal{H}_{\mathcal{B}})$  (Bandit Algorithm).
6:   Pull arm  $A_t$  and receive reward  $R_t$ .
7:   Add  $(A_t, R_t)$  into  $\mathcal{H}_{\mathcal{B}}$ .
8: end for

```

A.4 The QCD+ Algorithm

Before we present the algorithm, we introduce the notion of a change detector \mathcal{D} . The change detector takes as input the bandit history for a specific arm a , $\mathcal{H}_{\mathcal{B},a}$, and yields “True” if, according to its test, there has been a change in that specific arm so far, and yields “False” otherwise. The QCD+ algorithm is an adaptation of the GLRklUCB algorithm, proposed in Besson et al. (2022), and is obtained by removing the forced exploration and the local restart conditions. Thus, the algorithm is given as follows.

Algorithm 3 QCD+

Input: change detector \mathcal{D} , bandit algorithm \mathcal{B} , horizon T , number of arms A

Initialization: history list $\mathcal{H}_{\mathcal{B}} \leftarrow \emptyset$.

```

1: for  $t = 1, 2, \dots, T$  do
2:    $A_t \leftarrow \mathcal{B}(\mathcal{H}_{\mathcal{B}})$  (Bandit Algorithm).
3:   Pull arm  $A_t$  and receive reward  $R_t$ .
4:   Add  $(A_t, R_t)$  into  $\mathcal{H}_{\mathcal{B}}$ .
5:   if  $\mathcal{D}(\mathcal{H}_{\mathcal{B},A_t}) = \text{True}$  (Change Detected) then
6:      $\mathcal{H}_{\mathcal{B}} \leftarrow \emptyset$ .
7:   end if
8: end for

```

B THEORETICAL PROOFS

B.1 Theory Preliminaries

To enhance the clarity and flow of the proofs, we include key points from the main paper.

Definition 1 Assume an agent that interacts with an environment through a behavior-policy set Π over a horizon T , that is, a fixed number of time steps. The environment has a collection of T reward functions $f_1, \dots, f_T : \Pi \rightarrow [0, 1]$, unknown to the agent, that model the environment’s feedback. In each time step $t = 1, \dots, T$, the agent selects a policy $\pi_t \in \Pi$ and receives the feedback in the form of a random reward $R_t \in [0, 1]$ with mean $f_t(\pi_t) = \mathbb{E}[R_t | \pi_t]$. The goal is to minimize the dynamic regret, $\text{Reg}_D(T) = \mathbb{E} \left[\sum_{t=1}^T (f_t^* - f_t(\pi_t)) \right]$, where $f_t^* = \max_{\pi \in \Pi} f_t(\pi)$ is the best expected reward at time t .

Definition 2 (Following Wei and Luo (2021).) $\Delta : [T] \rightarrow \mathbb{R}$ is a non-stationarity measure, if it satisfies $\Delta(t) \geq \max_{\pi \in \Pi} |f_t(\pi) - f_{t+1}(\pi)|$, for all t . For interval $[s, e]$, the cumulative non-stationarity is given as $\Delta_{[s,e]} = \sum_{\tau=s}^{e-1} \Delta(\tau)$, with $\Delta_{[s,s]} = 0$, and the number of changes (+1) as $L_{[s,e]} = 1 + \sum_{\tau=s}^{e-1} \mathbb{1}[\Delta(\tau) \neq 0]$.

Assumption 1 (Following Wei and Luo (2021).) ALG outputs $\tilde{f}_t \in [0, 1]$ at the beginning of each time step t . Suppose there exists a non-increasing function $\rho : [T] \rightarrow \mathbb{R}$ such that $\rho(t) \geq 1/\sqrt{t}$ and $C(t) = t\rho(t)$ is non-decreasing. Additionally, assume $\Delta(t)$ to be a non-stationary measure. Then, for any $t \in [T]$ such that $\Delta_{[1,t]} \leq \rho(t)$, the following holds with probability at least $1 - \delta/T$:

1. $\tilde{f}_t \geq \min_{\tau \in [1,t]} f_\tau^* - \Delta_{[1,t]}$, and
2. $\frac{1}{t} \sum_{\tau=1}^t (\tilde{f}_\tau - R_\tau) \leq \rho(t) + \Delta_{[1,t]}$.

Lemma 1 (Adapted from Wei and Luo (2021).) Let $\hat{n} = \log_2 T + 1$, $\hat{\rho}(t) = 6\hat{n} \log(T/\delta)\rho(t)$ and $t' = t - \text{alg.s} + 1$. Suppose Assumption 1 holds for ALG, and that $n \leq \log_2 T$. Then, for any instance alg, with start at alg.s and finish at alg.e, that is maintained by MALG and any $t \in [\text{alg.s}, \text{alg.e}]$ that satisfies $\Delta_{[\text{alg.s},t]} \leq \rho(t')$, MALG satisfies with probability at least $1 - \delta/T$:

1. $\tilde{g}_t \geq \min_{\tau \in [\text{alg.s},t]} f_\tau^* - \Delta_{[\text{alg.s},t]}$, and
2. $\frac{1}{t'} \sum_{\tau=\text{alg.s}}^t (\tilde{g}_\tau - R_\tau) \leq \hat{\rho}(t') + \hat{n}\Delta_{[\text{alg.s},t]}$.

Finally, we restate the soft-Oh notation, $\tilde{\mathcal{O}}(\cdot)$, and introduce the soft-Omega notation, $\tilde{\Omega}(\cdot)$. We write $h_1(x) = \tilde{\mathcal{O}}(h_2(x))$ with high probability, if $h_1(x) = \mathcal{O}(\text{poly}(\log(T/\delta))h_2(x))$ with probability $1 - \delta$ for some $\delta \in (0, 1)$. On the other hand, $h_1(x) = \tilde{\Omega}(h_2(x))$ with high probability, if $h_1(x) = \Omega(\text{poly}(\log(T/\delta))h_2(x))$ with probability $1 - \delta$ for some $\delta \in (0, 1)$

B.2 Proof Of Lemma 2

Lemma 2 Test 1 will not be triggered for any value of the horizon that satisfies $\rho(T) > \frac{1}{54(\log_2 T + 1) \log(T/\delta)}$ and Test 2 will also not be triggered if $\rho(T) > \frac{1}{18(\log_2 T + 1) \log(T/\delta)}$.

Proof:

According to Test 1, in order to detect non-stationarity, the following condition needs to hold,

$$\frac{1}{2^m} \sum_{\tau=\text{alg.s}}^{\text{alg.e}} R_\tau - \min_{\tau \in [t_n, t]} \tilde{g}_\tau \geq 54(\log_2 T + 1) \log(T/\delta) \rho(2^m). \quad (1)$$

To effectively detect using Test 1, we need to ensure that the LHS in Equation (1) is indeed greater than the RHS. Thereupon, when the LHS is maximized and the RHS is minimized, Test 1 must always work. By definition, \tilde{g}_τ is the active instance of \tilde{f}_τ at round τ , and hence $\tilde{g}_\tau \in [0, 1]$, while we have that $R_\tau \in [0, 1]$. Additionally, since $\rho(t)$ is a non-increasing function and because $m = 0, 1, \dots, n$, its minimum value is achieved when $m = n$. Therefore, in the most ideal detection scenario, which is not necessarily met in practice, the following needs to hold,

$$\frac{1}{2^m} \sum_{\tau=\text{alg.s}}^{\text{alg.e}} 1 \geq 54(\log_2 T + 1) \log(T/\delta) \rho(2^n).$$

Note that if this condition is not met, then, with absolute certainty, no non-stationarity will be detected by Test 1. Finally, since the largest value that n can have, to ensure the performance guarantees of MASTER, is equal to $\log_2 T$, we can minimize the RHS even further. In particular, since the alg instance has length equal to 2^m ,

$$1 \geq 54(\log_2 T + 1) \log(T/\delta) \rho(2^{\log_2 T}),$$

which implies,

$$\rho(T) \leq \frac{1}{54(\log_2 T + 1) \log(T/\delta)}. \quad (2)$$

Regarding Test 2, the same idea as before is applied,

$$\frac{1}{t - t_n + 1} \sum_{\tau=t_n}^t (\tilde{g}_\tau - R_\tau) \geq 18(\log_2 T + 1) \log(T/\delta) \rho(t')$$

Since $\tilde{g}_\tau, R_\tau \in [0, 1]$ for any $\tau \in \{1, \dots, T\}$, we have

$$\frac{1}{t - t_n + 1} \sum_{\tau=t_n}^t 1 = 1 \geq 18(\log_2 T + 1) \log(T/\delta) \rho(2^n).$$

Thus, we have

$$\rho(T) \leq \frac{1}{18(\log_2 T + 1) \log(T/\delta)}. \quad (3)$$

Hence, if the opposite of the inequalities (2) and (3) hold, neither Test 1 nor Test 2 will be triggered for any value of the horizon, which concludes the proof.

B.3 Proof Of Theorem 1

We first restate the theorem.

Theorem 1 *If Assumption 1 is satisfied, the thresholds of Test 1 and Test 2 can be crossed only if $\delta \geq T \exp\left(-\frac{\sqrt{T}}{54(\log_2 T + 1)}\right)$, and since $\delta < 1$, T must be at least 1.24 billion.*

Proof:

According to Lemma 2, in order to ensure that Test 1 and Test 2 can be triggered, the following conditions need to hold:

$$\rho(T) \leq \frac{1}{54(\log_2 T + 1) \log(T/\delta)} \quad \text{and} \quad \rho(T) \leq \frac{1}{18(\log_2 T + 1) \log(T/\delta)}.$$

Based on Assumption 1, the function ρ needs to satisfy, $\rho(t) \geq 1/\sqrt{t}$. Thereupon, it is evident that if the following conditions hold, then Test 1 and Test 2 cannot be triggered:

$$\frac{1}{\sqrt{T}} > \frac{1}{54(\log_2 T + 1) \log(T/\delta)} \quad \text{and} \quad \frac{1}{\sqrt{T}} > \frac{1}{18(\log_2 T + 1) \log(T/\delta)}$$

Solving with respect to δ , we have that,

$$\delta < T \exp\left(-\frac{\sqrt{T}}{54 \log_2(T + 1)}\right) \quad \text{and} \quad \delta < T \exp\left(-\frac{\sqrt{T}}{18 \log_2(T + 1)}\right).$$

Thereupon, in order to ensure that both tests can be triggered, the following must hold:

$$\delta \geq T \exp\left(-\frac{\sqrt{T}}{54 \log_2(T + 1)}\right) \quad \text{and} \quad \delta \geq T \exp\left(-\frac{\sqrt{T}}{18 \log_2(T + 1)}\right).$$

Thus, by selecting the more constrained inequality, the first part of the theorem follows.

Thereupon, the following inequality needs to hold,

$$\delta \geq T \exp\left(-\frac{\sqrt{T}}{54 \log_2(T + 1)}\right).$$

However, recall that the for the probabilistic parameter $\delta \in (0, 1)$. Hence, since δ is trivially upper-bounded by 1, the following needs to hold,

$$1 > T \exp\left(-\frac{\sqrt{T}}{54 \log_2(T+1)}\right)$$

which we can rearrange as

$$\exp\left(\frac{\sqrt{T}}{54 \log_2(T+1)}\right) > T. \quad (4)$$

To make sure that either Test 1 or Test 2 can indeed detect the non-stationarity, we need to guarantee that the LHS in (4) is greater than the RHS in (4). Solving this inequality numerically, we get that T needs to be at least equal to 1246257461, which gives the second part of the theorem.

B.4 Proof Of Corollary 2

We first restate Corollary 2.

Corollary 2 *If Assumption 1 holds, then the $\text{Reg}_D(T)$ bound of MASTER is at least*

$$B_D(T) = 24(\log_2(T) + 1) \log(T) \sqrt{T} (1 + 15 \log(T)).$$

Therefore, whenever

$$T \leq 4 \times 10^{14}, \quad T < B_D(T),$$

i.e., the $\text{Reg}_D(T)$ bound of MASTER is trivial.

Proof:

To prove the corollary, we follow the approach used in the proof of the regret bound for MASTER, as presented in Theorem 2 of Wei and Luo (2021). The proof begins by deriving a regret bound for any length- 2^n block, which is then tailored to a specific form of $C(t)$. Since MASTER employs the two non-stationarity detection tests that can trigger restarts, the regret between two restarts is appropriately upper bounded using the refined form of the length- 2^n block regret bound. Thus, the regret bound for any length- 2^n block has to be tight, otherwise, the final regret bound would be sub-optimal. This bound essentially serves as the foundational bound for the algorithm, with the final regret bound constructed from the exact or upper bounds of its components.

The regret bound for any length- 2^n block is provided in Lemma 4 from Wei and Luo (2021) and its proof is given in Lemma 16 from Wei and Luo (2021). We present Lemma 16 from Wei and Luo (2021) below.

Lemma 3 *(Adapted from Wei and Luo (2021).) Let the high-probability events described in Lemma 1 hold. Then with high probability, the following holds for any length- 2^n block of MASTER,*

$$\sum_{\tau=t_n}^{E_n} (f_\tau^* - R_\tau) \leq 4\hat{C}(2^n) + 96\hat{n} \sum_{i=1}^{\ell} \hat{C}(|\mathcal{I}_i|) + 60 \sum_{m=0}^n \frac{\rho(2^m)}{\rho(2^n)} \hat{C}(2^m) \log(T/\delta).$$

where $\hat{C}(t) = t\hat{\rho}(t)$, t_n is the start of the block and $E_n \leq t_n + 2^n - 1$ is the end of the block and \mathcal{I}_i are consecutive intervals that partition $[t_n, E_n]$ such that $\Delta_{\mathcal{I}_i} \leq \rho(|\mathcal{I}_i|)$ for all i .

Note that the dynamic regret in Wei and Luo (2021) is given with respect to R_t rather than its expectation as shown in Definition 1. This is evident in Lemma 3, as $\sum_{\tau=t_n}^{E_n} (f_\tau^* - R_\tau)$ is upper-bounded with high probability, rather than the expectation $\mathbb{E}[\sum_{\tau=t_n}^{E_n} (f_\tau^* - R_\tau)]$ being upper-bounded. The authors of Wei and Luo (2021) choose to express the dynamic regret this way, even though they compare with approaches that follow Definition 1 (see e.g. Auer et al. (2019); Cheung et al. (2020b)). However, this can trivially be extended to Definition 1, using the Azuma-Hoeffding inequality, as will be discussed next.

