Online Change-point Detection for Cooperative Multi-Agent Reinforcement Learning

Fatemeh Saberi Khomami[0009−0007−0823−3887] and Julita Vassileva[0000−0001−5050−3106]

Department of Computer Science, University of Saskatchewan, Saskatoon, Canada
f.saberi@usask.ca, jiv@cs.usask.ca

Abstract. Cooperative multi-agent reinforcement learning (MARL) systems rely on past experience for learning coordinated behaviour, but this experience may become unreliable if the environment or task objective changes during training. In such cases, agents first need a way to recognize that the situation has changed before deciding how to adapt. This paper studies online change-point detection for cooperative MARL using reward-derived signals. We propose *Patterns of Past Rewards* (PPR), a lightweight algorithm-agnostic detector that smooths agents' return streams, highlights recent changes, and applies a statistical drift detector to flag significant shifts. We evaluate PPR in a custom Speaker–Listener environment based on the Multi-Agent Particle Environment (MPE) under two controlled non-stationarity scenarios. Our results show a trade-off between detection speed and alarm stability. A smoothed-return baseline detects earlier but produces many repeated alarms. In contrast, applying the detector directly to raw returns often misses the shift. PPR offers a more balanced approach by limiting redundant detections while still identifying the controlled shifts. These findings suggest that PPR can serve as a lightweight, reward-based monitoring tool for identifying controlled task changes during cooperative MARL training.

Keywords: Multi-Agent Reinforcement Learning · Non-stationarity · Change-point Detection · Multi-Agent Systems

1 Introduction

Multi-agent reinforcement learning (MARL) has become an important approach for cooperative tasks in which multiple agents must coordinate over time, such as warehouse logistics, traffic management, and collaborative robotics [1, 9]. In these systems, agents typically acquire policies by repeatedly interacting with an environment and with each other, gradually shaping behaviour through accumulated experience [2]. This reliance on past experience is acceptable as long as the learning circumstances remain compatible with the conditions seen during training. However, real-world multi-agent systems are rarely stationary. Observation channels can change, reward functions can be updated, and the composition or roles of agents may vary over time [8]. When such shifts (non-stationarities) occur, previously effective policies can become unreliable.

This version of the contribution has been accepted for publication, after peer review (when applicable) but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-032-41794-7_24. Use of this Accepted Version is subject to the publisher's Accepted Manuscript terms of use https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms

This paper is motivated by a fundamental question: *when should a learning system stop trusting what it has learned?* In cooperative MARL, this question matters because agents depend on past observations, coordination patterns, and reward histories, which may become invalid after a shift [14]. If the system continues to trust stale experience for too long, learning may become inefficient or misleading. If it reacts too aggressively, it may discard useful knowledge and destabilize coordination. The challenge is to detect when the environment has changed enough so that prior experience should be treated with caution, rather than treating non-stationarity as a single abstract property of the process.

In MARL and non-stationary reinforcement learning (RL), shifts are usually addressed through adaptation, such as continued training, exploration adjustment, or other forms of policy recovery [18,19]. These approaches are useful, but they focus primarily on online adaptation mechanisms, rather than treating the detection of non-stationarity itself as a central objective. In practice, detection must come first because without an online signal of change, an agent has no clear reason to adjust its trust in prior knowledge. Thus, change-point detection is a foundational step.

To address this problem, we build on the idea that reward-derived signals implicitly encode information about environment dynamics, task objectives, and agent coordination quality, and if a change regarding any of these aspects occurs, it should be reflected in the reward signal. We designed *Patterns of Past Rewards* (PPR), a lightweight, algorithm-agnostic detector that operates on windows of episodic returns without modifying the underlying MARL algorithm. We evaluate this approach in a custom Speaker–Listener environment derived from the Multi-Agent Particle Environment (MPE) [11, 12], where we introduce controlled shifts during training. The experiments are designed to assess whether online change-point detection is feasible in these controlled cooperative settings. The paper's contribution is a step toward trustworthy multi-agent systems under objective and environmental shifts, in which systems first detect such shifts and then decide how to adapt.

