# Detecting Rewards Deterioration in Episodic Reinforcement Learning

Ido Greenberg [1] Shie Mannor [1 2]

## Abstract

In many RL applications, once training ends, it is vital to detect any deterioration in the agent performance as soon as possible. Furthermore, it often has to be done without modifying the policy and under minimal assumptions regarding the environment. In this paper, we address this problem by focusing directly on the rewards and testing for degradation. We consider an episodic framework, where the rewards within each episode are not independent, nor identically-distributed, nor Markov. We present this problem as a multivariate mean-shift detection problem with possibly partial observations. We define the mean-shift in a way corresponding to deterioration of a temporal signal (such as the rewards), and derive a test for this problem with optimal statistical power. Empirically, on deteriorated rewards in control problems (generated using various environment modifications), the test is demonstrated to be more powerful than standard tests – often by orders of magnitude. We also suggest a novel Bootstrap mechanism for False Alarm Rate control (BFAR), applicable to episodic (non-i.i.d) signal and allowing our test to run sequentially in an online manner. Our method does not rely on a learned model of the environment, is entirely external to the agent, and in fact can be applied to detect changes or drifts in any episodic signal.

## 1. Introduction

Reinforcement learning (RL) algorithms have recently demonstrated impressive success in a variety of sequential decision-making problems (Badia et al., 2020; Hessel et al., 2018). While most RL works focus on the maximization of rewards under various conditions, a key issue in real-world RL tasks is the safety and reliability of the system (Dulac-Arnold et al., 2019; Chan et al., 2020), arising in both offline and online settings.

In **offline settings**, comparing the agent performance in different environments is important for generalization (e.g., in sim-to-real and transfer learning). The comparison may indicate the difficulty of the problem or help to select the right learning algorithms. Uncertainty estimation, which could help to address this challenge, is currently considered a hard problem in RL, in particular for model-free methods (Yu et al., 2020).

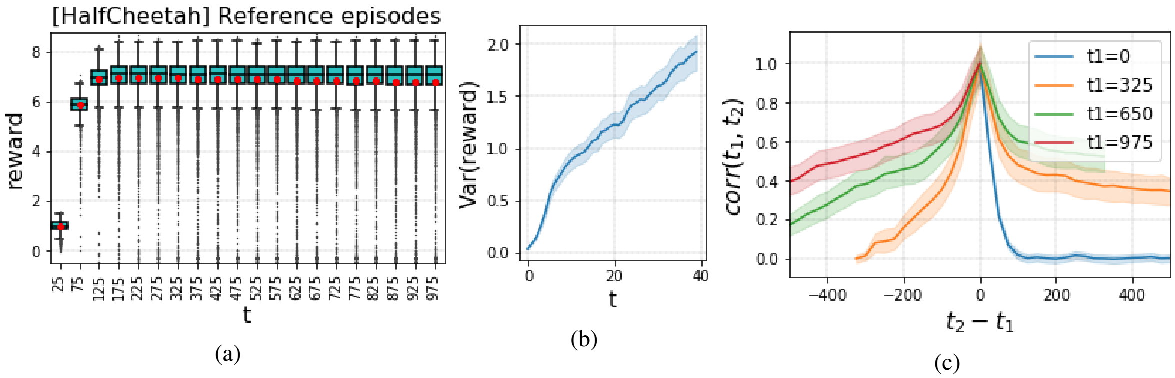
In **online settings**, where a fixed, already-trained agent runs continuously, its performance may be affected (gradually or abruptly) by changes in the controlled system or its surroundings, or when reaching unfamiliar states. Some works address robustness to changes (Lecarpentier & Rachelson, 2019; Lee et al., 2020), yet performance degradation is sometimes inevitable, and should be detected as soon as possible. The detection allows us to fall back into manual control, send the agent to re-train, guide diagnosis, or even bring the agent to halt. This problem is inherently different from robustness to changes during training: it focuses on safety and reliability, in post-training phase where intervention in the policy is limited or forbidden (Matsushima et al., 2020). It also operates in different time-scales: while training may take millions of episodes, changes should often be detected within tens of episodes, and critical failures – within less than an episode.

Such post-training performance-awareness is essential for any autonomous system in risk-intolerant applications, such as autonomous driving and medical devices. For example, when an autonomous car starts acting suspiciously with a passenger sitting inside, activating a training process and exploring for new policies is not an option. **The priority is to notice the suspicious behavior as soon as possible**, so that it can be alerted in time to save lives.

Many sequential statistical tests exist for detection of mean degradation in a random process. However, common methods (Page, 1954; Lan, 1994; Harel et al., 2014) assume independent and identically distributed (i.i.d) samples, while in RL the feedback from the environment is usually both highly correlated over consecutive time-steps, and varies over the life-time of the task (Korenkevych et al., 2019).

[1] Department of Electric Engineering, Technion, Israel [2] Nvidia Research. Correspondence to: Ido Greenberg <gido@campus.technion.ac.il>, Shie Mannor <shie@ee.technion.ac.il>.

*Proceedings of the $38^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).



(a)

(b)

(c)

Figure 1: Properties of the rewards of a fixed agent in HalfCheetah, estimated over $N = 10000$ episodes of $T = 1000$ time-steps: (a) distribution of rewards per time-step; (b) variance per time-step; (c) correlation($t_1, t_2$) vs. $t_2 - t_1$. The estimations are in resolution of 25 time-steps, i.e., every episode was split into 40 intervals of 25 consecutive steps, and each sample is the average over an interval.

This is demonstrated in Fig. 1.

A possible solution is to apply statistical tests to large blocks of data assumed to be i.i.d (Ditzler et al., 2015). This is particularly common in RL, where the episodic settings allow a natural blocks-partition (see for example Colas et al. (2019)). However, this approach requires complete episodes for change detection, while a faster response is often required. Furthermore, naively applying a statistical test on the accumulated feedback (e.g., sum of rewards) from complete episodes, ignores the dependencies within the episodes and misses vital information, leading to highly sub-optimal tests (as demonstrated in Section 6.2).

In this work, we devise an optimal test for detection of degradation of the rewards in an episodic RL task (or in any other episodic signal), based on the covariance structure within the episodes. Even in absence of the assumptions that guarantee its optimality, the test is still asymptotically superior to the common approach of comparing the mean reward (Colas et al., 2019). The test can detect changes and drifts in both the offline and the online settings defined above. Since tuning of the False Alarm Rate (FAR) of a sequential test usually relies on the underlying signal being i.i.d, we also suggest a novel Bootstrap mechanism for FAR control (BFAR) in sequential tests on episodic signals. The suggested procedures rely on the ability to estimate the correlations within the episodes, e.g., through a ”reference dataset” of episodes.

Since the test is applied directly to the rewards, it is model-free in the following senses: the underlying process is not assumed to be known, to be Markov, or to be observable at all (as opposed to other works, e.g., Banerjee et al. (2016)), and we require no knowledge about the process or the running policy. Furthermore, as the rewards are simply referred to as episodic time-series, the test can be similarly applied to detect changes in any episodic signal.