According to Lemma 3, notice that $E_n \leq t_n + 2^n - 1$, since the block may terminate earlier due to the non-stationarity detection tests being triggered. However, the regret bound is independent of t_n and E_n , as it essentially bounds the entire length- 2^n block. Thus, let N_B denote the random number of blocks that cover

the entire horizon T . We denote each block by B_i , for all $i \in \{0, 1, \dots, N_B - 1\}$, where $B_i = [t_i, E_i]$, and $t_0 = 1, E_{N_B-1} = T$. Thus, it is evident that the entire regret over the horizon T can be expressed as follows,

$$\sum_{n=0}^{N_B-1} \sum_{\tau=t_n}^{E_n} (f_\tau^* - R_\tau) = \sum_{\tau=1}^T (f_\tau^* - R_\tau).$$

Then, by bounding the regret of each different block using Lemma 3, we can provide the following (tight) bound for the regret,

$$\sum_{\tau=1}^T (f_\tau^* - R_\tau) \leq \sum_{n=0}^{N_B-1} \left[4\hat{C}(2^n) + 96\hat{n} \sum_{i=1}^{\ell} \hat{C}(|\mathcal{I}_i|) + 60 \sum_{m=0}^n \frac{\rho(2^m)}{\rho(2^n)} \hat{C}(2^m) \log(T/\delta) \right].$$

This reasoning mirrors the approach used in the proof of Theorem 2 from Wei and Luo (2021), as detailed in the Appendices D and E. However, unlike the analysis from Wei and Luo (2021), we do not need to establish an upper bound on the regret in our case. This allows us to bypass the examination of MASTER's behavior between restarts. Instead, summing the relevant terms for the regret, we directly obtain the tightest possible regret bound for MASTER. According to Definition 1, the dynamic regret $\text{Reg}_D(T)$ is given as,

$$\text{Reg}_D(T) = \sum_{\tau=1}^T (f_\tau^* - R_\tau) + \sum_{\tau=1}^T (R_\tau - \mathbb{E}[R_\tau])$$

which follows from the linearity of the expectation. Hence, this implies that for the dynamic regret of MASTER,

$$\text{Reg}_D(T) \leq \sum_{n=0}^{N_B-1} \left[4\hat{C}(2^n) + 96\hat{n} \sum_{i=1}^{\ell} \hat{C}(|\mathcal{I}_i|) + 60 \sum_{m=0}^n \frac{\rho(2^m)}{\rho(2^n)} \hat{C}(2^m) \log(T/\delta) \right] + \sum_{\tau=1}^T (R_\tau - \mathbb{E}[R_\tau]). \quad (5)$$

Notice that for $\sum_{\tau=1}^T (R_\tau - \mathbb{E}[R_\tau])$, we can use the Azuma-Hoeffding inequality to provide a high-probability (with probability at least $1 - \delta$) positive upper bound of sub-linear order in the $\tilde{\mathcal{O}}(\cdot)$ notation, that will not affect the order-optimal regret bounds of MASTER. This extra term, however, is not of interest in our derivation, as we only need to provide a lower-bound on the dynamic regret bound of MASTER, by considering a lower bound on the RHS of Equation (5). This means that we can only focus on the terms obtained by the regret bound on the blocks of MASTER.

Thus, let $U^M(\text{Reg}_D(T))$ denote the dynamic regret bound of MASTER, that is,

$$U^M(\text{Reg}_D(T)) = \sum_{n=0}^{N_B-1} \left[4\hat{C}(2^n) + 96\hat{n} \sum_{i=1}^{\ell} \hat{C}(|\mathcal{I}_i|) + 60 \sum_{m=0}^n \frac{\rho(2^m)}{\rho(2^n)} \hat{C}(2^m) \log(T/\delta) \right] + \sum_{\tau=1}^T (R_\tau - \mathbb{E}[R_\tau]).$$

The number of blocks in the horizon N_B , depends on the size of the horizon and the non-stationarity detection. It is evident, that the smallest number of blocks is obtained when no restarts happen due to the non-stationarity detection scheme, and in this case N_B becomes deterministic. Thereupon, we can consider the case where no restarts happen due to the non-stationarity detection as a lower bound on $U^M(\text{Reg}_D(T))$. This lower bound essentially corresponds to the case for which the non-stationarity is weak enough for MASTER to handle only with the guarantees of Lemma 1. However, notice that the following needs to hold,

$$\sum_{n=0}^{N_B-1} 2^n \geq T.$$

Due to this observation, we can equivalently study the case where we have only a single block with $n = \lceil \log_2(T) \rceil$. By selecting $n = \lceil \log_2(T) \rceil$ we obtain lower regret compared to the doubling trick that MASTER does. In the case of the doubling trick, the overall performance will deteriorate due to the excess restarting since the horizon is unknown. Conversely, using the knowledge of the horizon will not lead to unnecessary restarts for a given horizon T , which leads to a smaller overall regret. Thereupon, studying the case where we fix $n = \lceil \log_2(T) \rceil$, rather than using the doubling trick, is another lower bound for $U^M(\text{Reg}_D(T))$ when we assume no restarts due to the non-stationarity tests.

Based on the above, we have the following,

$$U^M(\text{Reg}_D(T)) \geq 4\hat{C}(2^n) + 60\frac{\rho(2^n)}{\rho(2^n)}\hat{C}(2^n)\log(T/\delta)$$

as we can only select a single term from the sum $60 \sum_{m=0}^n \frac{\rho(2^m)}{\rho(2^m)}\hat{C}(2^m)\log(T/\delta)$ to derive a lower bound on $U^M(\text{Reg}_D(T))$, and since $n = \lceil \log_2(T) \rceil$ and $\hat{C}(t)$ is non-decreasing,

$$U^M(\text{Reg}_D(T)) \geq 4\hat{C}(T) + 60\hat{C}(T)\log(T/\delta).$$

According to the definition of $\hat{C}(t)$ and $\hat{\rho}(t)$, we have that,

$$4\hat{C}(T) + 60\hat{C}(T)\log(T/\delta) = 24(\log_2(T) + 1)\log(T/\delta)T\rho(T)(1 + 15\log(T/\delta)).$$

However, based on Assumption 1, $\rho(t) \geq 1/\sqrt{t}$ and since $\delta \in (0, 1)$,

$$24(\log_2(T) + 1)\log(T/\delta)T\rho(T)(1 + 15\log(T/\delta)) > 24(\log_2(T) + 1)\log(T)\sqrt{T}(1 + 15\log(T)).$$

Thus, setting,

$$B_D(T) = 24(\log_2(T) + 1)\log(T)\sqrt{T}(1 + 15\log(T))$$

we finally have that the bound of the dynamic regret of MASTER is lower bounded as,

$$U^M(\text{Reg}_D(T)) > B_D(T).$$

According to Definition 1, since the rewards are bounded in $[0, 1]$, the worst-case linear dynamic regret is exactly equal to T . Moreover, this worst-case regret is a trivial upper bound for the dynamic regret of any algorithm in the NS-RL problem with such rewards, and thus, in order for MASTER's regret bound to not be trivial, we need,

$$U^M(\text{Reg}_D(T)) < T.$$

However, we can show that for $T \leq 409189687210680$,

$$T < B_D(T) \leq U^M(\text{Reg}_D(T))$$

which means that, whenever

$$T \leq 4 \times 10^{14}, \quad T < U^M(\text{Reg}_D(T))$$

which concludes the proof.

B.5 Proof Of Theorem 2

We first restate the theorem.

Theorem 2 *Let the geometric change-point parameter be $\eta = \Theta(T^{-\xi})$, for some $\xi \in (0, 1)$. Set the intervals between random restarts of the RR algorithm to be i.i.d. $\text{Geo}(\eta_R)$, for some $\eta_R \in (0, 1)$ with $\eta_R > \eta$ (order-wise). If the regret of base algorithm is $\mathcal{O}(\text{poly}(\log(T)))$, and $\eta_R = \sqrt{\eta/\text{poly}(\log(T))}$, the RR algorithm is order optimal for PS-MABs.*

Proof:

To prove the theorem, we first define the notion of (stationary) regret. So far, we have given the definition of dynamic regret as shown in Definition 1. Under a stationary environment, we have that the reward function is stationary, i.e. $f_1 = f_2 = \dots = f_T = f$. Thus, in this case the regret can be defined as,

$$\text{Reg} = \mathbb{E} \left[\sum_{t=1}^T (f^* - f(\pi_t)) \right]$$

where $f^* = \max_{\pi \in \Pi} f(\pi)$. In the case of MABs, the set of policies Π is equivalent to the set of arms $\mathcal{A} = \{1, \dots, A\}$. Moreover, it is evident that the expectation $f(\pi)$ corresponds to the mean of an arm μ_α , where $\alpha \in \Pi = \mathcal{A}$, i.e.,

$$f(\pi) \equiv \mu_a = \mathbb{E}[R_t | \alpha].$$

Thus, the regret of a stationary bandit algorithm \mathcal{B} , $R_{\mathcal{B}}(T)$, essentially corresponds to,

$$R_{\mathcal{B}} = \mathbb{E} \left[\sum_{t=1}^T (\mu_{\alpha^*} - \mu_{A_t}) \right]$$

where $\alpha^* = \arg \max_a \mu_a$ and A_t is selected according to the bandit algorithm \mathcal{B} . Thus, we can proceed with the analysis of the regret for RR.

Let $(r^{(k)})_{k=1}^\infty$ denote an instance of random restart points drawn such that $(r^{(k)} - r^{(k-1)}) \stackrel{\text{i.i.d.}}{\sim} \text{Geo}(\eta_R)$, with $r^{(0)} := 1$. The total number of the random restarts within the horizon T is given by $N_R := \sup\{k \in \mathbb{N} : r^{(k)} \leq T\}$. Consider a fixed case of the random restart points, where the restart points are set to some arbitrary fixed values in the horizon, expressed by the event $E_{\mathcal{T}} := \{r^{(1)} = t_1, r^{(2)} = t_2, r^{(3)} = t_3, \dots\}$, where $\mathcal{T} = \{t_k\}_{k=1}^\infty$ is a fixed, increasing, arbitrary sequence of positive integers, larger than one. Let, $\Delta_{t,a} = \max_{a^*} \mu_{t,a^*} - \mu_{t,a}$, and since $R_t \in [0, 1]$ based on Definition 1, $\Delta_{t,a} \leq 1$ for all $a \in \Pi$ and for all t . Thus, the dynamic regret of the RR algorithm can be expressed as

$$\text{Reg}_D(T) = \mathbb{E} \left[\sum_{t=1}^T \Delta_{t,A_t} \right] = \mathbb{E} \left[\sum_{k=1}^{N_R} \sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} + \sum_{t=r^{(N_R)}}^T \Delta_{t,A_t} \right].$$

We focus on the regret accumulated between any two consecutive random restart points, denoted as $r^{(k-1)} = t_{k-1}$ and $r^{(k)} = t_k$. For the purpose of our analysis, we treat these specific instances of restart points as fixed (i.e., we condition on their occurrence), allowing us to ignore the randomness introduced by the random scheduling. Let $C_k := \{\text{no change in } [t_{k-1}, t_k - 1]\}$. Then, we have the following inequality for the regret between consecutive restart points, conditioned on $E_{\mathcal{T}}$,

$$\begin{aligned} \mathbb{E} \left[\sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} \middle| E_{\mathcal{T}} \right] &= \Pr\{C_k\} \mathbb{E} \left[\sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} \middle| E_{\mathcal{T}}, C_k \right] + \Pr\{\bar{C}_k\} \mathbb{E} \left[\sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} \middle| E_{\mathcal{T}}, \bar{C}_k \right] \\ &\stackrel{(a)}{\leq} \Pr\{C_k\} R_{\mathcal{B}}(t_k - t_{k-1}) + \Pr\{\bar{C}_k\}(t_k - t_{k-1}) \\ &\stackrel{(b)}{\leq} R_{\mathcal{B}}(T) + \Pr\{\bar{C}_k\}(t_k - t_{k-1}). \end{aligned}$$

In step (a), we use the crude linear bound $t_k - t_{k-1}$ to bound $\mathbb{E} \left[\sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} \middle| E_{\mathcal{T}}, \bar{C}_k \right]$ and $R_{\mathcal{B}}(t_k - t_{k-1})$ to bound $\mathbb{E} \left[\sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} \middle| E_{\mathcal{T}}, C_k \right]$, since the piecewise stationary bandit becomes stationary in $[t_{k-1}, t_k - 1]$ given C_k . In step (b), we apply the fact that $\Pr\{C_k\} \leq 1$ and $R_{\mathcal{B}}(T)$ is increasing with T . Then, we derive an upper bound on $\Pr\{\bar{C}_k\}$ as follows:

$$\begin{aligned} \Pr\{\bar{C}_k\} &= 1 - \Pr\{C_k\} \\ &= 1 - (1 - \eta)^{t_k - t_{k-1}} \\ &\stackrel{(a)}{\leq} 1 - [1 - \eta(t_k - t_{k-1})] \\ &= \eta(t_k - t_{k-1}) \end{aligned}$$

where step (a) results from the inequality $(1 - x)^n \geq 1 - nx$ for $x \in (0, 1)$ and $n \in \mathbb{N}$. Hence, we have:

$$\mathbb{E} \left[\sum_{t=r^{(k-1)}}^{r^{(k)}-1} \Delta_{t,A_t} \middle| E_{\mathcal{T}} \right] \leq R_{\mathcal{B}}(T) + \eta(t_k - t_{k-1})^2.$$

Following the previous reasoning, we equivalently have that,

$$\mathbb{E} \left[\sum_{t=t_{N_R}}^T \Delta_{t,A_t} \middle| E_{\mathcal{T}} \right] \leq R_{\mathcal{B}}(T) + \eta(t_{N_R+1} - t_{N_R})^2.$$

Thereupon, since the previous bounds hold for any arbitrary set of random restarting points, we can substitute the random variables and upper bound the initial regret as follows:

$$\begin{aligned}
 \text{Reg}_D(T) &\leq \mathbb{E} \left[\sum_{k=1}^{N_R} \left[R_B(T) + \eta \left(r^{(k)} - r^{(k-1)} \right)^2 \right] + R_B(T) + \eta \left(r^{(N_R+1)} - r^{(N_R)} \right)^2 \right] \\
 &= \mathbb{E}[N_R + 1]R_B(T) + \eta \mathbb{E} \left[\sum_{k=1}^{N_R+1} \left(r^{(k)} - r^{(k-1)} \right)^2 \right] \\
 &= \mathbb{E}[N_R + 1]R_B(T) + \eta \mathbb{E}[N_R + 1] \mathbb{E} \left[\left(r^{(1)} - r^{(0)} \right)^2 \right] \\
 &= (\eta_R T + 1)R_B(T) + \eta(\eta_R T + 1) \left(\frac{2 - \eta_R}{\eta_R^2} \right)
 \end{aligned}$$

where the above is derived using the geometric i.i.d. assumption and Wald's Identity, as N_R is a stopping time. Hence, since $R_B(T) = \mathcal{O}(\text{poly}(\log(T)))$, the following holds,

$$\begin{aligned}
 \text{Reg}_D(T) &\leq \frac{2\eta T}{\eta_R} + \eta_R T R_B(T) + \eta \left(\frac{2 - \eta_R}{\eta_R^2} \right) + R_B(T) \\
 \text{Reg}_D(T) &= \mathcal{O} \left(\frac{2\eta T}{\eta_R} + \eta_R T \text{poly}(\log(T)) + \eta \left(\frac{2 - \eta_R}{\eta_R^2} \right) + \text{poly}(\log(T)) \right).