The remainder of the paper is organized as follows. Section 2 reviews cooperative MARL, non-stationarity, and online change-point detection. Section 3 introduces the PPR pipeline. Section 4 describes the experimental setup and controlled shift scenarios. Section 5 presents the results and ablation comparison. Finally, Section 6 summarizes the conclusions of the study.

2 Background

2.1 Cooperative MARL under Partial Observability

MARL adds flexibility to RL by introducing a multi-agent setting in which agents interact in a shared environment over time to adapt to a more complex goal. In fully cooperative settings with partial observability, the problem is often modelled as a decentralized partially observable Markov decision process (Dec-POMDP) [4]. This formalism captures the key properties of our setting: multiple agents, local observations, and a shared reward signal.

A Dec-POMDP is defined by the tuple $(\mathcal{I}, \mathcal{S}, \mathcal{A}, T, \mathcal{R}, \mathcal{Z}, \mathcal{O}, \gamma)$, where $\mathcal{I} = \{1, \ldots, N\}$ is the set of agents, $\mathcal{S}$ is the state space, $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the joint action space, $T(s' \mid s, \mathbf{a})$ is the transition function, giving the probability of reaching state s' when the joint action $\mathbf{a}$ is taken in state s, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the shared reward function, $\mathcal{Z}$ is the joint observation space, $\mathcal{O}(\mathbf{z} \mid s', \mathbf{a})$ is the observation function, giving the probability that the agents receive the joint observation $\mathbf{z}$ after the joint action $\mathbf{a}$ leads to state s', and $\gamma \in [0, 1)$ is the discount factor [4]. At each discrete time step $t = 0, 1, 2, \ldots$, the environment is in a state $s_t \in \mathcal{S}$. Each agent $i \in \mathcal{I}$ receives a local observation z_t^i, selects an action $a_t^i \in \mathcal{A}_i$ according to its policy, and the joint action $\mathbf{a}_t$ determines the next state through T. The agents then receive a shared reward $r_t = \mathcal{R}(s_t, \mathbf{a}_t)$.

The objective in cooperative MARL is to learn policies that maximize the expected discounted team return,

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{1}$$

Unlike single-agent RL, each agent acts based only on its local observation history rather than the full environment state. As a result, performance depends not only on the quality of individual decisions, but also on coordination among agents over time [2, 14]. In this work, we instantiate our framework in a two-agent cooperative communication task derived from MPE, where agents must coordinate through messages and actions to achieve a shared goal [11, 12].

2.2 Non-stationarity in RL and MARL

Many RL algorithms assume that the transition dynamics and reward function do not change over time. In such stationary environments, experience from the past can be a representative of future interactions [8, 15]. In practice, this assumption often fails because changes in the environment's behaviour, reward structure, or observation function can alter the distribution of trajectories an agent experiences over time.

Numerous studies have introduced approaches for single-agent RL in non-stationary environments. Hamadanian et al. considered environments whose behaviour depends on a time-varying latent context and studied how to update policies online while limiting forgetting of behaviour that remains useful across contexts [7].

In multi-agent settings, non-stationarity arises from both changes in the environment and other agents simultaneously learning. For any individual agent, changes in teammates' or opponents' policies can make the effective transition dynamics non-stationary, even if the environment itself remains unchanged [14]. Papoudakis et al. surveyed this phenomenon in deep MARL and reviewed techniques such as centralized critics, opponent modelling, and experience replay modifications that aim to reduce instability due to changing co-players. More recent work proposed multi-timescale learning rules to better track non-stationary behaviour in decentralized cooperative MARL [13], and fast adaptation schemes for sudden policy changes in teammates [19].

2.3 Change-point Detection

Change-point detection seeks to identify when the distribution generating a sequence of observations has changed. In RL, this is important because shifts in dynamics or rewards can make earlier experience less reliable [5, 6, 10]. In cooperative MARL, the problem is amplified because one agent's policy change can alter the observations and rewards seen by others [2, 14].