We demonstrate the new procedures in the environments of Pendulum (OpenAI), HalfCheetah and Humanoid (MuJoCo; Todorov et al., 2012). BFAR is shown to successfully control the false alarm rate. The suggested test detects degradation faster and more often than three alternative tests – in certain cases by orders of magnitude.

The paper is organized as follows: Section 3 formulates the offline setup (individual tests) and the online setup (sequential tests). Section 4 defines the model of an episodic signal, and derives an optimal degradation-test for such a signal. Section 5 shows how to adjust the test for online settings and control the false alarm rate. Section 6 describes the experiments, and Section 7 discusses related works.

To the best of our knowledge, we are the first to exploit the covariance between rewards in post-training phase to test for changes in RL-based systems. Our main contribution is an optimal test that can detect deterioration in agent rewards and other episodic signals reliably, in much shorter times than current standard tests. We also suggest a novel bootstrap mechanism to control false alarm rate of such tests on episodic (non-i.i.d) data. Finally, we lay a new framework for statistical tests on episodic signals, which opens the way for further research on this problem.

## 2. Preliminaries

**Reinforcement learning and episodic framework:** A Reinforcement Learning (RL) problem is usually modeled as a sequential *decision process*, where a learning *agent* has to repeatedly make decisions that affect its future states and rewards. The process is often organized as a finite sequence of time-steps (an *episode*) that repeats multiple times in different variants, e.g., with different initial states. Common examples are board and video games (Brockman et al., 2016), as well as more realistic problems such as autonomous driving tasks.

Once the agent is fixed (which is the case in this work), the rewards of the decision process essentially reduce to a (decision-free) random process $\{X_t\}_{t=1}^n$, which can be defined by its PDF ($f_{\{X_t\}_{t=1}^n} : \mathbb{R}^n \rightarrow [0, \infty)$). $\{X_t\}$ usually depend on each other: even in the popular *Markov Decision Process* (Bellman, 1957), where the dependence goes only a single step back, long-term correlations may still carry information if the states are not observable by the agent.

**Hypothesis tests:** Consider a parametric probability function $p(X|\theta)$ describing a random process, and consider two different hypotheses $H_0, H_A$ determining the value (*simple hypothesis*) or allowed values (*complex hypothesis*) of $\theta$. When designing a test to decide between the hypotheses, the basic metrics for the test efficacy are its *significance* $P(\text{not reject } H_0|H_0) = 1 - \alpha$ and its *power* $P(\text{reject } H_0|H_A) = \beta$. A hypothesis test with significance $1 - \alpha$ and power $\beta$ is *optimal* if any test with as high significance $1 - \tilde{\alpha} \geq 1 - \alpha$ has smaller power $\tilde{\beta} \leq \beta$.

The likelihood of the hypothesis $H : \theta \in \Theta$ given data $X$ is defined as $L(H|X) = \sup_{\theta \in \Theta} p(X|\theta)$. According to Neyman-Pearson lemma (Neyman et al., 1933), a threshold-test on the likelihood ratio $LR(H_0, H_A|X) = L(H_0|X)/L(H_A|X)$ is optimal. The threshold is uniquely determined by the desired significance level $\alpha$, though is often difficult to practically calculate given $\alpha$.

In many practical applications, a hypothesis test is repeatedly applied as the data change or grow, a procedure known as a *sequential test*. If the null hypothesis $H_0$ is true, and any individual hypothesis test falsely rejects $H_0$ with some probability $\alpha$, then the probability that at least one of the multiple tests will reject $H_0$ is $\alpha_0 > \alpha$, termed *family-wise type-I error* (or *false alarm rate* when associated with frequency). See Appendix A for more details about hypothesis testing and sequential tests in particular.

Common approaches for sequential tests, such as CUSUM (Page, 1954; Ryan, 2011) and $\alpha$-spending functions (Lan, 1994; Pocock, 1977), usually require strong assumptions such as independence or normality, as further discussed in Appendix B.

## 3. Problem Setup

In this work, we consider two setups where detecting performance deterioration is important – sequential degradation-tests and individual degradation-tests. The individual tests, in addition to their importance in offline settings such as sim-to-real and transfer learning, are used in this work as building-blocks for the online sequential tests.

Both setups assume a fixed agent that was previously trained, and aim to detect whenever the agent performance begins to deteriorate, e.g., due to environment changes. The ability to notice such changes is essential in many real-world problems, as explained in Section 1.

**Setup 1** (**Individual degradation-test).** We consider a fixed trained agent (policy must be fixed but is not necessarily optimal), whose rewards in an episodic environment (with episodes of length $T$) were previously recorded for multiple episodes (the *reference dataset*). The agent runs in a new environment for $n$ time-steps (both $n < T$ and $n \geq T$ are valid). The goal is to decide whether the rewards in the new environment are smaller than the original environment or not. If the new environment is identical, the probability of a false alarm must not exceed $\alpha$.

**Setup 2** (**Sequential degradation-test).** As in Setup 1, we consider a fixed trained agent with reference data of multiple episodes. This time the agent keeps running in the same environment, and at a certain point in time its rewards begin to deteriorate, e.g., due to changes in the environment. The goal is to alert to the degradation as soon as possible. As long as the environment has not changed, the probability of a false alarm must not exceed $\alpha_0$ per $\tilde{h}$ episodes.

Note that while in this work the setups focus on degradation, they can be easily modified to look for any change (as positive changes may also indicate the need for further training, for example).

## 4. Optimization of Individual Tests

To tackle the problem of Setup 1, we first define the properties of an episodic signal and the general assumptions regarding its degradation.

**Definition 4.1** ($T$-long episodic signal)**.** *Let $n, T \in \mathbb{N}$, and write $n = KT + \tau_0$ (for non-negative integers $K, \tau_0$ with $\tau_0 \leq T$). A sequence of real-valued random variables $\{X_t\}_{t=1}^n$ is a $T$-long episodic signal, if its joint probability density function can be written as*

$$f_{\{X_t\}_{t=1}^n}(x_1, ..., x_n) = \left[\prod_{k=0}^{K-1} f_{\{X_t\}_{t=1}^T}(\{x_{kT+t}\}_{t=1}^T)\right] \cdot f_{\{X_t\}_{t=1}^{\tau_0}}(\{x_{KT+t}\}_{t=1}^{\tau_0}) \tag{1}$$

*(where an empty product is defined as 1). We further denote $\boldsymbol{\mu_0} := E[(X_1, ..., X_T)^\top] \in \mathbb{R}^T$, $\Sigma_0 := Cov((X_1, ..., X_T)^\top, (X_1, ..., X_T)) \in \mathbb{R}^{T \times T}$.*

Note that the episodic signal consists of i.i.d episodes, but is not assumed to be independent or identically-distributed within the episodes – a setup particularly popular in RL.