 \end{aligned}$$

Ignoring the constants that are independent of the horizon, since $\eta_R > \eta$ (order-wise) and $\eta_R \in (0, 1)$, it is evident that the leading terms in the regret, correspond to $\frac{\eta T}{\eta_R}$ and $\eta_R T \text{poly}(\log(T))$, and in order to ensure that we have the appropriate sub-linear order,

$$\eta T / \eta_R = \eta_R T \text{poly}(\log(T)) \Rightarrow \eta_R = \sqrt{\eta / \text{poly}(\log(T))}.$$

Thus, we have that,

$$\begin{aligned}
 \text{Reg}_D(T) &= \mathcal{O} \left(\frac{2\eta T \sqrt{\text{poly}(\log(T))}}{\sqrt{\eta}} + \frac{\sqrt{\eta}}{\sqrt{\text{poly}(\log(T))}} T \text{poly}(\log(T)) \right. \\
 &\quad \left. + \eta \left(\frac{2\text{poly}(\log(T)) - \sqrt{\eta \text{poly}(\log(T))}}{\eta} \right) + \text{poly}(\log(T)) \right) \\
 &= \mathcal{O} \left(3\sqrt{\eta} T \sqrt{\text{poly}(\log(T))} - \sqrt{\eta \text{poly}(\log(T))} + 3\text{poly}(\log(T)) \right).
 \end{aligned}$$

Finally, since $\eta = \Theta(T^{-\xi})$, $\xi \in (0, 1)$, we have that,

$$\text{Reg}_D(T) = \mathcal{O} \left(3T^{1-\xi/2} \sqrt{\text{poly}(\log(T))} - T^{-\xi/2} \sqrt{\text{poly}(\log(T))} + 3\text{poly}(\log(T)) \right)$$

which implies,

$$\text{Reg}_D(T) = \mathcal{O} \left(3T^{1-\xi/2} \sqrt{\text{poly}(\log(T))} + 3\text{poly}(\log(T)) \right).$$

Hence, it is evident that, $\text{Reg}_D(T) = \tilde{\mathcal{O}}(T^{1-\xi/2})$.

To show that the RR algorithm is order-optimal, we need to show that the optimal minimax bound is $\tilde{\Omega}(T^{1-\xi/2})$ as well.

In the context of NS-MABs, the non-stationarity measure, according to Wei and Luo (2021), is defined as $\Delta(t) = \max_\alpha |\mu_{t,\alpha} - \mu_{t+1,\alpha}|$. In the PS-MAB setting, each change-point introduces a non-negative non-stationarity $\Delta(t)$. Hence, the number of changes $L = 1 + \sum_{\tau=1}^{T-1} \mathbf{1}[\Delta(\tau) \neq 0]$ and the cumulative non-stationarity $\Delta = \sum_{\tau=1}^{T-1} \Delta(\tau)$ have the same order in T . Therefore, the minimax optimal regret bound in $\tilde{\mathcal{O}}(\cdot)$ notation is $\min \left\{ \Delta^{1/3} T^{2/3} + \sqrt{T}, \sqrt{LT} \right\} = \sqrt{LT}$. To demonstrate that RR is order-optimal, we analyze the behavior of \sqrt{LT} .

Due to the geometric change-point assumption, it is evident that the number of changes L is distributed as a binomial random variable, i.e., $L \sim \text{B}(T, \eta)$. Now, first notice that by Jensen's inequality,

$$\mathbb{E}[\sqrt{L}] \leq \sqrt{\mathbb{E}[L]}$$

and by the definition of the binomial random variable, $\mathbb{E}[L] = \eta T$, which implies,

$$\mathbb{E}[\sqrt{L}] \leq \sqrt{\eta T}.$$

On the other hand, since $\sqrt{x} \geq 1 + \frac{x-1}{2} - \frac{(x-1)^2}{2}$ for any $x \geq 0$, by plugging $\frac{L}{\mathbb{E}[L]}$ into x and then taking expectation, we have,

$$\frac{\mathbb{E}[\sqrt{L}]}{\sqrt{\mathbb{E}[L]}} = \mathbb{E}\left[\sqrt{\frac{L}{\mathbb{E}[L]}}\right] \geq \mathbb{E}\left[1 + \frac{L/\mathbb{E}[L] - 1}{2} - \frac{(L/\mathbb{E}[L] - 1)^2}{2}\right] = 1 - \frac{\text{Var}(L)}{2(\mathbb{E}[L])^2}.$$

Since $\mathbb{E}[L] = \eta T$ and $\text{Var}(L) = \eta(1 - \eta)T$, we have,

$$\mathbb{E}[\sqrt{L}] \geq \sqrt{\eta T} \left(1 - \frac{1 - \eta}{2\eta T}\right).$$

Because $\eta = \Theta(T^{-\xi})$ and $\xi \in (0, 1)$, we can show that

$$\mathbb{E}[\sqrt{LT}] = \tilde{\Omega}\left(T^{(2-\xi)/2} \left(1 - \frac{1 - T^{-\xi}}{2T^{(1-\xi)}}\right)\right) = \tilde{\Omega}\left(T^{1-\xi/2}\right).$$

Hence, the RR algorithm is order-optimal. Note that RR can be made order-optimal for a case of deterministic change-points, as long as we focus on an instance of the geometric change-point model.

C Experimental Details

C.1 Technical Details

The experiments were conducted using the C++ programming language. Moreover, all the experiments were conducted on a server with an Intel Xeon Gold 6148 CPU and allocating 200 GB of memory.

C.2 Environment Details

Geometric Change-Points. The change-points are generated using a geometric distribution. More specifically, for a given horizon T , the probability of the geometric distribution is set exactly equal to $\eta = T^{-\xi}$ for some $\xi \in (0, 1)$.

Deterministic Change-Points. For a given horizon T , the number of change-points is set equal to N_C , with $N_C < T$. Then, the change-points are placed every $\text{round}(T/N_C)$ time-steps in the horizon, where $\text{round}(\cdot)$ corresponds to the rounding operation. This ensures that the change-points are roughly evenly placed. In order to ensure that RR can be applied in this deterministic scenario, we fix a geometric parameter η and we set the number of change-points equal to $N_C = \lceil \eta T \rceil$ in all of the runs.

Uniform Problem. For this problem, the means of all arms are initialized in $[0, 1]$. Then in each change-point, the number of arms that are going to change is chosen uniformly, but is always greater than one. For each arm, the magnitude of the change is uniformly selected in $[0.1, 0.4]$ with probability 0.5 to be a positive or negative change. If a change results in a mean exceeding 1, the change is subtracted instead. Similarly, if it would result in a mean below 0, the change is added.

Worst-Case Problem. In this problem, the means of all arms are initialized at 0.3, plus a small positive offset uniformly selected from $[0.0005, 0.005]$, ensuring that the means remain close to each other. At each change point, the arm with the lowest mean changes to match the highest mean plus a small gap uniformly selected from $[0.005, 0.05]$, ensuring that the change is small, but enough to affect the environment. If the mean of any arm exceeds 0.99 at a change point, then the process is reset and follows the same initialization, but ensuring that the arm with the lowest mean, becomes the arm with the highest mean.

C.3 Algorithm Details

The algorithm of GLRklUCB is employed with global restarts and the Bernoulli GLR change-point detector, as specified in Besson et al. (2022). The Bernoulli GLR change-point detector is also used for the QCD+ algorithm. When employing the Bernoulli GLR change-point detector, as it was similarly suggested in Besson et al. (2022), its threshold is set to be equal to $\beta(n, \delta) = \log(4n\sqrt{n}/\delta)$.

C.4 Additional Results

The experiments were conducted for horizons ranging from $T = 1000$ to $T = 100000$, with increments of approximately 5000, averaged over 4000 independent trials. For geometric change-points, we use $\xi \in \{0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$, and for deterministic change-points, $N_C \in \{[T^{0.7}], [T^{0.6}], [T^{0.5}], [T^{0.4}], [T^{0.3}], [T^{0.2}]\}$.

Due to size limitations and the minimal differences observed at every 5000-step increment, we present the dynamic regret results using horizon increments of 10000, with the addition of $T = 2000$ and $T = 5000$. Omitted results and the presented results are also available in high resolution at the attached link related to our code.

Moreover, since the experiments were conducted using the C++ programming language, which is very fast in computations, there was no significant difference in the average running time until the horizon became adequately large among the methods, with the exception of QCD+UCB, which was always significantly faster. Given the significant computational demands of MASTER, the lack of a noticeable difference for these horizons is due to the complexity of the klUCB algorithm, which is computationally demanding. Therefore, we only report the average running time for horizons of $T = 60000$ and beyond, while the dynamic regret and non-stationarity detection results are presented for the full range mentioned earlier.

Table 3: Average Time Per Run (sec) For Different Change-Points, $T=60000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	1.39 ± 0.00	1.39 ± 0.00	1.39 ± 0.00	1.39 ± 0.00	1.40 ± 0.06	1.40 ± 0.04
MASTER	Worst	1.39 ± 0.01	1.39 ± 0.01	1.39 ± 0.00	1.39 ± 0.04	1.39 ± 0.04	1.39 ± 0.04
RR_p.05	Unif	0.68 ± 0.01	0.68 ± 0.01	0.67 ± 0.01	0.67 ± 0.02	0.67 ± 0.03	0.67 ± 0.04
RR_p.05	Worst	0.58 ± 0.01	0.57 ± 0.02	0.57 ± 0.02	0.58 ± 0.03	0.64 ± 0.05	0.74 ± 0.02
GLRklUCB	Unif	0.83 ± 0.01	0.82 ± 0.01	0.86 ± 0.01	0.95 ± 0.03	1.17 ± 0.09	1.75 ± 0.41
GLRklUCB	Worst	0.79 ± 0.01	0.88 ± 0.02	0.97 ± 0.05	1.18 ± 0.10	1.66 ± 0.23	2.68 ± 0.78
QCD+UCB	Unif	0.05 ± 0.01	0.05 ± 0.00	0.06 ± 0.01	0.11 ± 0.02	0.30 ± 0.10	0.86 ± 0.42
QCD+UCB	Worst	0.04 ± 0.00	0.06 ± 0.01	0.09 ± 0.01	0.17 ± 0.04	0.34 ± 0.10	0.70 ± 0.30
QCD+klUCB	Unif	1.01 ± 0.00	1.01 ± 0.01	1.03 ± 0.01	1.10 ± 0.03	1.32 ± 0.10	1.93 ± 0.44
QCD+klUCB	Worst	1.00 ± 0.01	1.06 ± 0.04	1.17 ± 0.13	1.42 ± 0.32	1.79 ± 0.24	2.83 ± 0.80

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	2212	737	245	82	28	10
MASTER	Unif	1.40 ± 0.00	1.41 ± 0.07	1.40 ± 0.01	1.40 ± 0.01	1.41 ± 0.19	1.43 ± 0.16
MASTER	Worst	1.40 ± 0.01	1.40 ± 0.05	1.41 ± 0.11	1.41 ± 0.01	1.40 ± 0.02	1.42 ± 0.20
RR_p.05	Unif	0.68 ± 0.00	0.68 ± 0.05	0.67 ± 0.00	0.65 ± 0.01	0.72 ± 0.12	0.70 ± 0.11
RR_p.05	Worst	0.58 ± 0.01	0.58 ± 0.01	0.59 ± 0.05	0.62 ± 0.01	0.60 ± 0.00	0.77 ± 0.14
GLRklUCB	Unif	0.87 ± 0.01	0.84 ± 0.05	0.84 ± 0.01	0.89 ± 0.02	1.01 ± 0.15	1.32 ± 0.18
GLRklUCB	Worst	0.79 ± 0.01	0.89 ± 0.03	0.98 ± 0.08	1.15 ± 0.06	1.43 ± 0.06	2.07 ± 0.29
QCD+UCB	Unif	0.07 ± 0.01	0.05 ± 0.02	0.05 ± 0.00	0.07 ± 0.00	0.15 ± 0.06	0.42 ± 0.10
QCD+UCB	Worst	0.05 ± 0.00	0.07 ± 0.03	0.10 ± 0.03	0.17 ± 0.00	0.21 ± 0.00	0.52 ± 0.11
QCD+klUCB	Unif	1.02 ± 0.01	1.02 ± 0.06	1.02 ± 0.00	1.03 ± 0.01	1.14 ± 0.16	1.43 ± 0.16
QCD+klUCB	Worst	1.00 ± 0.00	1.06 ± 0.02	1.42 ± 0.28	1.24 ± 0.02	1.58 ± 0.08	2.22 ± 0.34

Table 4: Average Time Per Run (sec) For Different Change-Points, $T=70000$, 4000 Runs.

Algorithm	PB	(a) Random Change-Points (Columns: ξ)					
		0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	1.76 ± 0.01	1.76 ± 0.00	1.76 ± 0.01	1.76 ± 0.01	1.76 ± 0.07	1.77 ± 0.13
MASTER	Worst	1.76 ± 0.00	1.76 ± 0.02	1.76 ± 0.02	1.76 ± 0.01	1.76 ± 0.01	1.76 ± 0.02
RR-p.05	Unif	0.79 ± 0.01	0.79 ± 0.01	0.79 ± 0.01	0.78 ± 0.02	0.79 ± 0.05	0.79 ± 0.05
RR-p.05	Worst	0.67 ± 0.01	0.66 ± 0.01	0.66 ± 0.03	0.68 ± 0.04	0.74 ± 0.06	0.86 ± 0.03
GLRklUCB	Unif	0.97 ± 0.01	0.96 ± 0.01	1.00 ± 0.01	1.12 ± 0.03	1.40 ± 0.11	2.19 ± 0.52
GLRklUCB	Worst	0.92 ± 0.01	1.03 ± 0.03	1.15 ± 0.07	1.40 ± 0.17	2.03 ± 0.30	3.44 ± 1.08
QCD+UCB	Unif	0.06 ± 0.01	0.06 ± 0.00	0.07 ± 0.01	0.14 ± 0.02	0.39 ± 0.12	1.14 ± 0.53
QCD+UCB	Worst	0.05 ± 0.00	0.07 ± 0.01	0.10 ± 0.01	0.20 ± 0.04	0.42 ± 0.13	0.91 ± 0.40
QCD+klUCB	Unif	1.18 ± 0.00	1.18 ± 0.01	1.20 ± 0.01	1.30 ± 0.03	1.58 ± 0.12	2.41 ± 0.57
QCD+klUCB	Worst	1.16 ± 0.01	1.25 ± 0.05	1.39 ± 0.17	1.73 ± 0.46	2.19 ± 0.30	3.66 ± 1.13

Algorithm	PB	(b) Deterministic Change-Points (Columns: N_C)					
		2464	808	265	87	29	10
MASTER	Unif	1.76 ± 0.01	1.83 ± 0.02	1.80 ± 0.01	1.77 ± 0.00	1.79 ± 0.11	1.85 ± 0.20
MASTER	Worst	1.77 ± 0.01	1.79 ± 0.06	1.81 ± 0.12	1.81 ± 0.11	1.77 ± 0.08	1.82 ± 0.09
RR-p.05	Unif	0.79 ± 0.00	0.79 ± 0.01	0.79 ± 0.00	0.80 ± 0.00	0.82 ± 0.05	0.75 ± 0.19
RR-p.05	Worst	0.67 ± 0.01	0.66 ± 0.03	0.68 ± 0.01	0.69 ± 0.05	0.71 ± 0.03	0.86 ± 0.03
GLRklUCB	Unif	1.02 ± 0.01	0.96 ± 0.01	0.97 ± 0.01	1.05 ± 0.05	1.20 ± 0.07	1.58 ± 0.13
GLRklUCB	Worst	0.93 ± 0.01	1.09 ± 0.03	1.15 ± 0.07	1.35 ± 0.11	1.95 ± 0.11	2.88 ± 0.36
QCD+UCB	Unif	0.11 ± 0.00	0.06 ± 0.00	0.06 ± 0.00	0.09 ± 0.00	0.18 ± 0.03	0.52 ± 0.13
QCD+UCB	Worst	0.07 ± 0.00	0.07 ± 0.01	0.11 ± 0.02	0.19 ± 0.04	0.34 ± 0.04	0.48 ± 0.01
QCD+klUCB	Unif	1.19 ± 0.01	1.18 ± 0.02	1.18 ± 0.00	1.24 ± 0.05	1.35 ± 0.06	1.73 ± 0.21
QCD+klUCB	Worst	1.17 ± 0.01	1.35 ± 0.04	1.46 ± 0.10	1.52 ± 0.12	2.05 ± 0.09	3.02 ± 0.38

 Table 5: Average Time Per Run (sec) For Different Change-Points, $T=80000$, 4000 Runs.