Classical likelihood-based sequential methods, such as Page's CUSUM and Shiryaev-type tests, provided a strong foundation for quick change detection [3, 6]. These methods are effective when a likelihood model is available or can be approximated, but they are less directly suited to model-free RL monitoring, where the signal is often a reward trace or another summary statistic rather than a fully specified transition model [5,10]. For this reason, model-free detectors have received increasing attention in RL. Canonaco et al. combined a statistical hypothesis test with an importance-sampling transformation and a CUSUM-style upper layer to detect non-stationarity online, then triggered adaptation by resetting optimizer state [5]. More recent work also studied change-point detection in offline RL by testing the stationarity of the optimal Q-function and using the detected change point to select the data segment for subsequent learning [10]. These studies support the broader view that detection and adaptation are distinct steps: the first determines whether a change has occurred, and the second decides how to respond.

In streaming settings, a practical alternative is sliding-window distribution testing. KSWIN follows this idea by comparing a reference window and a recent window with a Kolmogorov–Smirnov (KS) test, and it is implemented in the River library as an online concept-drift detector with tunable window and significance parameters [16]. This makes it suitable for reward-stream monitoring in RL, where the signal is available online but is often noisy. In our setting, change-point detection serves as a diagnostic step that indicates when past experience should be treated with caution before any later adaptation mechanism is applied.

Reward-based monitoring has also been used in non-stationary RL and traffic-control settings. Salkham and Cahill used moving-average rewards as part of an online change-detection mechanism in fluctuating traffic control [17]. The key limitation of reward-based monitoring is that reward-derived sequences are often noisy, which can reduce detector stability and delay the identification of meaningful shifts.

3 Our Approach

In this section, we describe *Patterns of Past Rewards* (PPR), a lightweight, algorithm-agnostic online detector that monitors the episodic return stream during cooperative MARL training. The episodic return is a reward-derived signal computed from the rewards collected within an episode. PPR does not modify the underlying learning algorithm or its update rules; instead, it reads the return signal produced during training, transforms observed episode returns, and

applies statistical drift detection to identify when recent return patterns differ from previous patterns.

Let R_e denote the return observed at the end of episode e. The simple moving average (SMA) stage maps R_e to $\bar{R}_e$, the exponential moving variation (EMV) stage maps $\bar{R}_e$ to V_e, and KSWIN maps V_e to the binary drift indicator $d_e \in \{0, 1\}$, which indicates whether drift is detected after episode e. The overall pipeline is

$$R_e \xrightarrow{\text{SMA}} \bar{R}_e \xrightarrow{\text{EMV}} V_e \xrightarrow{\text{KSWIN}} d_e. \tag{2}$$

The SMA stage reduces short-term fluctuations in the return stream, while the EMV stage emphasizes changes in the smoothed return trajectory. KSWIN then tests whether the recent transformed return distribution differs from the distribution in the previous window.

Simple Moving Average (SMA). The episodic return stream can contain high-frequency fluctuations caused by exploration noise, stochastic transitions, and ordinary learning dynamics. Applying a drift detector directly to this raw sequence can therefore produce unstable alarms that reflect transient variation rather than a persistent change in the training regime. PPR first applies SMA to obtain a local trend estimate of the recent return pattern. Given a window size w, PPR keeps a sliding buffer of the w most recent episodic returns, and the smoothed return at episode e is

$$\bar{R}_e = \frac{1}{w} \sum_{i=e-w+1}^{e} R_i. \tag{3}$$

In practice, the moving average can only be computed once at least w episode returns have been observed, so the detector begins after an initial warm-up period.