In the analysis below we assume that both $\boldsymbol{\mu_0}$ and $\Sigma_0$ are known. This can be achieved either with detailed domain knowledge, or by estimation from the recorded reference dataset of Setup 1, assuming it satisfies Eq. (1). The estimation errors decrease as $\mathcal{O}(1/\sqrt{N})$ with the number $N$ of reference episodes, and are distributed according to the Central Limit Theorem (for means) and Wishart distribution (K. V. Mardia & Bibby, 1979) (for covariance). While in this work we use up to $N = 10000$ reference episodes, Appendix J shows that $N = 300$ reference episodes are sufficient for reasonable results in HalfCheetah, for example. Note that correlations estimation has been already discussed in several other RL works (Alt et al., 2019).

Fig. 1 demonstrates the estimation of mean and covariance parameters for a trained agent in the environment of HalfCheetah, from a reference dataset of $N = 10000$ episodes. This also demonstrates the non-trivial correlations structure in the environment. According to Fig. 1b, the variance in the rewards varies and does not seem to reach stationarity within the scope of an episode. Fig. 1c shows the autocorrelation function $ACF(t_2 - t_1) = corr(t_1, t_2)$ for different reference times $t_1$. The correlations clearly last for hundreds of time-steps, and depend on the time $t_1$ rather than merely on the time-difference $t_2 - t_1$. This means that the autocorrelation function is not expressive enough for the actual correlations structure.

Once the per-episode parameters $\boldsymbol{\mu_0} \in \mathbb{R}^T, \Sigma_0 \in \mathbb{R}^{T\times T}$ are known, the mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n\times n}$ of the whole signal can be derived directly: $\boldsymbol{\mu}$ consists of periodic repetitions of $\boldsymbol{\mu_0}$, and $\Sigma$ consists of copies of $\Sigma_0$ as $T \times T$ blocks along its diagonal. For both, the last repetition is cropped if $n$ is not an integer multiplication of $T$. In other words, by taking advantage of the episodic setup, we can treat the temporal univariate non-i.i.d signal as a multivariate signal with easily-measured mean and covariance – even if the signal ends in the middle of an episode.

The degradation in the signal $X = \{X_t\}_{t=1}^n$ is defined through the difference between two hypotheses. The null hypothesis $H_0$ states that $X$ is a $T$-long episodic signal with expectations $\boldsymbol{\mu_0} \in R^T$ and invertible covariance matrix $\Sigma_0 \in R^{T\times T}$. Our first alternative hypothesis ($H_A$) – uniform degradation – states that $X$ is a $T$-long episodic signal with the same covariance $\Sigma_0$ but smaller expectations: $\exists \epsilon \geq \epsilon_0, \forall 1 \leq t \leq T : (\tilde{\boldsymbol{\mu}}_{\mathbf{0}})_t = (\boldsymbol{\mu_0})_t - \epsilon$. Note that this hypothesis is complex ($\epsilon \geq \epsilon_0$), where $\epsilon_0$ can be tuned according to the minimal degradation magnitude of interest. In fact, Theorem 4.1 shows that the optimal corresponding test is independent of the choice of $\epsilon_0$.

**Theorem 4.1** (Optimal test for uniform degradation)**.** *Define the uniform-degradation weighted-mean $s_{unif}(X) := W \cdot X$, where $W := \mathbf{1}^\top \cdot \Sigma^{-1} \in \mathbb{R}^n$ (and $\mathbf{1}$ is the all-1 vector). If the distribution of $X$ is multivariate normal, then a threshold-test on $s_{unif}$ is optimal.*

*Proof Sketch (see full proof in Appendix E).* According to Neyman-Pearson lemma (Neyman et al., 1933), a threshold-test on the likelihood-ratio (LR) between $H_0$ and $H_A$ is optimal. Since $H_A$ is complex, the LR is a minimum over $\epsilon \in [\epsilon_0, \infty)$. Lemma 1 shows that $\exists s_0 : s_{unif} \geq s_0 \Rightarrow \epsilon = \epsilon_0$ and $s_{unif} \leq s_0 \Rightarrow \epsilon = \epsilon(s_{unif})$. The rest of the proof substitutes $\epsilon$ in both domains of $s_{unif}$ to prove monotony of the LR in $s_{unif}$, from which we can conclude monotony in $s_{unif}$ over all $\mathbb{R}$. □

Following Theorem 4.1, we define the Uniform Degradation Test (**UDT**) to be a threshold-test on $s_{unif}$, i.e., "declare a degradation if $s_{unif} < \kappa$" for a pre-defined $\kappa$. If the weights are calculated in advance, $s_{unif}$ can be calculated in $\mathcal{O}(n)$ time, and updated in $\mathcal{O}(1)$ with every new sample.

Recall that test optimality is defined in Section 2 as having maximal power per significance level. To achieve the significance $\alpha$ required in Setup 1, we apply a bootstrap mechanism that randomly samples episodes from the reference data and calculates the corresponding statistic (e.g., $s_{unif}$). This yields a bootstrap-estimate of the statistic's distribution under $H_0$, and the $\alpha$-quantile of the estimated distribution is chosen as the test-threshold ($\kappa = q_\alpha(s_{unif}|H_0)$).

$H_A$ is intended for degradation in a temporal signal, and derives a different optimal statistic than standard mean-change tests in multivariate variables (e.g., Hotelling). In Section 6, this is indeed demonstrated to be more powerful for rewards degradation. Also note that by explicitly referring to the temporal dimension, we allow detections even before the first episode is completed.

Theorem 4.1 relies on multivariate normality assumption, which is often too strong for real-world applications. Theorem 4.2 guarantees that if we remove the normality assumption, it is still beneficial to look into the episodes instead of considering them as atomic blocks; that is, UDT is still asymptotically better than a test on the simple mean $s_{simp} = \sum_{t=1}^n X_t/n$. Note that "asymptotic" refers to the signal length $n \to \infty$ (while $T$ remains constant), and is translated in the sequential setup into a "very long lookback-horizon $h$" (rather than very long running time).

**Theorem 4.2** (Asymptotic power of UDT)**.** *Denote the length of the signal $n = K \cdot T$, assume a uniform degradation of size $\frac{\epsilon}{\sqrt{K}}$, and let two threshold-tests $\tau_{simp}$ on $s_{simp}$ and UDT on $s_{unif}$ be tuned to have significance $\alpha$. Then*

$$\begin{aligned}\lim_{K\to\infty} P\left(\tau_{simp} \textit{ rejects } \middle| H_A\right) = \Phi\left(q_\alpha^0 + \frac{\epsilon T}{\sqrt{\mathbf{1}^\top \Sigma_0 \mathbf{1}}}\right) \\ \leq \Phi\left(q_\alpha^0 + \epsilon\sqrt{\mathbf{1}^\top \Sigma_0^{-1} \mathbf{1}}\right) = \lim_{K\to\infty} P\left(\textit{UDT rejects } \middle| H_A\right)\end{aligned} \tag{2}$$

*where $\Phi$ is the CDF of the standard normal distribution, and $q_\alpha^0$ is its $\alpha$-quantile.*

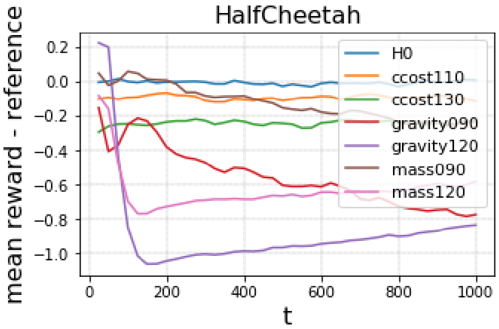


Figure 2: Rewards degradation of a fixed agent in HalfCheetah following changes in gravity, mass, and control-cost, over $N = 5000$ episodes per scenario.