Algorithm	PB	(a) Random Change-Points (Columns: ξ)					
		0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	2.06 ± 0.01	2.06 ± 0.00	2.06 ± 0.00	2.07 ± 0.01	2.07 ± 0.11	2.07 ± 0.11
MASTER	Worst	2.07 ± 0.02	2.06 ± 0.02	2.06 ± 0.03	2.06 ± 0.06	2.06 ± 0.06	2.07 ± 0.10
RR-p.05	Unif	0.90 ± 0.01	0.90 ± 0.01	0.90 ± 0.02	0.90 ± 0.03	0.90 ± 0.06	0.90 ± 0.06
RR-p.05	Worst	0.77 ± 0.01	0.76 ± 0.01	0.76 ± 0.02	0.77 ± 0.04	0.83 ± 0.06	0.98 ± 0.04
GLRklUCB	Unif	1.11 ± 0.01	1.10 ± 0.01	1.15 ± 0.02	1.29 ± 0.03	1.69 ± 0.16	2.68 ± 0.68
GLRklUCB	Worst	1.06 ± 0.01	1.18 ± 0.03	1.33 ± 0.09	1.66 ± 0.33	2.43 ± 0.38	4.24 ± 1.34
QCD+UCB	Unif	0.07 ± 0.01	0.06 ± 0.00	0.09 ± 0.01	0.18 ± 0.03	0.49 ± 0.14	1.46 ± 0.69
QCD+UCB	Worst	0.06 ± 0.01	0.08 ± 0.01	0.12 ± 0.02	0.25 ± 0.05	0.50 ± 0.15	1.14 ± 0.50
QCD+klUCB	Unif	1.34 ± 0.01	1.35 ± 0.01	1.38 ± 0.01	1.49 ± 0.04	1.85 ± 0.15	2.92 ± 0.72
QCD+klUCB	Worst	1.33 ± 0.02	1.43 ± 0.05	1.62 ± 0.19	2.04 ± 0.58	2.63 ± 0.38	4.50 ± 1.39

Algorithm	PB	(b) Deterministic Change-Points (Columns: N_C)					
		2705	875	283	92	30	10
MASTER	Unif	2.07 ± 0.01	2.13 ± 0.10	2.13 ± 0.13	2.11 ± 0.04	2.07 ± 0.01	2.12 ± 0.06
MASTER	Worst	2.10 ± 0.06	2.10 ± 0.05	2.09 ± 0.04	2.07 ± 0.02	2.13 ± 0.03	2.09 ± 0.04
RR-p.05	Unif	0.90 ± 0.02	0.90 ± 0.03	0.90 ± 0.04	0.93 ± 0.02	0.89 ± 0.01	0.89 ± 0.02
RR-p.05	Worst	0.78 ± 0.04	0.75 ± 0.04	0.73 ± 0.03	0.74 ± 0.01	0.83 ± 0.01	0.97 ± 0.03
GLRklUCB	Unif	1.16 ± 0.01	1.11 ± 0.10	1.14 ± 0.08	1.22 ± 0.02	1.41 ± 0.01	1.73 ± 0.06
GLRklUCB	Worst	1.06 ± 0.06	1.22 ± 0.05	1.35 ± 0.19	1.66 ± 0.29	2.35 ± 0.18	3.22 ± 0.37
QCD+UCB	Unif	0.10 ± 0.00	0.06 ± 0.00	0.07 ± 0.01	0.10 ± 0.01	0.23 ± 0.02	0.48 ± 0.01
QCD+UCB	Worst	0.06 ± 0.01	0.09 ± 0.02	0.10 ± 0.01	0.19 ± 0.01	0.45 ± 0.01	0.61 ± 0.03
QCD+klUCB	Unif	1.36 ± 0.01	1.36 ± 0.03	1.37 ± 0.07	1.43 ± 0.03	1.58 ± 0.01	1.91 ± 0.02
QCD+klUCB	Worst	1.33 ± 0.05	1.47 ± 0.05	1.70 ± 0.15	1.89 ± 0.31	2.48 ± 0.18	3.38 ± 0.39

 Table 6: Average Time Per Run (sec) For Different Change-Points, $T=90000$, 4000 Runs.

Algorithm	PB	(a) Random Change-Points (Columns: ξ)					
		0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	2.58 ± 0.02	2.58 ± 0.02	2.58 ± 0.00	2.58 ± 0.05	2.58 ± 0.06	2.60 ± 0.19
MASTER	Worst	2.58 ± 0.01	2.58 ± 0.00	2.58 ± 0.03	2.58 ± 0.04	2.59 ± 0.11	2.59 ± 0.13
RR-p.05	Unif	1.02 ± 0.01	1.01 ± 0.01	1.01 ± 0.02	1.01 ± 0.05	1.01 ± 0.08	1.02 ± 0.10
RR-p.05	Worst	0.87 ± 0.01	0.85 ± 0.02	0.85 ± 0.03	0.87 ± 0.06	0.93 ± 0.08	1.10 ± 0.02
GLRklUCB	Unif	1.24 ± 0.01	1.23 ± 0.01	1.30 ± 0.02	1.47 ± 0.04	1.91 ± 0.17	3.18 ± 0.80
GLRklUCB	Worst	1.20 ± 0.01	1.34 ± 0.03	1.51 ± 0.10	1.90 ± 0.35	2.84 ± 0.46	5.17 ± 1.71
QCD+UCB	Unif	0.08 ± 0.01	0.07 ± 0.01	0.10 ± 0.01	0.21 ± 0.04	0.59 ± 0.17	1.81 ± 0.82
QCD+UCB	Worst	0.07 ± 0.00	0.09 ± 0.01	0.14 ± 0.02	0.29 ± 0.06	0.59 ± 0.18	1.40 ± 0.64
QCD+klUCB	Unif	1.51 ± 0.01	1.52 ± 0.01	1.56 ± 0.01	1.70 ± 0.07	2.14 ± 0.19	3.46 ± 0.84
QCD+klUCB	Worst	1.50 ± 0.01	1.61 ± 0.06	1.84 ± 0.24	2.37 ± 0.73	3.09 ± 0.45	5.51 ± 1.80

Algorithm	PB	(b) Deterministic Change-Points (Columns: N_C)					
		2938	939	300	96	31	10
MASTER	Unif	2.64 ± 0.07	2.65 ± 0.08	2.64 ± 0.02	2.60 ± 0.03	2.63 ± 0.09	2.64 ± 0.20
MASTER	Worst	2.63 ± 0.05	2.68 ± 0.09	2.65 ± 0.23	2.64 ± 0.02	2.64 ± 0.16	2.67 ± 0.25
RR-p.05	Unif	1.03 ± 0.06	1.03 ± 0.04	1.01 ± 0.01	1.06 ± 0.00	1.06 ± 0.08	1.00 ± 0.16
RR-p.05	Worst	0.87 ± 0.01	0.88 ± 0.05	0.89 ± 0.14	0.86 ± 0.01	0.89 ± 0.06	1.12 ± 0.05
GLRklUCB	Unif	1.31 ± 0.05	1.24 ± 0.05	1.27 ± 0.01	1.36 ± 0.01	1.63 ± 0.08	2.38 ± 0.17
GLRklUCB	Worst	1.20 ± 0.01	1.35 ± 0.07	1.60 ± 0.22	1.76 ± 0.14	2.44 ± 0.27	4.36 ± 1.01
QCD+UCB	Unif	0.13 ± 0.02	0.07 ± 0.02	0.08 ± 0.00	0.12 ± 0.00	0.30 ± 0.04	1.03 ± 0.21
QCD+UCB	Worst	0.07 ± 0.00	0.11 ± 0.03	0.13 ± 0.06	0.20 ± 0.01	0.33 ± 0.01	1.22 ± 0.13
QCD+klUCB	Unif	1.55 ± 0.07	1.53 ± 0.05	1.53 ± 0.01	1.59 ± 0.01	1.82 ± 0.14	2.58 ± 0.14
QCD+klUCB	Worst	1.50 ± 0.00	1.61 ± 0.06	2.03 ± 0.28	2.02 ± 0.18	2.78 ± 0.28	4.58 ± 0.67

Table 7: Average Time Per Run (sec) For Different Change-Points, $T=100000$, 4000 Runs.

		(a) Random Change-Points (Columns: ξ)					
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	3.33 ± 0.03	3.34 ± 0.02	3.34 ± 0.04	3.33 ± 0.03	3.36 ± 0.04	3.35 ± 0.20
MASTER	Worst	3.32 ± 0.03	3.32 ± 0.01	3.33 ± 0.02	3.33 ± 0.05	3.35 ± 0.08	3.36 ± 0.15
RR.p.05	Unif	1.13 ± 0.02	1.12 ± 0.01	1.12 ± 0.02	1.12 ± 0.03	1.12 ± 0.05	1.12 ± 0.07
RR.p.05	Worst	0.96 ± 0.01	0.94 ± 0.02	0.94 ± 0.03	0.96 ± 0.05	1.03 ± 0.09	1.23 ± 0.05
GLRklUCB	Unif	1.38 ± 0.01	1.37 ± 0.01	1.45 ± 0.02	1.65 ± 0.05	2.18 ± 0.20	3.76 ± 1.02
GLRklUCB	Worst	1.33 ± 0.03	1.49 ± 0.04	1.68 ± 0.11	2.14 ± 0.37	3.27 ± 0.53	6.08 ± 2.06
QCD+UCB	Unif	0.09 ± 0.01	0.08 ± 0.01	0.11 ± 0.01	0.24 ± 0.04	0.72 ± 0.20	2.24 ± 1.04
QCD+UCB	Worst	0.08 ± 0.01	0.10 ± 0.01	0.16 ± 0.02	0.33 ± 0.07	0.68 ± 0.19	1.65 ± 0.71
QCD+kIUCB	Unif	1.68 ± 0.01	1.68 ± 0.01	1.73 ± 0.01	1.90 ± 0.04	2.44 ± 0.22	4.12 ± 1.09
QCD+kIUCB	Worst	1.67 ± 0.01	1.80 ± 0.06	2.06 ± 0.28	2.71 ± 0.84	3.60 ± 0.55	6.42 ± 2.13

		(b) Deterministic Change-Points (Columns: N_C)					
Algorithm	PB	3163	1000	317	101	32	10
MASTER	Unif	3.35 ± 0.42	3.32 ± 0.04	3.35 ± 0.22	3.32 ± 0.01	3.32 ± 0.15	3.32 ± 0.02
MASTER	Worst	3.38 ± 0.09	3.38 ± 0.10	3.40 ± 0.08	3.38 ± 0.06	3.37 ± 0.03	3.35 ± 0.07
RR.p.05	Unif	1.15 ± 0.26	1.14 ± 0.04	1.16 ± 0.12	1.12 ± 0.01	1.10 ± 0.04	1.08 ± 0.11
RR.p.05	Worst	0.97 ± 0.05	0.95 ± 0.05	0.94 ± 0.02	1.00 ± 0.01	1.04 ± 0.01	1.24 ± 0.01
GLRklUCB	Unif	1.46 ± 0.28	1.38 ± 0.03	1.43 ± 0.17	1.52 ± 0.01	1.83 ± 0.06	2.67 ± 0.20
GLRklUCB	Worst	1.33 ± 0.06	1.48 ± 0.07	1.66 ± 0.06	2.04 ± 0.11	2.97 ± 0.17	5.06 ± 0.94
QCD+UCB	Unif	0.12 ± 0.08	0.07 ± 0.01	0.13 ± 0.05	0.13 ± 0.00	0.38 ± 0.04	1.09 ± 0.14
QCD+UCB	Worst	0.09 ± 0.02	0.11 ± 0.03	0.16 ± 0.02	0.41 ± 0.01	0.73 ± 0.00	1.05 ± 0.03
QCD+kIUCB	Unif	1.71 ± 0.29	1.69 ± 0.03	1.73 ± 0.17	1.78 ± 0.01	2.02 ± 0.07	2.88 ± 0.17
QCD+kIUCB	Worst	1.68 ± 0.06	1.77 ± 0.07	1.95 ± 0.07	2.35 ± 0.16	3.24 ± 0.22	5.09 ± 0.77

 Table 8: Mean Number Of Declared Changes For Different Change-Points, $T = 1000$, 4000 Runs.

		(a) Random Change-Points (Columns: ξ)					
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	0.54	1.09	1.39	1.39	1.19	0.89
GLRklUCB	Worst	2.24	1.80	1.12	0.24	0.02	0.00
QCD+UCB	Unif	0.45	1.02	1.40	1.47	1.26	0.98
QCD+UCB	Worst	2.25	1.93	1.18	0.37	0.03	0.01
QCD+kIUCB	Unif	0.70	1.20	1.42	1.36	1.11	0.82
QCD+kIUCB	Worst	2.59	1.86	1.13	0.23	0.01	0.00

		(b) Deterministic Change-Points (Columns: N_C)					
Algorithm	PB	126	64	32	16	8	4
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	0.25	0.57	0.52	0.78	1.07	1.15
GLRklUCB	Worst	2.56	2.06	1.00	0.51	0.00	0.00
QCD+UCB	Unif	0.00	1.00	0.00	0.00	1.00	1.00
QCD+UCB	Worst	3.00	2.00	1.00	0.00	0.00	0.00
QCD+kIUCB	Unif	0.41	0.55	0.94	0.72	1.05	1.16
QCD+kIUCB	Worst	2.88	2.04	1.01	0.26	0.00	0.00

 Table 9: Mean Number Of Declared Changes For Different Change-Points, $T = 2000$, 4000 Runs.