Exponential Moving Variation (EMV). After smoothing the episodic return stream, PPR applies a second transformation before statistical drift detection is performed. PPR computes the EMV signal as

$$V_e = \beta(\bar{R}_e - \bar{R}_{e-1})^2 + (1 - \beta)V_{e-1}, \tag{4}$$

where V_e is the EMV value at episode e, and $\beta \in [0, 1]$ controls how strongly recent changes influence the updated variation signal. Larger values of β make V_e track the most recent change more closely, while smaller values retain past variation for longer.

This transformation converts changes in the smoothed return sequence into a non-negative variation signal. Gradual changes in $\bar{R}_e$ produce relatively small EMV values, whereas abrupt changes or bursts of variability produce larger values, making sudden changes in return dynamics more distinguishable before the signal is passed to KSWIN.

KSWIN drift test. The final stage applies KSWIN [16], a Kolmogorov–Smirnov sliding-window detector, to the transformed values V_e. KSWIN maintains a sliding window holding the $\mathcal{W}$ most recent values $V_{e-\mathcal{W}+1}, \ldots, V_e$, where $\mathcal{W}$ is the window size. The test sample $B = (V_{e-m+1}, \ldots, V_e)$ contains the m newest values in the window, and the reference sample A consists of m values drawn uniformly at random from the $\mathcal{W} - m$ earlier values in the window [16]. While the return pattern is stable, A and B follow the same distribution, so a significant difference between them indicates drift. KSWIN compares their empirical cumulative distribution functions F_A and F_B, where $F_X(x) = \frac{1}{m} \sum_{v \in X} \mathbf{1}\{v \leq x\}$ is the fraction of values in sample X that are less than or equal to x, using the two-sample Kolmogorov–Smirnov statistic:

$$D = \sup_x |F_A(x) - F_B(x)| . \tag{5}$$

This is the largest vertical distance between the two empirical distribution functions. Following the KSWIN implementation [16], the critical threshold is:

$$\tau_\alpha = \sqrt{\frac{-\ln(\alpha)}{m}}. \tag{6}$$

Here, α is the significance parameter; this threshold approximates the standard two-sample KS critical value at level α. A drift is flagged after episode e when $D > \tau_\alpha$.

4 Experiment Setup

4.1 Speaker–Listener Task

We evaluate PPR in a custom cooperative Speaker–Listener task derived from MPE with two agents and three landmarks. The agents have complementary roles: a speaker that observes the goal colour and communicates a message vector, and a listener that observes the landmark positions and the speaker's message and chooses movement actions. Both agents receive the same team reward, computed after every step as the negative squared Euclidean distance between the listener and the goal landmark, so learning is driven by shared task performance rather than by separate local objectives.

At the beginning of each episode, the positions of both agents and all three landmarks are independently resampled uniformly from $[-1, 1]^2$, and one landmark is randomly selected as the goal. The speaker outputs a learned continuous message vector $\mathbf{m} \in [0, 1]^3$, which is appended to the listener's observation without noise or communication cost. The listener's action is a five-dimensional continuous vector $[a_0, a_L, a_R, a_D, a_U] \in [0, 1]^5$, corresponding to no movement, left, right, down, and up.

The agents are not given an explicit symbolic mapping from colours to landmarks. Instead, they learn a communication convention through reward feedback, with the speaker learning to encode the goal colour in its message and the listener learning to interpret the message and move toward the corresponding landmark.