*Proof Sketch (see full proof in Appendix E).* Since the episodes of the signal are i.i.d, both $s_{simp}$ and $s_{unif}$ are asymptotically normal according to the Central Limit Theorem. The means and variances of both statistics are calculated in Lemma 2. Calculation of the variance of $s_{unif}$ relies on writing $s_{unif}$ as a sum of linear transformations of $X$ ($s_{unif} = \sum_{i=1}^{n} (\Sigma^{-1})_i X$), and using the relation between $\Sigma$ and $\Sigma_0$. The inequality between the resulted powers is shown to be equivalent to a matrix-form of the means-inequality, and is proved using Cauchy-Schwarz inequality for $\Sigma_0^{-1/2}\mathbf{1}$ and $\Sigma_0^{1/2}\mathbf{1}$. □

Motivated by Theorem 4.2, we define $G^2 := \frac{(\mathbf{1}^\top \Sigma_0^{-1}\mathbf{1})(\mathbf{1}^\top \Sigma_0 \mathbf{1})}{T^2}$ to be the asymptotic power gain of UDT, quantify it, and show that it increases with the heterogeneity of the spectrum of $\Sigma_0$. In particular, if the rewards are heterogeneous, the suggested test is guaranteed to detect uniform degradation with much higher probability than the standard mean-test.

**Proposition 4.1** (Asymptotic power gain)**.** *$G^2 = 1 + \sum_{i,j=1}^{T} w_{ij}(\lambda_i - \lambda_j)^2$, where $\{\lambda_i\}_{i=1}^{T}$ are the eigenvalues of $\Sigma_0$ and $\{w_{ij}\}_{i,j=1}^{T}$ are positive weights.*

*Proof Sketch (see full proof in Appendix E).* The result can be calculated after diagonalization of $\Sigma_0$, and the weights $\{w_{ij}\}$ are derived from the diagonalizing matrix. □

So far we assumed uniform degradation. In the context of RL, such a model may refer to changes in constant costs or action costs, as well as certain dynamics whose change influences various states in a similar way. Fig. 2 demonstrates the empiric degradation in the rewards of a fixed agent in HalfCheetah, following changes in gravity, mass and control-cost. It seems that some modifications indeed cause a quite uniform degradation, while in others the degradation is mostly restricted to certain ranges of time.

To model effects that are less uniform in time we suggest a partial degradation hypothesis, where some (unknown) entries of $\boldsymbol{\mu_0}$ are reduced by $\epsilon > 0$, and others do not change. The number $m = p \cdot T$ of the reduced entries is defined by a parameter $p \in (0, 1)$.

This time, calculation of the optimal test-statistic through the LR yields a minimum over $\binom{T}{m}$ possible subsets of decreased entries, which is computationally heavy. However, Theorem 4.3 shows that if we optimize for small values of $\epsilon$ (where optimality is indeed most valuable), a near-optimal statistic is $s_{part}$, which is the sum of the $m = p \cdot T$ smallest time-steps of $(X - \boldsymbol{\mu})$ after a $\Sigma^{-1}$-transformation (see formal definition in Definition D.11). The resulted time-complexity is $\mathcal{O}(nT)$. We define the Partial Degradation Test (**PDT**) as a threshold-test on $s_{part}$ with a parameter $p$.

**Theorem 4.3** (Near-optimal test for uniform degradation)**.** *Assume that $X$ is multivariate normal, and let $P_\alpha$ be the maximal power of a hypothesis test with significance $1 - \alpha$. The power of a threshold-test on $s_{part}$ with significance $1 - \alpha$ is $P_\alpha - \mathcal{O}(\epsilon)$.*

*Proof Sketch.* The expression to be minimized is shown to be the sum of two terms. One term is the sum of a subset of entries of $\Sigma^{-1}(X - \boldsymbol{\mu})$, which is minimized by simply taking the lowest entries (up to the constraint of consistency across episodes, which requires us to sum the rewards per time-step in advance). In Appendix E we bound the second term and its effects on the modified statistic and on the modified test-threshold. We show that the resulted decrease of rejection probability is $\mathcal{O}(\epsilon)$. □

## 5. Bootstrap for False Alarm Rate Control (BFAR)

For Setup 2, we suggest a sequential testing procedure: run an individual test every $d$ steps (i.e., $F = T/d$ test-points per episode), and return once any individual test declares a degradation. The tests can run according to Section 4, applied on the $h$ recent episodes. Multiple tests may be applied every test-point, e.g., with varying test-statistics $\{s\}$ or lookback-horizons $\{h\}$. This procedure, as implemented for the experiments of Section 6, is described in Fig. 3.

Setup 2 limits the probability of a false alarm to $\alpha_0$ in a run of $\tilde{h}$ episodes. To satisfy this condition, we set a uniform threshold $\kappa$ on the $p$-values of the individual tests (i.e., declare once a test returns $p$-val $< \kappa$). The threshold is determined using a Bootstrap mechanism for False Alarm control (**BFAR**, Algorithm 1).

While bootstrap methods for false alarm control are quite popular, they often rely on the data samples being i.i.d (Kharitonov et al., 2015; Abhishek & Mannor, 2017), which is crucial for the re-sampling to reliably mimic the source of the signal. To address the non-i.i.d signal, we take advantage of the episodic framework and sample whole episodes. We then use the re-sampled sequence to simulate tests on sub-sequences where the first and last episodes may be incomplete, as described below. This allows simulation of sequences of various lengths (including non-integer number of episodes) without assuming independence, normality, or identical distributions within the episodes.