		(a) Random Change-Points (Columns: ξ)					
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	1.44	2.65	3.17	2.95	2.31	1.60
GLRklUCB	Worst	4.71	3.59	2.07	0.90	0.14	0.01
QCD+UCB	Unif	1.16	2.46	3.14	3.04	2.44	1.70
QCD+UCB	Worst	4.59	3.67	2.21	1.01	0.25	0.03
QCD+kIUCB	Unif	1.71	2.79	3.05	2.73	2.11	1.42
QCD+kIUCB	Worst	5.20	3.36	1.84	0.88	0.15	0.01

		(b) Deterministic Change-Points (Columns: N_C)					
Algorithm	PB	205	96	45	21	10	5
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	1.03	0.96	2.65	3.24	2.52	1.98
GLRklUCB	Worst	5.13	3.15	2.30	1.01	0.14	0.00
QCD+UCB	Unif	1.00	1.00	3.00	3.00	4.00	2.00
QCD+UCB	Worst	5.00	3.00	3.00	1.00	0.00	0.00
QCD+kIUCB	Unif	0.85	1.01	2.74	3.05	3.19	2.23
QCD+kIUCB	Worst	5.53	3.01	2.11	1.00	0.17	0.00

Table 10: Mean Number Of Declared Changes For Different Change-Points, $T = 5000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	4.69	7.46	7.80	6.44	4.55	2.77
GLRklUCB	Worst	11.37	7.56	4.42	2.26	0.72	0.11
QCD+UCB	Unif	3.89	7.17	7.98	6.80	4.89	3.02
QCD+UCB	Worst	11.08	7.91	5.51	2.82	0.88	0.26
QCD+klUCB	Unif	5.25	7.59	7.44	6.03	4.13	2.50
QCD+klUCB	Worst	11.88	6.72	3.71	2.07	0.71	0.09

(b) Deterministic Change-Points (Columns: N_C)						
Algorithm	PB	389	166	71	31	13
MASTER	Unif	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	3.36	5.19	9.05	8.86	5.38
GLRklUCB	Worst	11.54	8.56	5.05	3.03	0.98
QCD+UCB	Unif	1.00	3.00	9.00	9.00	7.00
QCD+UCB	Worst	9.00	8.00	5.00	4.00	1.00
QCD+klUCB	Unif	3.99	5.20	8.98	8.37	4.08
QCD+klUCB	Worst	12.98	8.57	5.18	3.01	0.98
						0.06

 Table 11: Mean Number Of Declared Changes For Different Change-Points, $T = 10000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	10.57	15.23	14.61	11.30	7.18	3.94
GLRklUCB	Worst	21.21	12.99	8.01	4.49	1.20	0.31
QCD+UCB	Unif	9.02	14.99	15.10	11.88	7.66	4.22
QCD+UCB	Worst	20.93	14.03	11.00	6.24	1.52	0.64
QCD+klUCB	Unif	11.60	15.25	13.91	10.36	6.43	3.50
QCD+klUCB	Worst	21.30	11.12	6.34	3.41	1.17	0.28

(b) Deterministic Change-Points (Columns: N_C)						
Algorithm	PB	631	252	100	40	16
MASTER	Unif	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	4.84	11.79	14.11	14.58	7.65
GLRklUCB	Worst	21.31	11.93	6.31	5.66	1.23
QCD+UCB	Unif	4.00	13.00	13.00	16.00	11.00
QCD+UCB	Worst	22.00	12.00	8.00	12.00	2.00
QCD+klUCB	Unif	5.25	11.81	13.96	13.70	7.03
QCD+klUCB	Worst	22.23	9.64	5.15	3.79	1.24
						0.11

 Table 12: Mean Number Of Declared Changes For Different Change-Points, $T = 20000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	23.18	30.38	26.91	18.80	10.69	5.21
GLRklUCB	Worst	38.72	21.87	13.59	7.26	1.92	0.56
QCD+UCB	Unif	20.43	30.33	27.88	20.02	11.56	5.55
QCD+UCB	Worst	38.69	25.34	20.58	11.81	2.78	1.17
QCD+klUCB	Unif	25.08	29.86	25.19	17.22	9.63	4.68
QCD+klUCB	Worst	37.43	18.71	10.61	5.20	1.85	0.51

(b) Deterministic Change-Points (Columns: N_C)						
Algorithm	PB	1025	381	142	53	20
MASTER	Unif	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	14.18	28.21	28.32	21.07	12.20
GLRklUCB	Worst	35.36	20.82	15.04	12.29	2.11
QCD+UCB	Unif	11.00	27.00	29.00	21.00	15.00
QCD+UCB	Worst	37.00	26.00	26.00	25.00	3.00
QCD+klUCB	Unif	14.94	27.12	26.18	19.49	10.57
QCD+klUCB	Worst	33.97	16.30	11.05	6.62	2.08
						0.93

Table 13: Mean Number Of Declared Changes For Different Change-Points, $T = 30000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	36.25	44.93	37.85	24.82	13.41	6.11
GLRklUCB	Worst	54.32	30.20	18.53	9.39	2.54	0.75
QCD+UCB	Unif	32.46	45.20	39.44	26.50	14.45	6.52
QCD+UCB	Worst	54.72	36.38	29.41	16.02	4.02	1.62
QCD+klUCB	Unif	38.81	43.89	35.38	22.59	12.04	5.54
QCD+klUCB	Worst	51.41	25.41	14.02	6.68	2.36	0.69

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	1362	486	174	62	23	8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	25.80	42.86	38.75	29.42	16.60	6.73
GLRklUCB	Worst	50.45	29.63	17.89	8.95	2.05	1.01
QCD+UCB	Unif	21.00	40.00	42.00	32.00	18.00	8.00
QCD+UCB	Worst	47.00	34.00	20.00	13.00	3.00	1.00
QCD+klUCB	Unif	28.17	41.30	36.74	27.47	14.55	5.13
QCD+klUCB	Worst	50.05	24.27	16.81	7.82	2.03	1.01

 Table 14: Mean Number Of Declared Changes For Different Change-Points, $T = 40000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	49.12	58.96	47.95	30.22	15.50	6.74
GLRklUCB	Worst	69.09	37.81	22.92	11.65	3.14	0.88
QCD+UCB	Unif	44.89	59.45	49.92	32.30	16.69	7.13
QCD+UCB	Worst	69.89	47.26	37.66	19.92	5.29	2.02
QCD+klUCB	Unif	52.45	57.35	44.60	27.43	13.98	6.14
QCD+klUCB	Worst	64.55	31.68	17.12	7.91	2.81	0.80

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	1666	578	200	70	25	9
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	35.21	56.62	51.68	37.09	20.23	7.43
GLRklUCB	Worst	66.30	33.00	26.14	11.18	5.94	0.83
QCD+UCB	Unif	32.00	60.00	51.00	39.00	21.00	8.00
QCD+UCB	Worst	68.00	40.00	46.00	21.00	11.00	3.00
QCD+klUCB	Unif	38.36	56.47	49.78	35.09	18.82	6.55
QCD+klUCB	Worst	64.75	28.52	19.85	8.88	3.36	0.79

 Table 15: Mean Number Of Declared Changes For Different Change-Points, $T = 50000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	62.57	72.53	57.36	34.97	17.32	7.30
GLRklUCB	Worst	83.18	45.08	27.44	13.90	3.71	1.01
QCD+UCB	Unif	57.48	73.43	59.88	37.49	18.54	7.64
QCD+UCB	Worst	84.42	58.05	45.35	23.09	6.44	2.36
QCD+klUCB	Unif	66.14	70.39	53.19	31.69	15.57	6.62
QCD+klUCB	Worst	76.85	37.52	19.82	8.98	3.18	0.92

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	1947	660	224	76	26	9
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	43.44	73.24	67.07	48.33	26.89	9.00
GLRklUCB	Worst	86.39	43.53	26.07	19.61	6.61	1.11
QCD+UCB	Unif	40.00	73.00	72.00	50.00	30.00	9.00
QCD+UCB	Worst	87.00	52.00	56.00	48.00	12.00	3.00
QCD+klUCB	Unif	45.52	71.44	65.64	42.99	23.87	8.94
QCD+klUCB	Worst	82.58	39.78	17.64	8.70	4.13	1.08

Table 16: Mean Number Of Declared Changes For Different Change-Points, $T = 60000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	75.81	86.09	66.55	39.34	18.77	7.68
GLRklUCB	Worst	97.03	52.35	31.42	15.38	4.25	1.12
QCD+UCB	Unif	70.20	86.99	69.40	42.17	20.12	8.02
QCD+UCB	Worst	98.60	68.75	52.67	26.29	7.63	2.71
QCD+klUCB	Unif	79.83	83.07	61.47	35.55	16.96	7.03
QCD+klUCB	Worst	88.81	43.35	22.48	9.95	3.52	1.02

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	2212	737	245	82	28	10
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	53.48	79.56	71.50	49.25	25.33	8.84
GLRklUCB	Worst	95.25	52.85	34.92	11.68	7.03	1.14
QCD+UCB	Unif	46.00	86.00	73.00	54.00	28.00	9.00
QCD+UCB	Worst	95.00	62.00	67.00	18.00	13.00	3.00
QCD+klUCB	Unif	58.17	79.64	66.73	45.14	22.18	8.01
QCD+klUCB	Worst	85.25	46.06	22.47	10.82	4.79	1.08

 Table 17: Mean Number Of Declared Changes For Different Change-Points, $T = 70000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	88.99	99.08	75.13	43.34	20.11	8.02
GLRklUCB	Worst	109.87	58.83	34.78	16.57	4.71	1.20
QCD+UCB	Unif	82.91	100.62	78.58	46.57	21.56	8.37
QCD+UCB	Worst	112.11	79.35	59.54	29.00	8.70	3.01
QCD+klUCB	Unif	93.44	95.51	69.35	39.19	18.23	7.36
QCD+klUCB	Worst	100.06	48.65	24.90	10.78	3.82	1.07

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	2464	808	265	87	29	10
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	63.38	97.11	90.14	53.90	25.12	10.82
GLRklUCB	Worst	105.55	51.41	38.35	18.39	3.67	1.55
QCD+UCB	Unif	47.00	95.00	96.00	57.00	27.00	11.00
QCD+UCB	Worst	100.00	75.00	66.00	33.00	7.00	4.00
QCD+klUCB	Unif	68.35	93.91	85.27	48.62	23.75	10.01
QCD+klUCB	Worst	97.04	40.04	25.01	12.63	3.52	1.47

 Table 18: Mean Number Of Declared Changes For Different Change-Points, $T = 80000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	102.41	112.32	83.45	47.05	21.56	8.32
GLRklUCB	Worst	122.49	65.82	38.40	17.82	5.17	1.29
QCD+UCB	Unif	95.73	113.86	87.40	50.49	23.02	8.69
QCD+UCB	Worst	125.33	89.94	66.57	31.59	9.80	3.26
QCD+klUCB	Unif	107.36	107.98	77.02	42.57	19.51	7.68
QCD+klUCB	Worst	111.00	54.20	27.06	11.60	4.10	1.13

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	2705	875	283	92	30	10
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	73.83	110.55	98.46	60.01	26.89	12.25
GLRklUCB	Worst	125.41	59.89	43.19	24.44	4.53	1.58
QCD+UCB	Unif	56.00	121.00	107.00	67.00	29.00	13.00
QCD+UCB	Worst	125.00	87.00	97.00	51.00	10.00	4.00
QCD+klUCB	Unif	77.03	105.86	93.66	53.34	23.11	10.61
QCD+klUCB	Worst	116.00	48.69	24.61	13.61	4.13	1.52

Table 19: Mean Number Of Declared Changes For Different Change-Points, $T = 90000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	116.07	124.83	91.32	50.66	22.68	8.66
GLRklUCB	Worst	134.99	72.26	41.53	19.21	5.68	1.37
QCD+UCB	Unif	108.69	126.67	95.84	54.55	24.21	9.00
QCD+UCB	Worst	138.30	100.33	73.03	34.17	10.77	3.56
QCD+klUCB	Unif	121.27	119.69	84.19	45.82	20.60	8.02
QCD+klUCB	Worst	121.74	59.12	29.22	12.29	4.34	1.20

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	2938	939	300	96	31	10
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	88.66	126.74	101.10	66.25	29.02	9.00
GLRklUCB	Worst	132.11	70.82	44.71	25.60	8.77	1.47
QCD+UCB	Unif	73.00	127.00	107.00	71.00	30.00	9.00
QCD+UCB	Worst	130.00	82.00	118.00	46.00	17.00	4.00
QCD+klUCB	Unif	93.90	123.01	93.99	59.78	24.69	8.58
QCD+klUCB	Worst	121.04	62.76	28.57	14.45	5.01	1.15

 Table 20: Mean Number Of Declared Changes For Different Change-Points, $T = 100000$, 4000 Runs.

(a) Random Change-Points (Columns: ξ)							
Algorithm	PB	0.3	0.4	0.5	0.6	0.7	0.8
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	129.43	137.21	99.21	54.05	23.78	8.88
GLRklUCB	Worst	147.03	79.08	45.28	20.73	6.07	1.46
QCD+UCB	Unif	121.62	139.51	104.03	58.01	25.35	9.23
QCD+UCB	Worst	150.80	110.35	79.12	36.51	11.77	3.77
QCD+klUCB	Unif	134.84	131.53	91.32	48.76	21.53	8.22
QCD+klUCB	Worst	131.94	64.20	31.33	12.93	4.59	1.27

(b) Deterministic Change-Points (Columns: N_C)							
Algorithm	PB	3163	1000	317	101	32	10
MASTER	Unif	0.00	0.00	0.00	0.00	0.00	0.00
MASTER	Worst	0.00	0.00	0.00	0.00	0.00	0.00
GLRklUCB	Unif	98.45	136.94	119.82	69.49	27.38	8.31
GLRklUCB	Worst	148.35	83.78	51.12	21.52	6.74	1.19
QCD+UCB	Unif	84.00	149.00	122.00	83.00	28.00	9.00
QCD+UCB	Worst	141.00	95.00	102.00	36.00	13.00	4.00
QCD+klUCB	Unif	104.45	134.02	113.89	62.68	25.81	7.95
QCD+klUCB	Worst	130.42	71.08	34.60	14.37	4.38	1.11

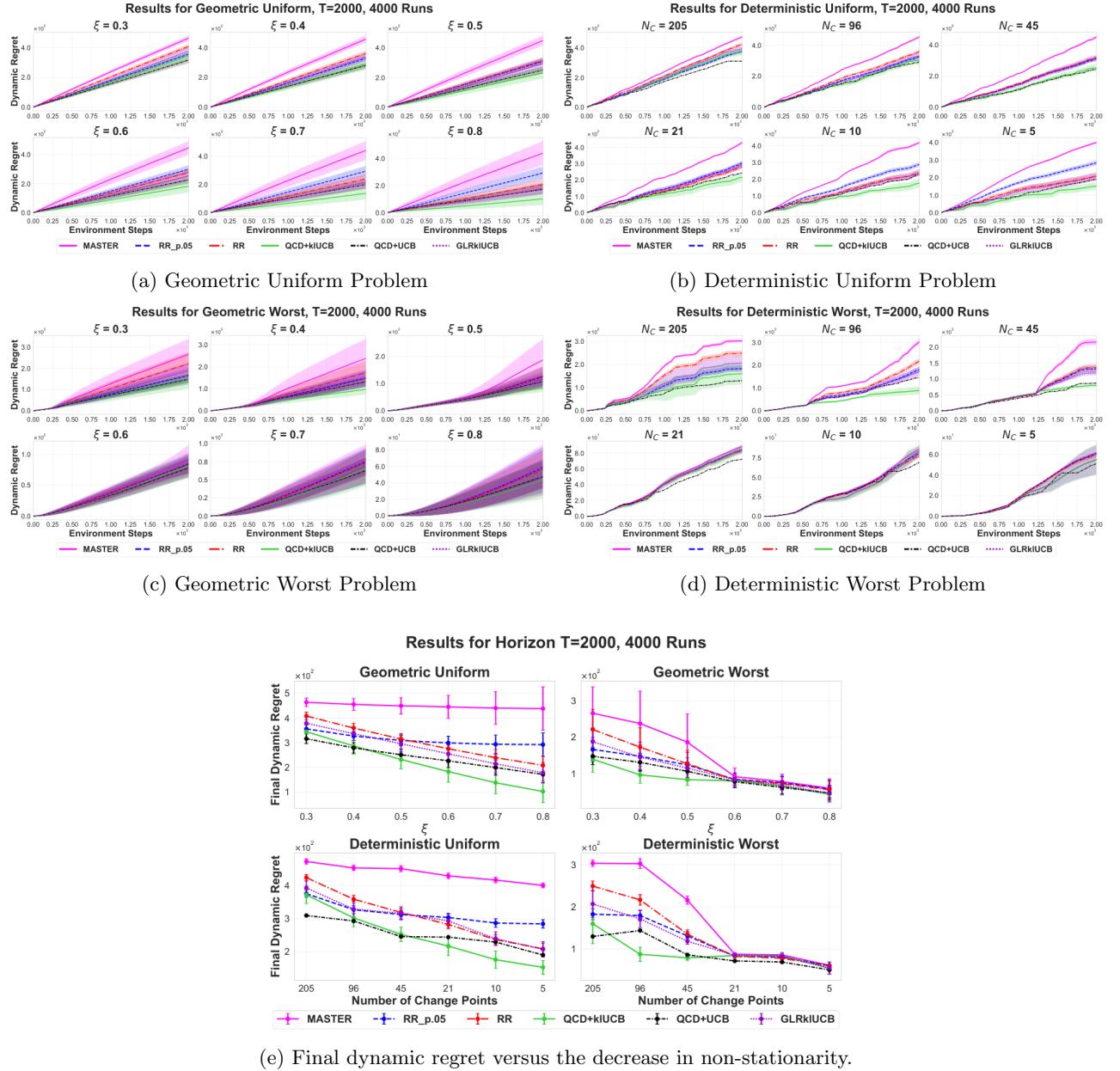


Figure 3: Results of the dynamic regret experiments $T = 2000$, averaged over 4000 independent runs.