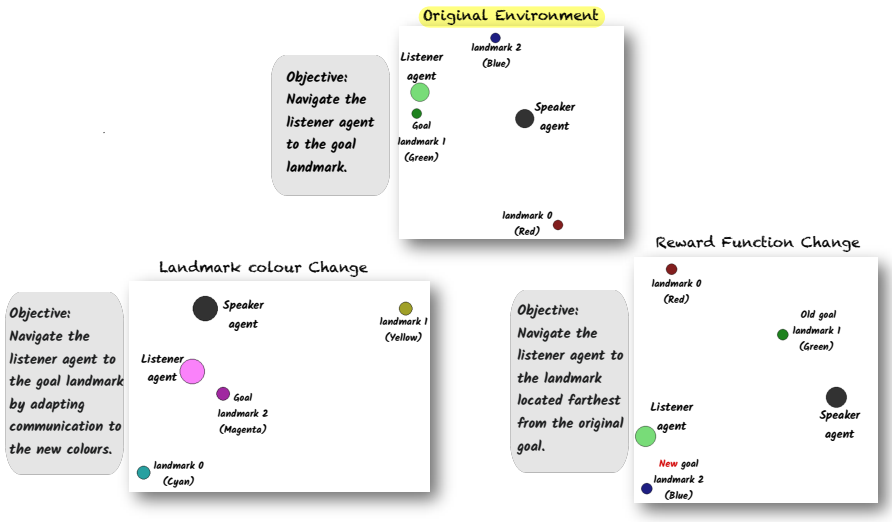


Fig. 1. Controlled shifts in the custom Speaker–Listener environment. The landmark-colour-change scenario reassigns landmark colours, while the reward-function-change scenario makes the landmark farthest from the original goal the new target.

4.2 Controlled Non-stationarity Scenarios

To study online change-point detection under controlled non-stationarity, we introduce two types of shifts during training. In the first scenario, the landmark colour assignment is changed during the course of training. For example, we reassign the landmarks from red, green, and blue to cyan, magenta, and yellow. The target is still defined through colour, but the colour-to-landmark association changes. As a result, the previously learned communication convention may no longer produce the correct coordination pattern, and the speaker and listener must re-establish a compatible mapping through further interaction.

In the second scenario, we change the reward definition while keeping the cooperative setting intact. After the shift, the rewarded target becomes the landmark farthest from the original goal's location, and the team reward is recomputed with respect to this new target. Therefore, the agents continue to solve a collaborative task, but the communication convention that was useful before the shift may no longer match the new reward structure. Figure 1 illustrates the original task and the two controlled shift scenarios used in our experiments.

4.3 Training and Detection Protocol

We train the agents with the multi-agent deep deterministic policy gradient (MADDPG) algorithm [11] in an online setting. PPR runs alongside training and monitors the episodic return stream without modifying the MARL algorithm or its update rules. The detector receives the episode-level return after each episode,

applies its internal transformations, and records a detected change point when a shift is flagged.

Each episode contains 25 environment steps. Training is run for $1,000,000$ steps, with the controlled shift introduced at step $500,000$, corresponding to episode $20,000$. MADDPG uses actor/critic learning rates of 3×10^{-4} and 10^{-3}, hidden layers of size 128 and 64, batch size 1024, replay buffer size 10^6, and 5000 warm-up steps. To assess the contribution of each stage in PPR, we compare the full pipeline, SMA $\rightarrow$ EMV $\rightarrow$ KSWIN, against two ablations. The first ablation, SMA + KSWIN, removes the EMV stage and applies KSWIN directly to the smoothed episodic return. The second ablation, raw return + KSWIN, removes both preprocessing stages and applies KSWIN directly to the episodic return stream. For the final comparison, all detector variants use the same matched detector setting: $w = 8$, $\beta = 2/3$, $\alpha = 2 \times 10^{-4}$, $\mathcal{W} = 25$, and $m = 10$. The sensitivity of PPR to alternative detector settings is examined in Section 5.

4.4 Evaluation Metrics

We evaluate detection performance using three metrics that capture both timeliness and alarm stability. Detection delay is the number of episodes between the true shift point and the first detector flag after the shift. Pre-shift detections counts the number of flags that occur before the true shift point and serves as a measure of false early alarms in shift experiments. Excess detections is the number of detector flags that occur after the first correct post-shift detection and captures the repeated alarm burden after the change has already been identified. Together, these metrics distinguish detectors that react quickly but over-trigger from detectors that are more conservative but may respond later.

A run is counted as a missed detection if the detector does not flag any drift after the true shift point within the training horizon. In such cases, we report the number of seeds in which detection occurred and compute delay only over detected runs.