**Algorithm 1:** BFAR: Bootstrap for FAR control

```
Input: reference dataset $x \in \mathbb{R}^{N \times T}$; statistic functions
$\{s\}$; lookback-horizons $\{h_1, ..., h_{max}\}$; test length
$\tilde{h} \in \mathbb{N}$; bootstrap repetitions $B \in \mathbb{N}$; desired
significance $\alpha_0 \in (0, 1)$;
Output: test threshold for individual tests;
Initialize $P = (1, ..., 1) \in [0, 1]^B$;
for $b$ in 1:$B$ do
    Initialize $Y \in \mathbb{R}^{(h_{max}+\tilde{h})T}$;
    for $k$ in 0:($h_{max}$+$\tilde{h}$-1) do
        Sample $j$ uniformly from $(1, ..., N)$;
        $Y[kT + 1 : kT + T] \leftarrow (x_{j1}, ..., x_{jT})$;
    for $t$ in test-points do
        for $h$ in lookback-horizons and $s$ in
          statistic functions do
            $y \leftarrow Y[t - hT : t]$;
            $p \leftarrow$ individual_test_pvalue($y$ vs. $x; s$)
            $P[b] \leftarrow \min(P[b], p)$;
Return $quantile_{\alpha_0}(P)$;
```

BFAR samples $h_{max}+\tilde{h}$ episodes (where $h_{max}$ is the maximal lookback-horizon) from reference data of $N$ episodes, to simulate sequential data $Y$. Then individual tests are simulated for any test-point along $\tilde{h}$ episodes, starting after $h_{max}$ episodes. The minimal $p$-value determines whether a detection would occur in $Y$. The whole procedure repeats $B$ times, creating a bootstrap estimate of the distribution of the minimal $p$-value along $\tilde{h}$ episodes. We choose the tests threshold to be the $\alpha_0$-quantile of this distribution, such that $\alpha_0$ of the bootstrap simulations would raise a false alarm.

Note that the statistic for the tests is given to BFAR as an input, making its choice independent of BFAR. BFAR can run in an offline manner (e.g., a single run before the deployment of the agent). It takes $\mathcal{O}(BF\tilde{h}\tilde{T})$ time, where $\tilde{T}$ is the time of a single update of all the test-statistics. Additional details are discussed in Appendices F,G.

# 6. Experiments

## 6.1. Methodology

We run experiments in standard RL environments as described below. For each environment, we train an agent using the PyTorch version (Kostrikov, 2018) of OpenAI's baseline (Dhariwal et al., 2017) of A2C algorithm (Mnih et al., 2016). We let the trained agent run in the environment for $N_0$ episodes and record its rewards, considered the *trusted reference data*. We then define several scenarios, and let the agent run for $M \times N$ episodes in each scenario (divided later into $M = 100$ blocks of $N$ episodes). One scenario is named $H_0$ and is identical to the reference up to the random initial-states. The other scenarios are defined per environment, and present environmental changes expected to harm the agent's rewards. The agent is *not* trained to adapt to these changes, and the goal is to test how long it takes for a degradation-test to detect the degradation.

Individual degradation-tests of length $n$ (Setup 1) are applied for every scenario over the first $n$ time-steps of each block. Sequential degradation-tests (Setup 2) are applied sequentially over the episodes of each block. Since the agent is assumed to run continuously as the environment changes from $H_0$ to an alternative scenario, each block is preceded by a random sample of $H_0$ episodes, as demonstrated in Fig. 3.

BFAR adjusts the tests thresholds to have a false alarm with probability $\alpha_0 = 5\%$ per $\tilde{h} = N$ episodes (where $N$ is the data-block size). Two lookback-horizons $h_1, h_2$ are chosen for every environment. The rewards are downsampled by a factor of $d$ before applying the tests, intended to reduce the parameters estimation error. Table 1 summarizes the setup of the various environments.

The experimented degradation-tests are a threshold-test on the simple **Mean**; **CUSUM** (Ryan, 2011); **Hotelling** (Hotelling, 1931); **UDT** and **PDT** (with $p = 0.9$) from Section 4; and a Mixed Degradation Test (**MDT**) that runs Mean, Hotelling and PDT in parallel – applying all three in every test-point (as permitted in Algorithm 1). All the degradation-tests are tuned according to the same reference data. Further implementation details are discussed in Appendix H.

## 6.2. Results

We run the tests in the environments of Pendulum (OpenAI), where the goal is to keep a pendulum pointing upwards; HalfCheetah (Todorov et al., 2012), where the goal

Table 1: Environments parameters
(episode length ($T$), reference episodes ($N_0$), test blocks ($M$), episodes per block ($N$), sequential test length ($\tilde{h}$), lookback horizons ($h_1, h_2$), tests per episode ($F = T/d$))

| Environment | $T$ | $N_0$ | $M$ | $N = \tilde{h}$ | $h_{1,2}$ | $F$ |
|---|---|---|---|---|---|---|
| Pendulum | 200 | 3e3 | 100 | 30 | 3,30 | 20 |
| HalfCheetah | 1000 | 1e4 | 100 | 50 | 5,50 | 40 |
| Humanoid | 200 | 5e3 | 100 | 30 | 3,30 | 10 |

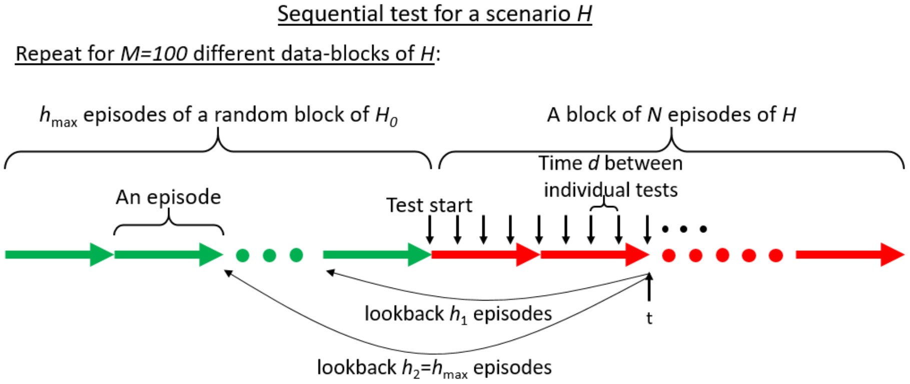

Sequential test for a scenario $H$
Repeat for $M$=100 different data-blocks of $H$:
$h_{\max}$ episodes of a random block of $H_0$
A block of $N$ episodes of $H$
An episode
Time $d$ between individual tests
Test start
lookback $h_1$ episodes
t
lookback $h_2$=$h_{\max}$ episodes

Figure 3: A summary of the sequential degradation-test procedure described in Section 6.1.

is for a 2D cheetah to run as fast as possible; and Humanoid, where the goal is for a person to walk without falling. In each environment we define the scenario *ccostx* of control cost increased to x% of its original value, as well as changed-dynamics scenarios specified in Appendix H.

In all the environments the rewards are clearly *not* independent, identically-distributed or normal (see Fig. 1 for example). Yet the false alarm rates are close to $\alpha_0 = 5\%$ per $\tilde{h}$ episodes in all the tests, as demonstrated in Fig. 4 (and in more details in Fig. 6 in Appendix I). These results under $H_0$ indicate that **BFAR tunes the thresholds properly** in spite of the complexity of the data. Note that BFAR never observed the data of scenario $H_0$ – only the reference data.