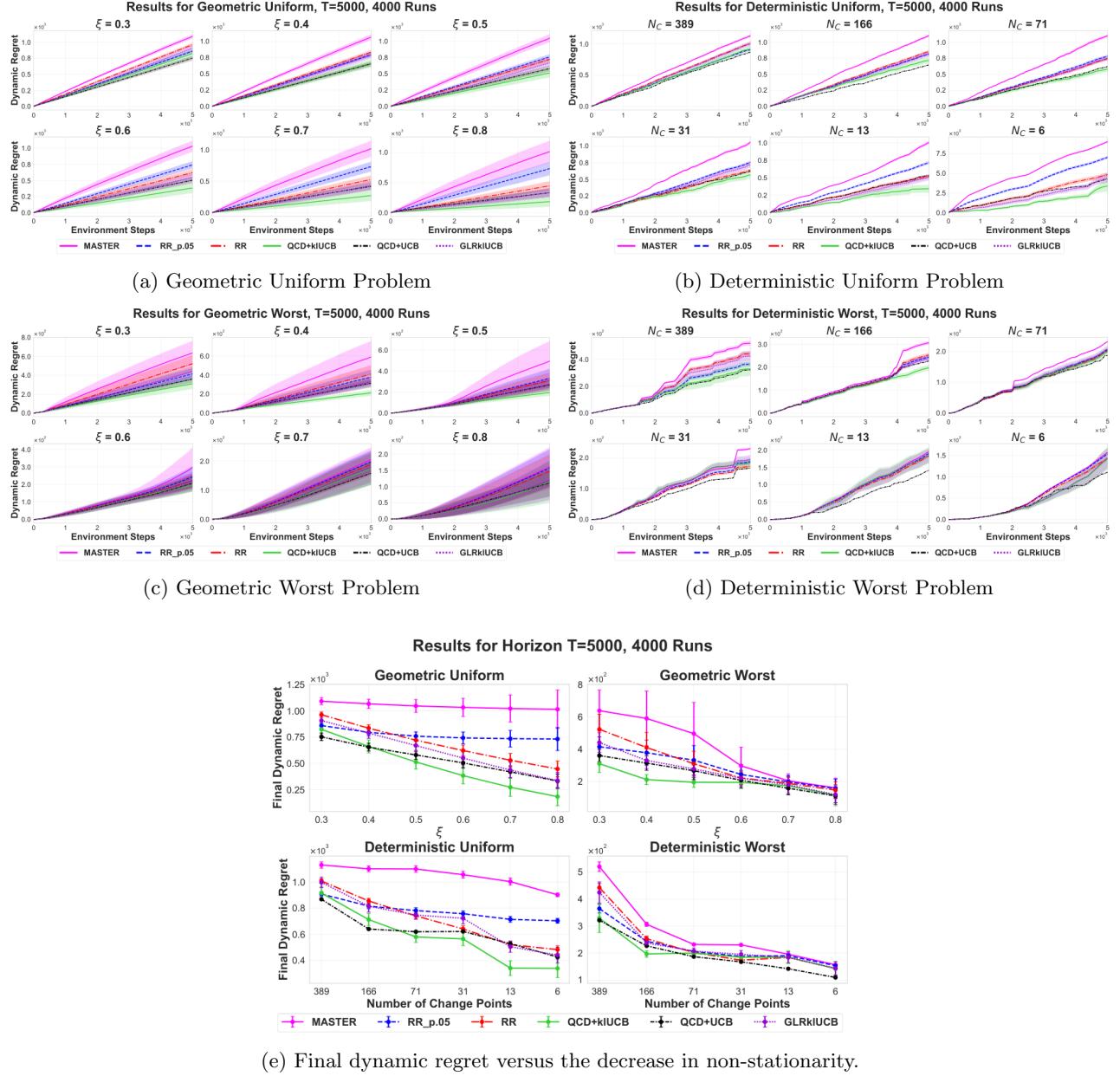


Figure 4: Results of the dynamic regret experiments $T = 5000$, averaged over 4000 independent runs.

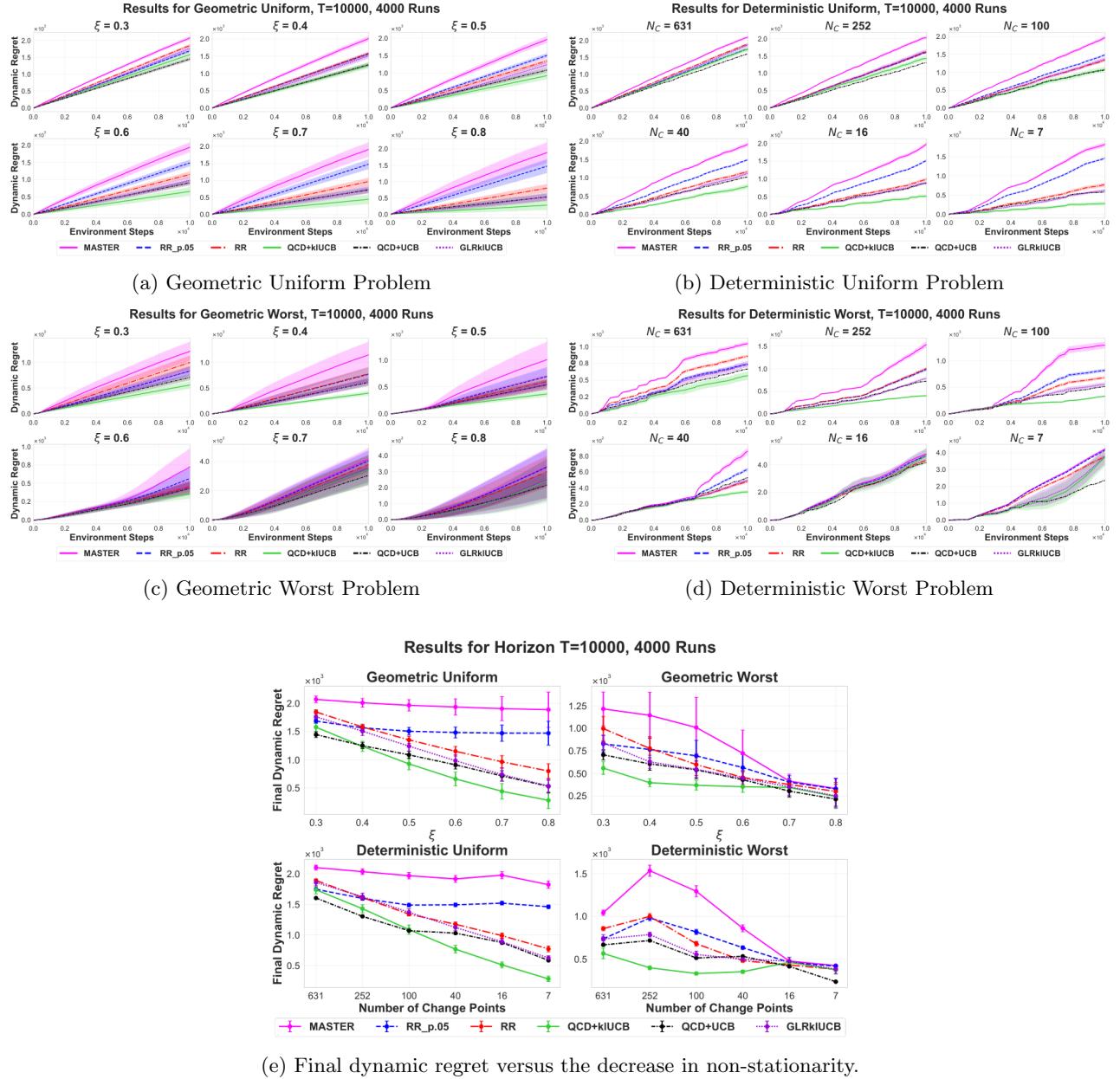


Figure 5: Results of the dynamic regret experiments $T = 10000$, averaged over 4000 independent runs.

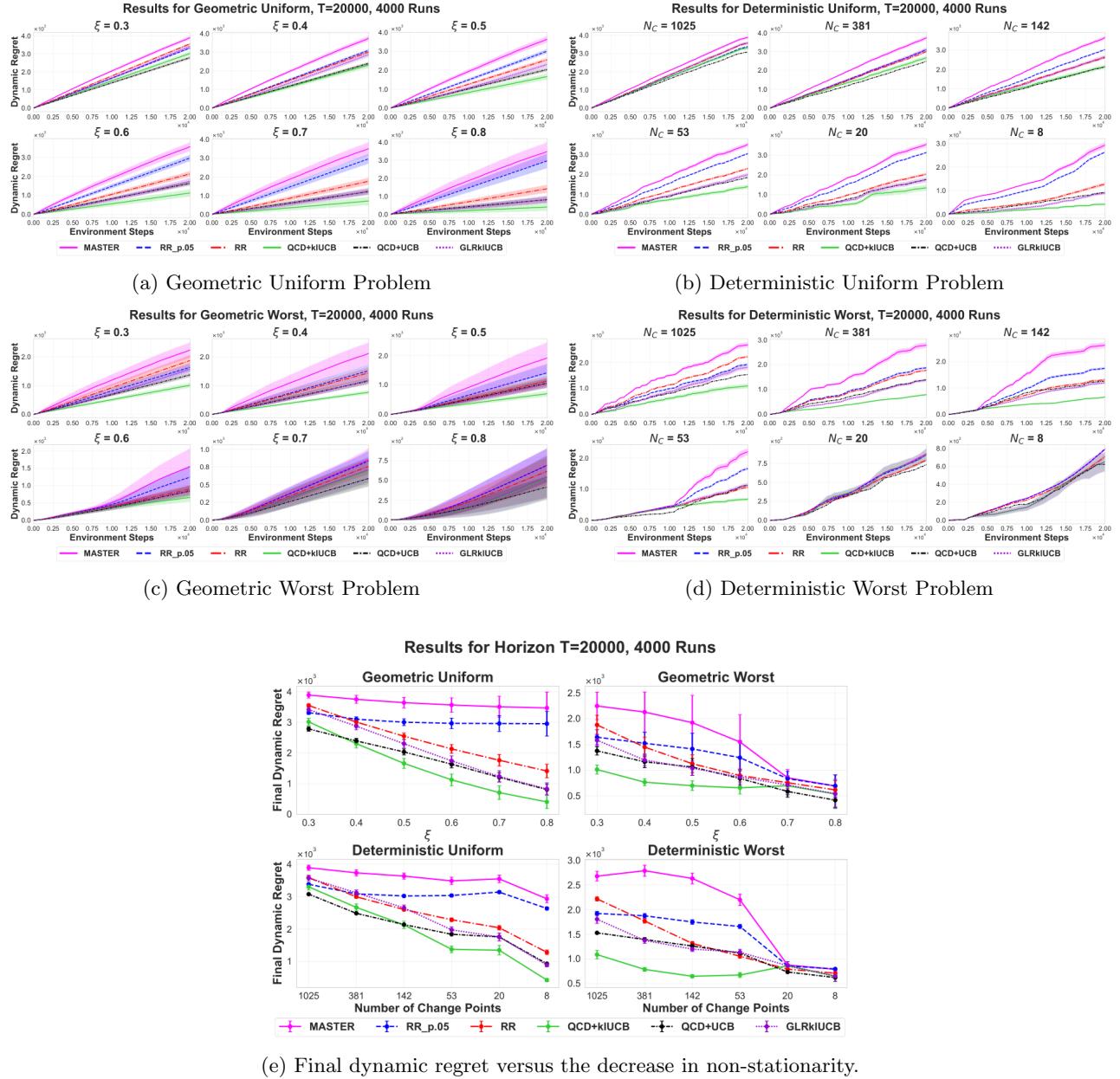


Figure 6: Results of the dynamic regret experiments $T = 20000$, averaged over 4000 independent runs.