5 Results

We first examine the learning dynamics under the two controlled shifts and then evaluate how the different detectors respond to these changes. For each scenario and detector variant, detection performance is evaluated over three independent random seeds, with the specific seed sets reported in the corresponding table captions, and all detection results are reported as means and standard deviations over these runs. Figure 2 compares shifted runs with matched no-shift runs for both scenarios. The dashed vertical line marks the true shift point at episode 20,000. In the reward-function-change panel, the shifted curve separates from the no-shift curve shortly after the change point, indicating a visible effect on learning performance. In the landmark-colour-change panel, the difference is more subtle, suggesting that some shifts do not cause an immediate collapse in average episodic return.

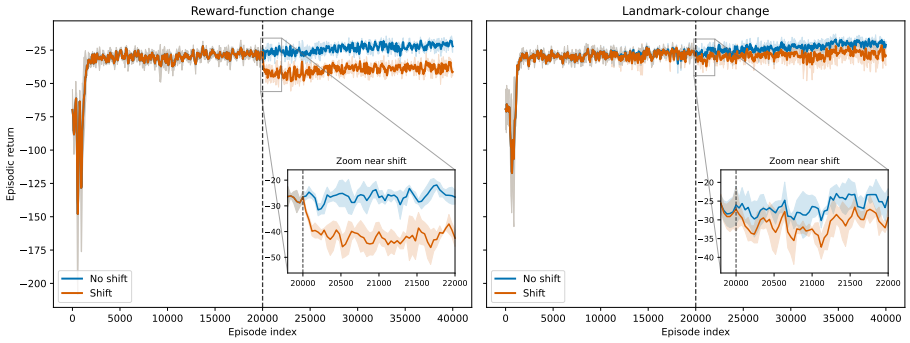


Fig. 2. Learning curves under controlled shifts compared with matched no-shift runs. The dashed vertical line marks the true shift point at episode 20,000.

Table 1. Detection results for the reward-function-change scenario. Lower values are better for all metrics. Detection delay is reported in episodes. Values are reported as mean ± sd over seeds 25–27.

Method	Detection delay	Pre-shift detections	Excess detections
PPR	250.7 ± 196.1	10.7 ± 4.6	11.0 ± 3.0
SMA + KSWIN	103.7 ± 76.7	298.3 ± 6.7	286.3 ± 17.6
Raw return + KSWIN	18424.0 (1/3 detected)	0.0 ± 0.0	0.0 ± 0.0

Table 2. Detection results for the landmark-colour-change scenario. Lower values are better for all metrics. Detection delay is reported in episodes. Values are reported as mean ± sd over seeds 22–24.

Method	Detection delay	Pre-shift detections	Excess detections
PPR	261.7 ± 56.6	8.3 ± 3.1	10.3 ± 1.5
SMA + KSWIN	52.0 ± 48.5	300.3 ± 14.3	295.7 ± 12.9
Raw return + KSWIN	4178.0 (1/3 detected)	0.3 ± 0.6	0.0 ± 0.0

Table 1 shows that SMA + KSWIN detects the reward-function change earlier than PPR, but it does so with hundreds of detections before and after the change, which makes the detector difficult to use in practice. Raw return + KSWIN is much more conservative. It produces almost no alarms, but it also fails to detect the shift reliably. Only one of the three runs triggered detection, and that detection occurred long after the true change point. PPR sits between these extremes, with slower detection than SMA + KSWIN but substantially fewer repeated alarms and better responsiveness than raw return alone. Detection delay under reward-function change also exhibits high variability across seeds, suggesting sensitivity to the underlying MARL training trajectory.

The same pattern appears in Table 2. SMA + KSWIN again detects earlier, but its alarm stream is highly unstable, with roughly 300 pre-shift and post-

Table 3. PPR parameter sensitivity for the landmark-colour-change scenario. KSWIN is fixed at $(\alpha, \mathcal{W}, m) = (2 \times 10^{-4}, 25, 10)$, and values are reported as mean ± sd over seeds 22–24.