**In most of the non-$H_0$ scenarios, our tests prove to be more powerful than the standard tests, often by extreme margins**. For example, increased control cost in all the environments and additive noise in Pendulum are all 100%-detected by the suggested tests, usually within few episodes (Fig. 4); whereas Mean, CUSUM and Hotelling have very poor detection rates. Mean did not detect degradation in Pendulum even after the control cost increased from 110% to 300%(!), while keeping the significance level constant ($\alpha_0 = 5\%$).

Note that we run the tests with two lookback-horizons in parallel, as allowed by BFAR. This proves useful: with +30% control cost in HalfCheetah, for example, the short lookback-horizon allows fast detection of degradation; but with merely +10%, the long horizon is necessary to notice the slight degradation over a large number of episodes. This is demonstrated in Fig. 11 in Appendix I.

Covariance-based tests reduce the weights of the highly-varying (and presumably noisier) time-steps. In HalfCheetah they turn out to be in the later parts of the episode. As a result, in certain scenarios, Mean, CUSUM and Hotelling (which do not exploit the different variances optimally) do better in individual tests of 100 samples (out of $T = 1000$) than they do in one or even 10 full episodes (see Fig. 10a in Appendix I). This does not occur in UDT and PDT. Es-

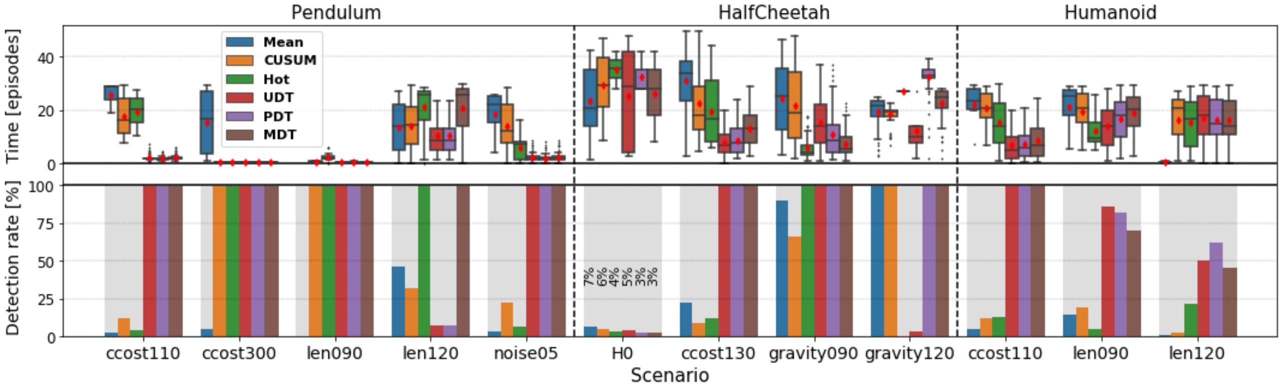


Figure 4: **Bottom**: percent of sequential tests that ended with degradation detection (high is good), over $M = 100$ runs with different seeds, for 3 standard tests and 3 variants of our test (**UDT**, **PDT** and **MDT**), in a sample of scenarios in Pendulum, HalfCheetah and Humanoid. **Top**: time until detection (low is good) – for the runs that ended with detection. The significance of the tests is shown for HalfCheetah in $H_0$ scenario (and for Pendulum and Humanoid as well in Fig. 6 in Appendix I).

sentially, we see that **ignoring the noise variability leads to violation of the principle that more data are better**.

In Pendulum, the ratio between variance of different steps may reach 5 orders of magnitude. This phenomenon increases the potential power of the covariance-based tests. For example, when the pole is shortened, negative changes in the highly-weighted time-steps are detected even when the mean of the whole signal increases. This feature allows us to detect slight changes in the environment before they develop into larger changes and cause damage.

On the other hand, a challenging situation arises when certain rewards decrease but the highly-weighted ones slightly increase (as in longer Pendulum's pole), which strongly violates the assumptions of Section 4. UDT is doomed to falter in such scenarios. PDT proves somewhat robust to this phenomenon since it is capable of focusing on a subset of time-steps, as demonstrated in increased gravity in HalfCheetah (Fig. 4). However, it cannot overcome the extreme weights differences in Pendulum. The one test that demonstrated robustness to all the experimented scenarios, including modified Pendulum's length and mass, is MDT. MDT combines Mean, Hotelling and PDT and does not fall far behind any of the three, in any of the scenarios. Hence, it presents excellent results in some scenarios and reasonable results in the others.

The tests were run on a single i9-10900X CPU core. BFAR (which needs to run only once and in an offline manner – before the deployment of the agent) took around 30 minutes per environment and test-statistic (several hours in total). Any parallelization should accelerate the bootstrap linearly with the number of cores. The sequential (online) tests themselves ran for 10 minutes per scenario – for all the 6 test-statistics together and for thousands of episodes.

Detailed experiments results are available in Appendix I. The code of the experiments is available on GitHub.

## 7. Related Work

**Training in non-stationary environments** has been widely researched, in particular in the frameworks of Multi-Armed Bandits (Mukherjee & Maillard, 2019; Garivier & Moulines, 2011; Besbes et al., 2014; Lykouris et al., 2020; Alatur et al., 2020; Gupta et al., 2019; Jun et al., 2018), model-based RL (Lecarpentier & Rachelson, 2019; Lee et al., 2020) and general multi-agent environments (Hernandez-Leal et al., 2019). Banerjee et al. (2016) explicitly detect changes in the environment and modify the policy accordingly, but assume that the environment is Markov, fully-observable, and its transition model is known – three assumptions that we avoid and that do not hold in many real-world problems. Safe exploration during training in RL was addressed by Garcia & Fernandez (2015); Chow et al. (2018); Junges et al. (2016); Cheng et al. (2019); Alshiekh (2017). Note that our work refers to changes beyond the scope of the training phase: it addresses the stage where the agent is fixed and required not to train further, in particular not in an online manner. Robust algorithms may prevent degradation in the first place, but when they fail – or when their assumptions are not met – an external model-free monitor with minimal assumptions (as the one suggested in this work) is crucial.

**Sequential tests** were addressed by many over the years. Common approaches rely on strong assumptions such as samples independence (Page, 1954; Ryan, 2011) and normality (Pocock, 1977; O'Brien & Fleming, 1979). Generalizations exist for certain private cases (Lu & Jr., 2001; Xie & Siegmund, 2011), sometimes at cost of alternative assumptions such as known change-size (Lund et al., 2007). Samples independence is usually assumed also in recent works with numeric approaches (Kharitonov et al., 2015; Abhishek & Mannor, 2017; Harel et al., 2014), and is often justified by consolidating many samples (e.g., an episode) together as a single sample (Colas et al., 2019). Ditzler et al. (2015) wrote that "change detection is typically carried out by inspecting i.i.d features extracted from the incoming data stream, e.g., the sample mean". Certain works address cyclic signals monitoring (Zhou et al., 2005), but to the best of our knowledge, we are the first to devise an optimal test for mean change in temporal non-i.i.d signals, and a false alarm control mechanism for such non-i.i.d signals.