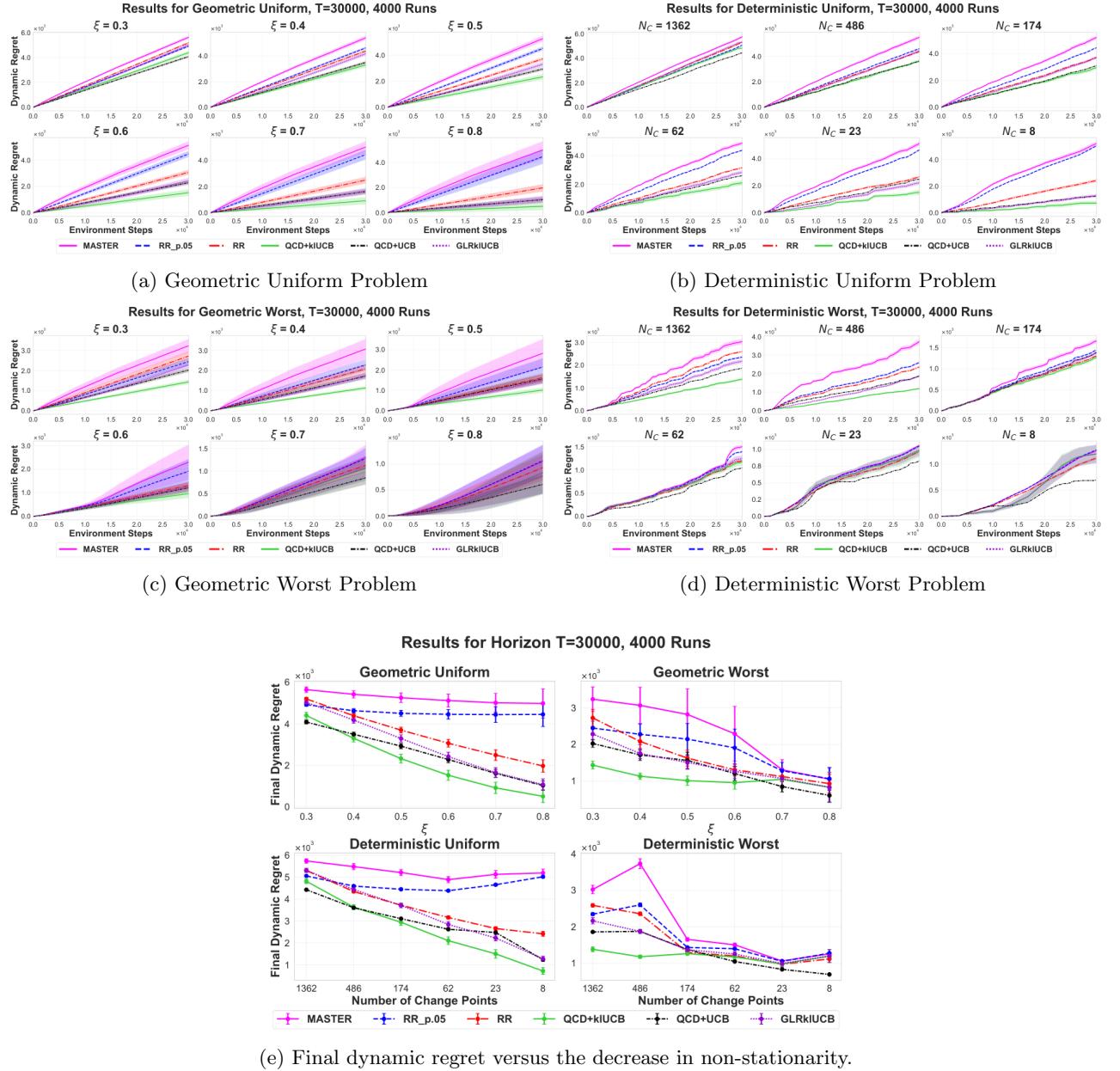


Figure 7: Results of the dynamic regret experiments $T = 30000$, averaged over 4000 independent runs.

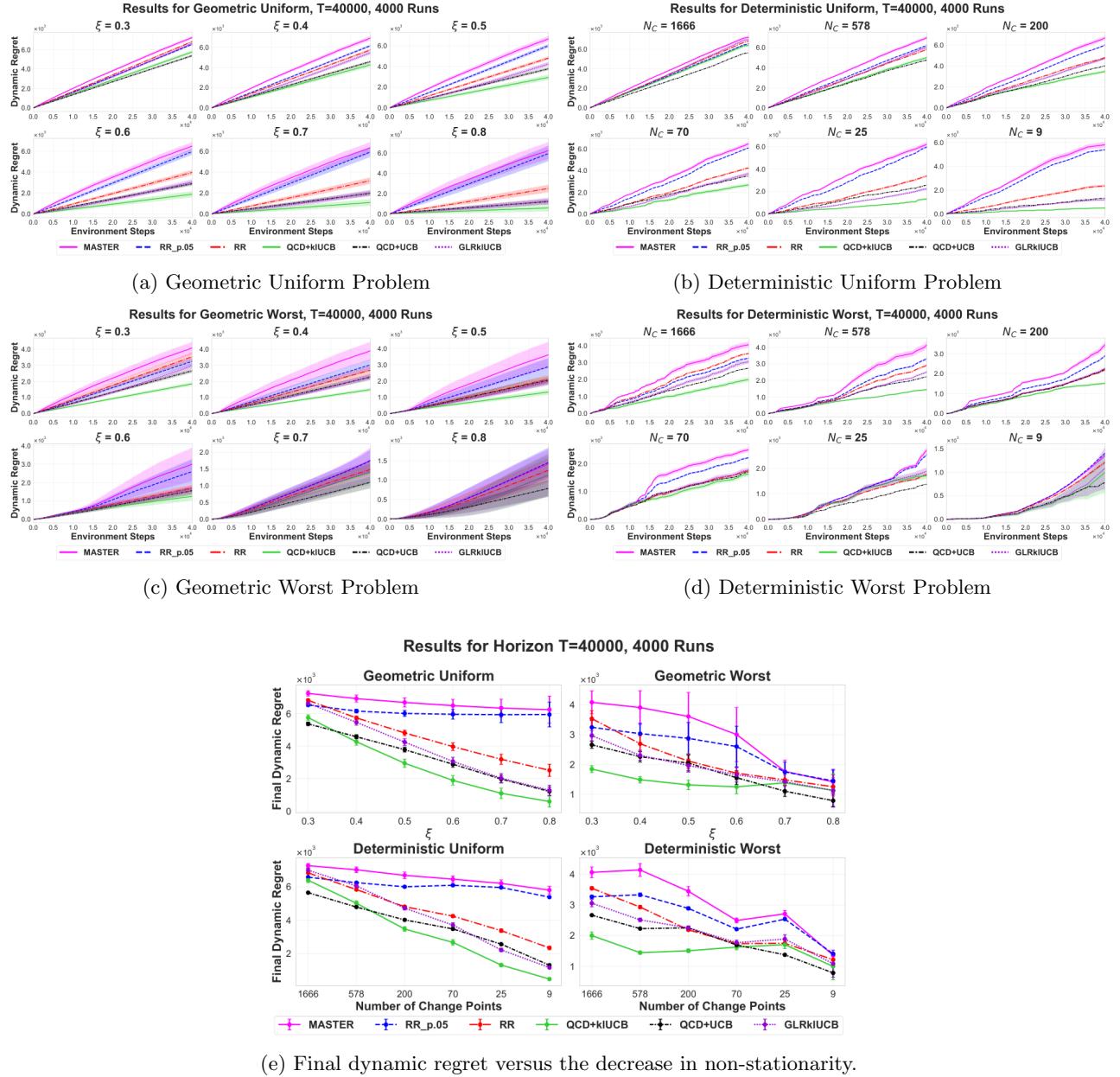


Figure 8: Results of the dynamic regret experiments $T = 40000$, averaged over 4000 independent runs.

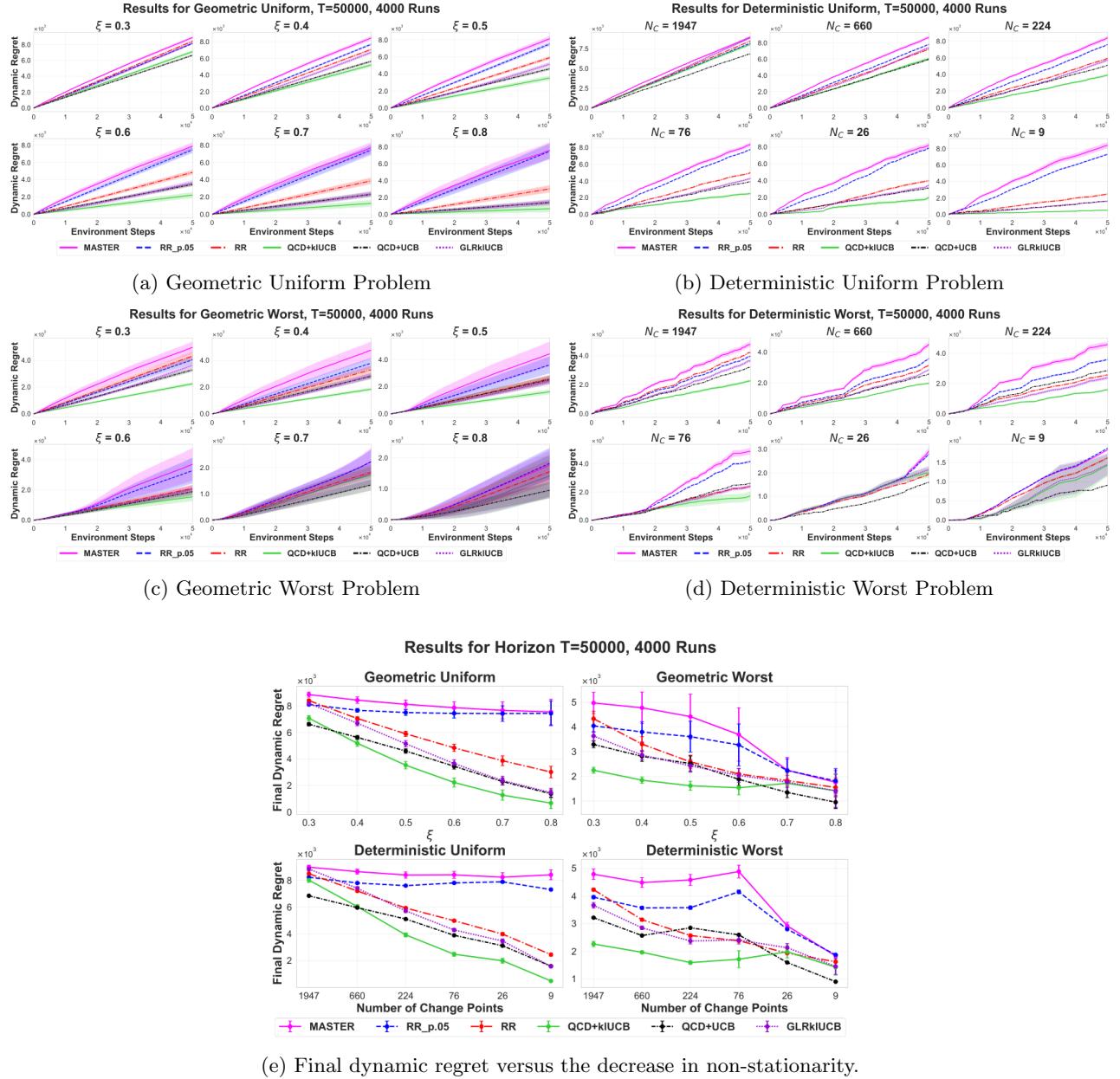


Figure 9: Results of the dynamic regret experiments $T = 50000$, averaged over 4000 independent runs.

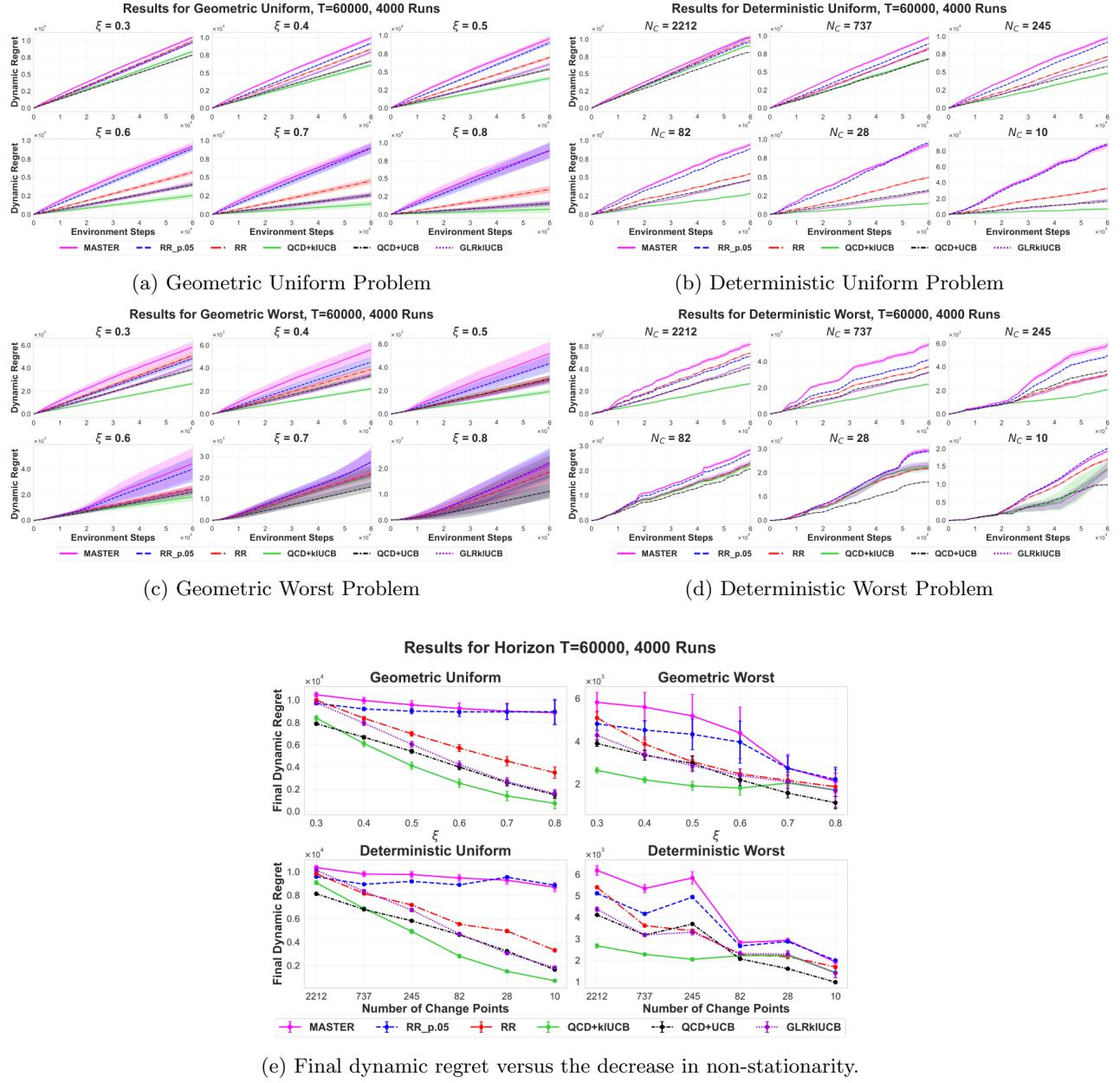


Figure 10: Results of the dynamic regret experiments $T = 60000$, averaged over 4000 independent runs.