Our work can be seen in part as converting a univariate temporal episodic signal into a $T$-dimensional multivariate signal. Many works addressed the problem of **change-point detection in multivariate variables**, e.g., using histograms comparison (Boracchi et al., 2018), Hotelling statistic (Hotelling, 1931), and K-L distance (Kuncheva, 2013). Hotelling in particular also looks for changed mean under unchanged covariance. However, unlike existing tests, we derive optimal tests for two different negative mean-change hypotheses, intended to detect degradation in temporal signals. Indeed, Section 6 demonstrates the advantage over Hotelling in such a context. In addition, by considering the temporal nature of the signal, we are able to handle "incomplete observations" and in particular obtain detections even within the middle of the first episode.

## 8. Summary

We introduced a novel approach that is optimal (under certain conditions) for detection of changes in episodic signals, exploiting the correlations structure as measured in a reference dataset. In environments of classic control (Pendulum) and MuJoCo (HalfCheetah, Humanoid), the suggested statistical tests detected degradation faster than alternatives, often by orders of magnitude. Certain conditions, such as combination of positive and negative changes in very heterogeneous signals, may cause instability in some of the suggested tests; however, this is shown to be solved by running the new test in parallel to standard tests – with only a small loss of test power.

We also introduced BFAR, a bootstrap mechanism that adjusts tests thresholds according to the desired false alarm rate in sequential tests. The mechanism empirically succeeded in providing valid thresholds for various tests in all the environments, in spite of the non-i.i.d data.

The suggested approach may contribute to development of reliable RL-based systems. Future research may consider different hypotheses, such as a permitted small degradation (instead of $H_0$) or a mix of degradation and improvement (instead of $H_A$); suggest additional stabilizing mechanisms for covariance-based tests; exploit other metrics than rewards for tests on model-based RL systems; and apply comparative tests of episodic signals beyond the scope of sequential change detection.

## Acknowledgements

This work was partially funded by the Israel Science Foundation under ISF grant number 2199/20. The authors wish to thank Guy Tennenholtz and Nadav Merlis for their helpful insights.

## References

Abhishek, V. and Mannor, S. A nonparametric sequential test for online randomized experiments. *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 610–6, 2017.

Alatur, P., Levy, K. Y., and Krause, A. Multi-player bandits: The adversarial case. *JMLR*, 2020.

Alshiekh, M. Safe reinforcement learning via shielding. *Logic in Computer Science*, 2017.

Alt, B., Sosic, A., and Koeppl, H. Correlation priors for reinforcement learning. *NeurIPS*, 2019.

Aminikhanghahi, S. and Cook, D. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51:339–367, 2016.

Badia, A. P. et al. Agent57: Outperforming the atari human benchmark. *ICML*, 2020.

Banerjee, T., Liu, M., and How, J. Quickest change detection approach to optimal control in markov decision processes with model changes, 09 2016.

Bellman, R. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957. ISSN 0022-2518.

Berry, D. A. and Fristedt, B. *Bandit problems*. Springer Netherlands, 1985. doi: 10.1007/978-94-015-3711-7.

Besbes, O., Gur, Y., and Zeevi, A. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in Neural Information Processing Systems (NIPS)*, 27, 2014.

Boracchi, G., Carrera, D., Cervellera, C., and Maccio, D. Quanttree: Histograms for change detection in multivariate data streams. *Proceedings of Machine Learning Research*, 80:639–648, 10–15 Jul 2018. URL `http://proceedings.mlr.press/v80/boracchi18a.html`.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Brook, D. et al. An approach to the probability distribution of cusum run length. *Biometrika*, 59(3):539–549, 1972.

Bylander, T. Lecture notes: Reinforcement learning. `http://www.cs.utsa.edu/~bylander/cs6243/reinforcement-learning.pdf`.

Chan, S. C. et al. Measuring the reliability of reinforcement learning algorithms. *ICLR*, 2020.

Chen, J. Conditional value at risk (cvar). `https://www.investopedia.com/terms/c/conditional_value_at_risk.asp`, 2020.

Cheng, R. et al. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. *AAAI Conference on Artificial Intelligence*, 2019.

Chow, Y. et al. A lyapunov-based approach to safe reinforcement learning. *NIPS*, 2018.

Colas, C., Sigaud, O., and Oudeyer, P.-Y. A hitchhiker's guide to statistical comparisons of reinforcement learning algorithms, 2019.

Dai, B., Ding, S., and Wahba, G. Multivariate bernoulli distribution. *Bernoulli*, 19(4):1465–1483, 09 2013. doi: 10.3150/12-BEJSP10. URL `https://doi.org/10.3150/12-BEJSP10`.

Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., and Zhokhov, P. Openai baselines. `https://github.com/openai/baselines`, 2017.

Dickey, D. A. and Fuller, W. A. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431, 1979. doi: 10.1080/01621459.1979.

10482531. URL https://doi.org/10.1080/01621459.1979.10482531.

Ditzler, G., Polikar, R., and Alippi, C. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 2015.

Dulac-Arnold, G., Mankowitz, D., and Hester, T. Challenges of real-world reinforcement learning, 2019.

Efron, B. Second thoughts on the bootstrap. *Statist. Sci.*, 18(2):135–140, 05 2003. doi: 10.1214/ss/1063994968. URL https://doi.org/10.1214/ss/1063994968.

Freedman, A. Convergence theorem for finite markov chains, 2017. URL https://math.uchicago.edu/~may/REU2017/REUPapers/Freedman.pdf.

Garcia, J. and Fernandez, F. A comprehensive survey on safe reinforcement learning. *JMLR*, 2015.

Garivier, A. and Moulines, E. On upper-confidence bound policies for switching bandit problems. *International Conference on Algorithmic Learning Theory*, pp. 174–188, 10 2011. doi: 10.1007/978-3-642-24412-4_16.

Goldman, M. Lecture notes in stat c141: The bonferroni correction. https://www.stat.berkeley.edu/ mgoldman/Section0402.pdf, 2008.

Gupta, A., Koren, T., and Talwar, K. Better algorithms for stochastic bandits with adversarial corruptions. *Proceedings of Machine Learning Research*, 2019.

Harel, M., Crammer, K., El-Yaniv, R., and Mannor, S. Concept drift detection through resampling. *International Conference on Machine Learning*, pp. II–1009–II–1017, 2014.

Henderson, P. et al. Deep reinforcement learning that matters. *AAAI*, 2017.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. A survey of learning in multiagent environments: Dealing with non-stationarity, 2019.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. *AAAI*, 2018.

Hotelling, H. The generalization of student's ratio. *Ann. Math. Statist.*, 2(3):360–378, 08 1931. doi: 10.1214/aoms/1177732979. URL https://doi.org/10.1214/aoms/1177732979.

Irwin, M. E. Lecture notes: Convergence in distribution and central limit theorem. http://www2.stat.duke.edu/~sayan/230/2017/Section53.pdf, 2006.

Jun, K.-S. et al. Adversarial attacks on stochastic bandits. *NeurIPS*, 2018.

Junges, S. et al. Safety-constrained reinforcement learning for mdps. *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2016.

K. V. Mardia, J. T. K. and Bibby, J. M. *Multivariate analysis*. Academic Press, 1979.

Kharitonov, E., Vorobev, A., Macdonald, C., Serdyukov, P., and Ounis, I. Sequential testing for early stopping of online experiments. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 473–482, 2015. doi: 10.1145/2766462.2767729. URL https://doi.org/10.1145/2766462.2767729.

Korenkevych, D., Mahmood, A. R., Vasan, G., and Bergstra, J. Autoregressive policies for continuous control deep reinforcement learning, 2019.

Kostrikov, I. Pytorch implementations of reinforcement learning algorithms. https://github.com/ikostrikov/pytorch-a2c-ppo-acktr-gail, 2018.

Kroese, D. P., Brereton, T., Taimre, T., and Botev, Z. Why the monte carlo method is so important today. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6: 386–392, 2014.

Kuncheva, L. I. Change detection in streaming multivariate data using likelihood detectors. *IEEE Transactions on Knowledge and Data Engineering*, 25(5):1175–1180, 2013. doi: 10.1109/TKDE.2011.226.

Lan, D. L. D. K. K. G. Interim analysis: The alpha spending function approach. *Statistics in Medicine*, 13:1341–52, 1994.

Lecarpentier, E. and Rachelson, E. Non-stationary markov decision processes: a worst-case approach using model-based reinforcement learning. *NeurIPS 2019*, abs/1904.10090, 2019. URL http://arxiv.org/abs/1904.10090.

Lee, K. et al. Context-aware dynamics model for generalization in model-based rl. *ICML*, 2020.

Lu, C.-W. and Jr., M. R. R. Cusum charts for monitoring an autocorrelated process. *Journal of Quality Technology*, 33(3):316–334, 2001. doi: 10.1080/00224065.2001.11980082. URL https://doi.org/10.1080/00224065.2001.11980082.

Lund, R., Wang, X. L., Lu, Q. Q., Reeves, J., Gallagher, C., and Feng, Y. Changepoint Detection in Periodic and Autocorrelated Time Series. *Journal of Climate*, 20(20): 5178–5190, 10 2007. ISSN 0894-8755. doi: 10.1175/JCLI4291.1. URL https://doi.org/10.1175/JCLI4291.1.

Lykouris, T., Mirrokni, V., and Leme, R. P. Bandits with adversarial scaling. *ICML*, 2020.

MathWorks. Conditional value-at-risk (cvar). https://www.mathworks.com/discovery/conditional-value-at-risk.html.

Matsushima, T., Furuta, H., Matsuo, Y., Nachum, O., and Gu, S. Deployment-efficient reinforcement learning via model-based offline optimization. *ArXiv*, abs/2006.03647, 2020.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. *Proceedings of Machine Learning Research*, 48:1928–1937, 20-22 Jun 2016.

MuJoCo. Halfcheetah-v2. https://gym.openai.com/envs/HalfCheetah-v2/.

Mukherjee, S. and Maillard, O.-A. Distribution-dependent and time-uniform bounds for piecewise i.i.d bandits. *arXiv preprint arXiv:1905.13159*, 2019.

Murphy, S. A., van der Laan, M. J., and Robins, J. M. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 2001.

Nachum, O., Ahn, M., Ponte, H., Gu, S. S., and Kumar, V. Multi-agent manipulation via locomotion using hierarchical sim2real. *PMLR*, 100:110–121, 30 Oct–01 Nov 2020. URL http://proceedings.mlr.press/v100/nachum20a.html.

NCSS. Cumulative sum (cusum) charts. https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/CUSUM_Charts.pdf.

Neyman, J., Pearson, E. S., and Pearson, K. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London*, 1933. doi: 10.1098/rsta.1933.0009.

O'Brien, P. C. and Fleming, T. R. A multiple testing procedure for clinical trials. *Biometrics*, 35(3):549–556, 1979. ISSN 0006341X, 15410420. URL http://www.jstor.org/stable/2530245.

OpenAI. Pendulum-v0. https://gym.openai.com/envs/Pendulum-v0/.

Page, E. S. Continuous Inspection Schemes. *Biometrika*, 41(1-2):100–115, 06 1954. ISSN 0006-3444. doi: 10.1093/biomet/41.1-2.100. URL https://doi.org/10.1093/biomet/41.1-2.100.

Pardo, F., Tavakoli, A., Levdik, V., and Kormushev, P. Time limits in reinforcement learning. *CoRR*, abs/1712.00378, 2017. URL http://arxiv.org/abs/1712.00378.

PennState College of Science. Lecture notes in stat 509: Alpha spending function approach. https://online.stat.psu.edu/stat509/node/81/.

Petrov, V. V. *Sums of Independent Random Variables*. Nauka, 1972.

Pocock, S. J. Group sequential methods in the design and analysis of clinical trials. *Biometrika*, 64(2):191–199, 08 1977. ISSN 0006-3444. doi: 10.1093/biomet/64.2.191. URL https://doi.org/10.1093/biomet/64.2.191.

Rockafellar, R. T. and Uryasev, S. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000. doi: 10.21314/JOR.2000.038.

Ryan, T. P. *Statistical Methods for Quality Improvement*. Wiley; 3rd Edition, 2011.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Wald, A. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics*, 16(2):117–186, 06 1945. doi: 10.1214/aoms/1177731118. URL https://doi.org/10.1214/aoms/1177731118.

Westgard, J., Groth, T., Aronsson, T., and Verdier, C. Combined shewhart-cusum control chart for improved quality control in clinical chemistry. *Clinical chemistry*, 23: 1881–7, 11 1977. doi: 10.1093/clinchem/23.10.1881.

Wilks, S. S. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Ann. Math. Statist.*, 9(1):60–62, 03 1938. doi: 10.1214/aoms/1177732360. URL https://doi.org/10.1214/aoms/1177732360.

Williams, S. M. et al. Quality control: an application of the cusum. *BMJ: British medical journal*, 304.6838:1359, 1992.

Xie, Y. and Siegmund, D. Weak change-point detection using temporal correlation, 2011.

Yashchin, E. On the analysis and design of cusum-shewhart control schemes. *IBM Journal of Research and Development*, 29(4):377–391, 1985.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization, 2020.

Zhao, X. et al. Assessing the safety and reliability of autonomous vehicles from road testing. *ISSRE*, 2019.

Zhou, S., Jin, N., and Jin, J. J. Cycle-based signal monitoring using a directionally variant multivariate control chart system. *IIE Transactions*, 37(11):971–982, 2005. doi: 10.1080/07408170590925553. URL `https://doi.org/10.1080/07408170590925553`.