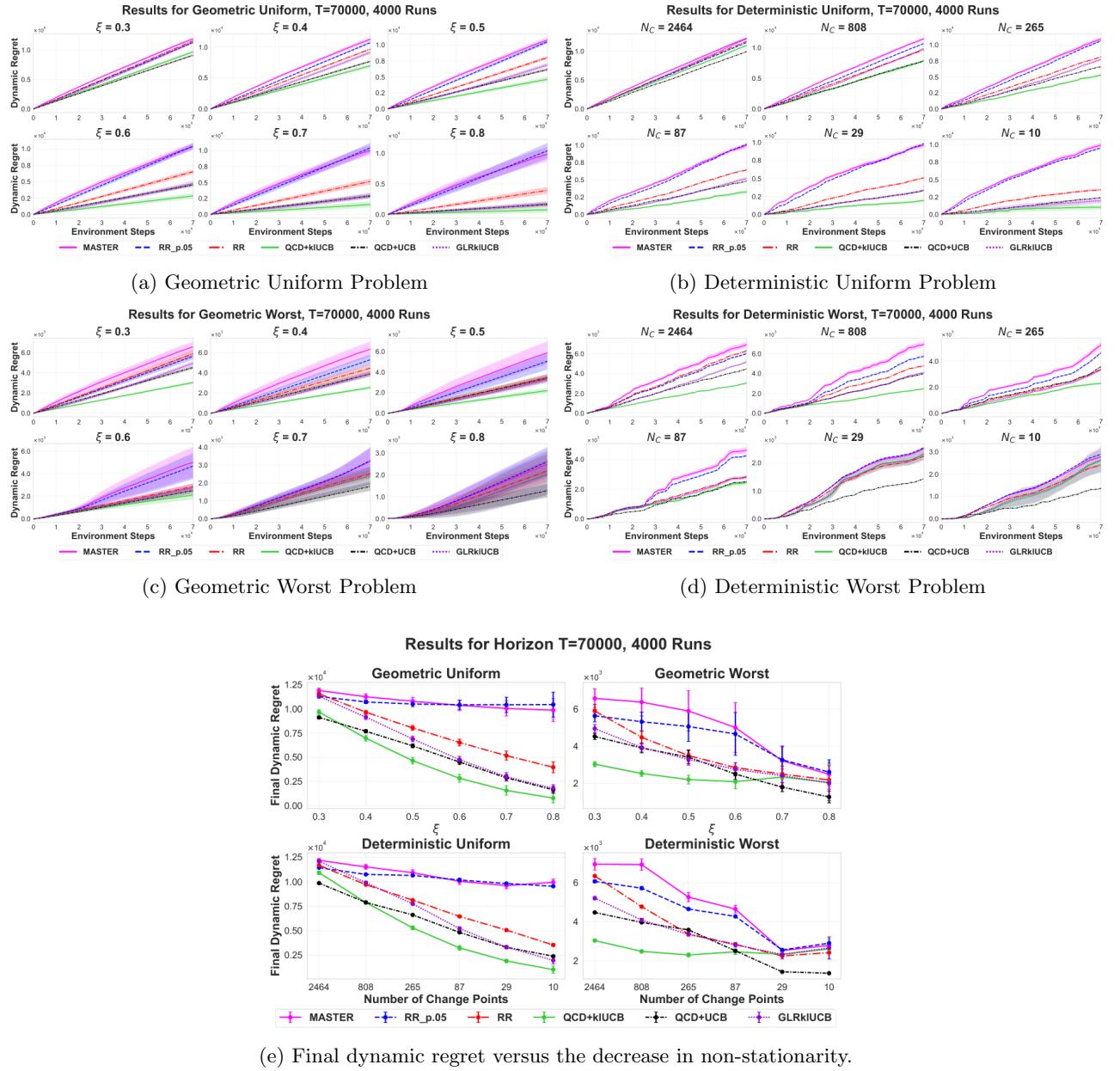


Figure 11: Results of the dynamic regret experiments $T = 70000$, averaged over 4000 independent runs.

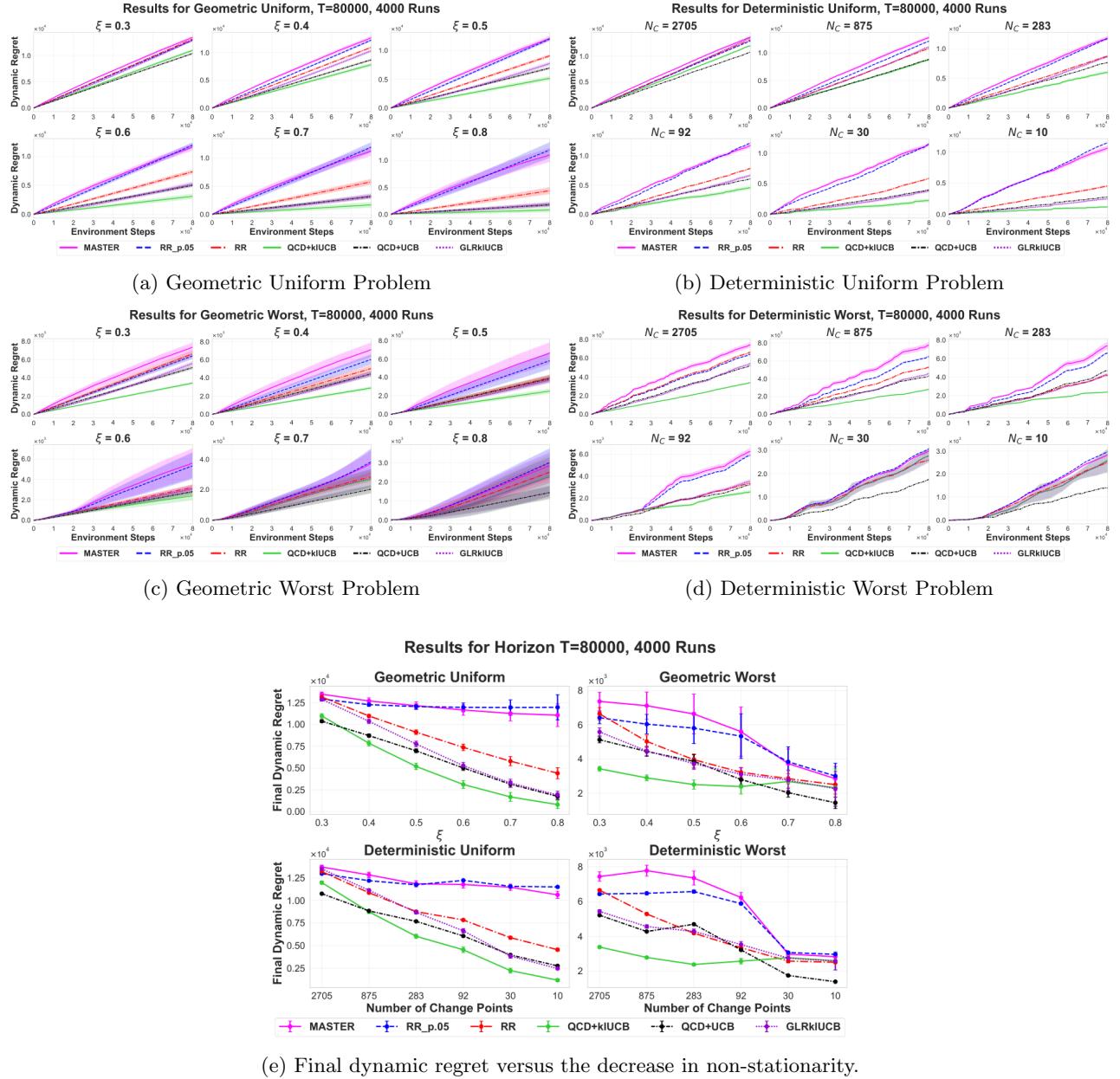


Figure 12: Results of the dynamic regret experiments $T = 80000$, averaged over 4000 independent runs.

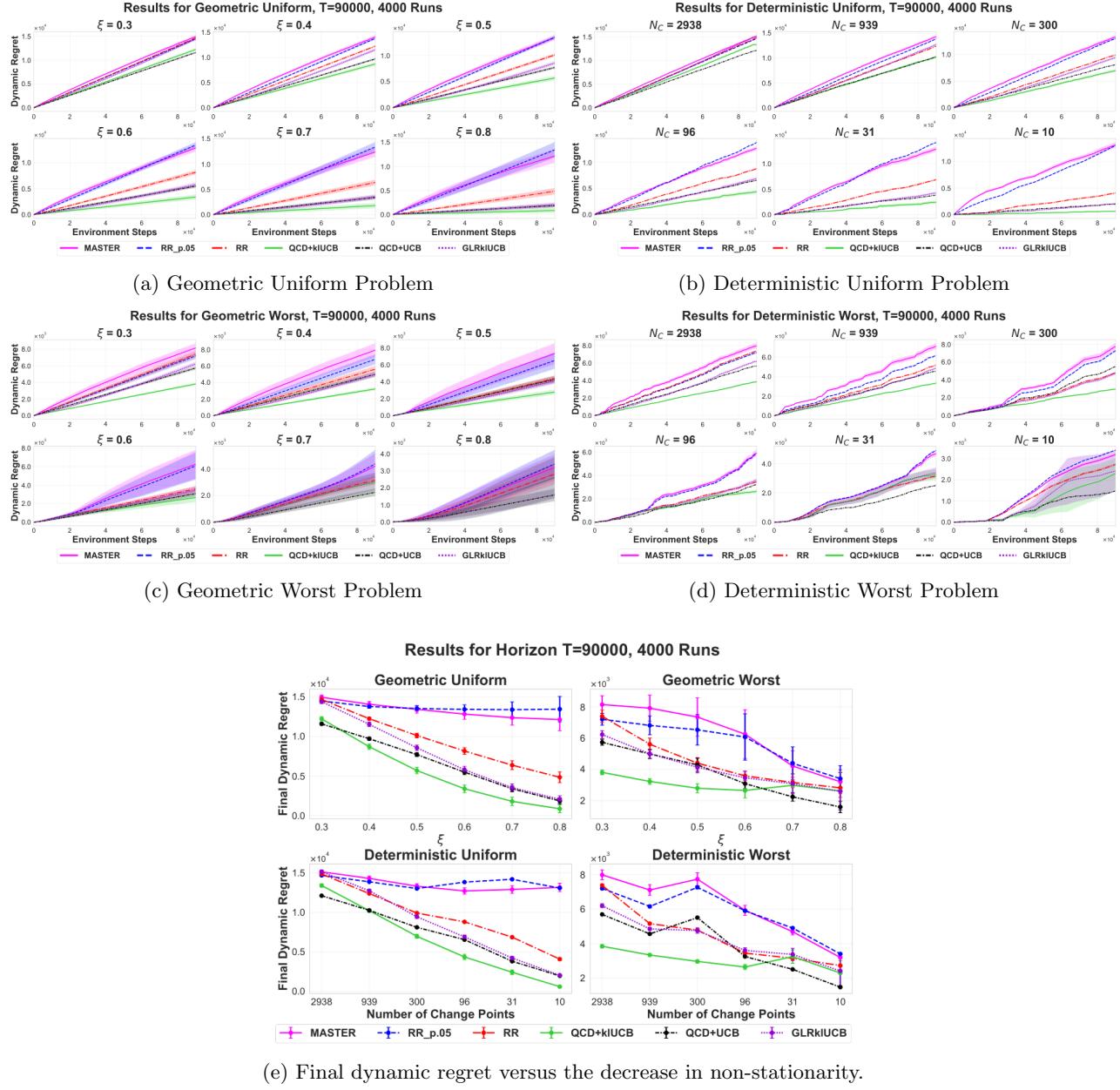


Figure 13: Results of the dynamic regret experiments $T = 90000$, averaged over 4000 independent runs.

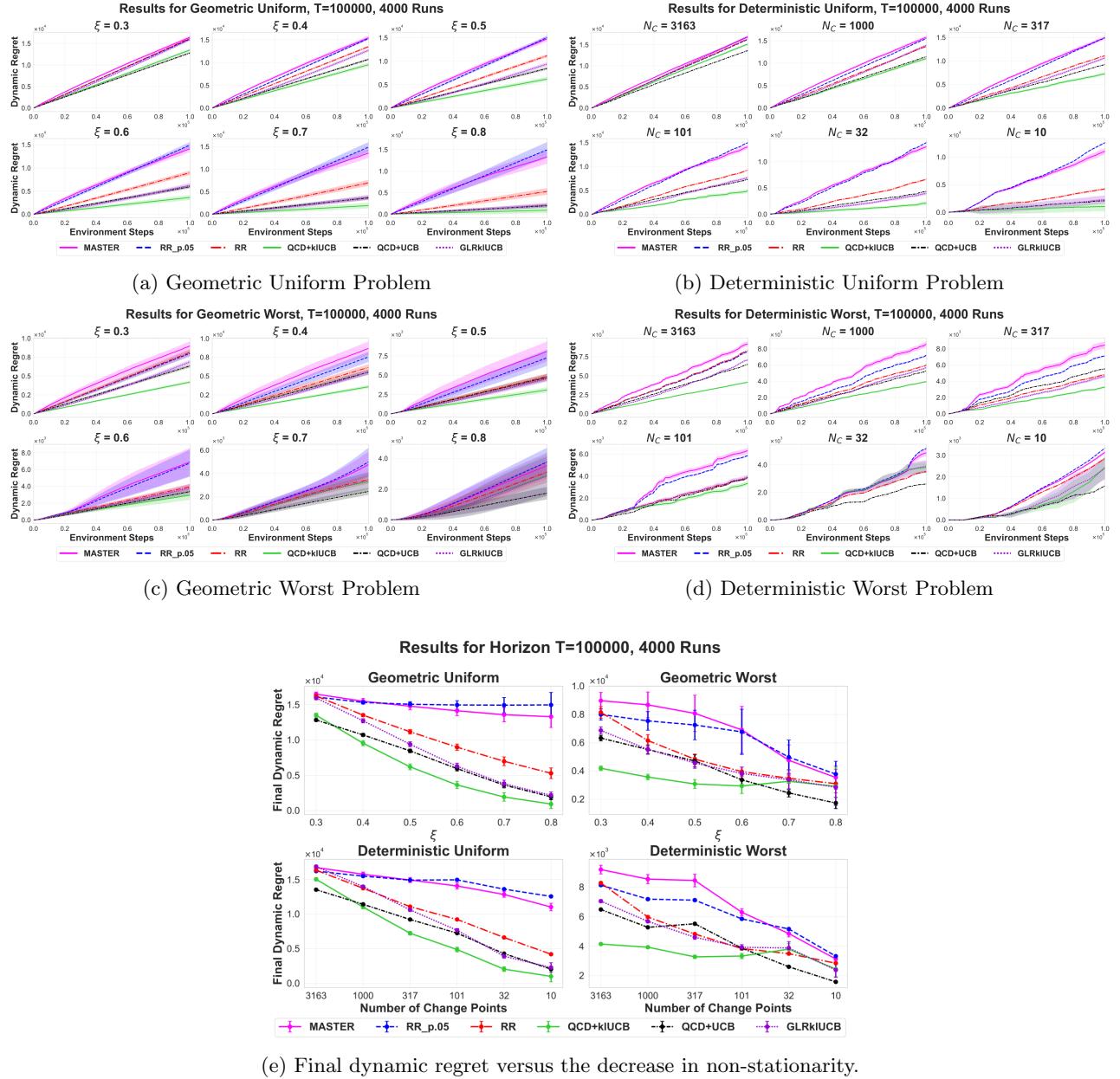


Figure 14: Results of the dynamic regret experiments $T = 100000$, averaged over 4000 independent runs.