Setting	w	β	Detection delay	Pre-shift detections	Excess detections
Final	**8**	**0.667**	$\mathbf{261.7 \pm 56.6}$	$\mathbf{8.3 \pm 3.1}$	$\mathbf{10.3 \pm 1.5}$
Smaller w	5	0.667	674.0 ± 286.6	21.3 ± 1.5	24.7 ± 2.5
Larger w	10	0.667	1100.3 ± 697.1	8.3 ± 2.5	7.3 ± 3.1
Smaller β	8	0.500	141.7 ± 86.7	44.3 ± 4.7	43.0 ± 2.6
Larger β	8	1.000	No detection (0/3)	0.0 ± 0.0	0.0 ± 0.0

shift detections. Raw return + KSWIN remains conservative and again misses most shifts. PPR does not minimize delay, yet it provides a more usable balance between timeliness and stability, especially when the observed return changes only mildly after the shift.

We further examine the effect of changing parameters w and β around the final detector setting. Table 3 shows that reducing β gives earlier detection but substantially increases pre-shift detections and repeated alarms, while increasing w reduces alarm burden but increases delay. The final configuration was selected as a balanced setting between timely detection and alarm stability.

Across both scenarios, the results indicate a clear sensitivity–stability trade-off. A detector that reacts very quickly may also over-trigger and produce many redundant alarms, whereas a very conservative detector may avoid false alarms but miss the shift altogether. Overall, PPR offers a more balanced operating point for online monitoring in cooperative MARL, where useful change-point detection must be both sufficiently sensitive and stable.

6 Conclusion

This paper examines online change-point detection in cooperative MARL, where shifts in reward structure or coordination requirements can make prior experience unreliable. To address this problem, we propose PPR, a lightweight detector that monitors episodic returns through a sequence of SMA and EMV transformations, followed by a KSWIN drift test. Our method is algorithm-agnostic and can be applied alongside any existing MARL training procedures without modifying the underlying learner.

Our experiments in a custom Speaker–Listener environment investigate two types of shifts: a reward-function change and a landmark-colour change. Across both scenarios, the results show a clear sensitivity–stability trade-off. SMA + KSWIN detects changes earlier but produces a large number of pre-shift and post-shift detections, making its alarm stream difficult to use in practice. Raw return + KSWIN is much more conservative, but often fails to detect the shift within the training horizon. PPR offers a more balanced outcome by reducing repeated alarms while remaining more responsive than raw return monitoring.

These results support the view that reward-derived signals can provide useful diagnostic information about non-stationarity in cooperative MARL. At the same time, the variation in detection delay across independent runs, especially in the reward-function-change scenario, shows that the same controlled shift can appear differently in the return stream depending on the stochastic MARL training trajectory. This indicates a limitation of reward-based monitoring. Because the detector observes only return-derived signals, its response can depend on how clearly the shift is reflected in the learned behaviour and episodic returns.

This study is limited to a controlled two-agent Speaker–Listener environment and two types of induced shifts. Therefore, the quantitative results should be interpreted as an initial empirical study. In addition, our comparisons focus on ablations of the PPR pipeline rather than on a broad set of external change-point detection baselines. Moreover, because PPR observes only reward-derived signals, a drift flag indicates a change in the return pattern but does not identify whether the source is the environment, reward function, teammate behaviour, coordination failure, or ordinary learning instability. Evaluating PPR beyond this setting remains necessary to assess its broader generality.

Future work will investigate adaptive detector parameter selection, broader monitoring signals beyond episodic returns, evaluation on more complex cooperative MARL benchmarks, alternative MARL algorithms, additional forms of non-stationarity such as teammate-policy or role changes, and comparisons with additional online change-point detection baselines, and the integration of PPR with downstream adaptation and continual-learning mechanisms, so that detected shifts can trigger appropriate recovery strategies.

Acknowledgments. This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Discovery Grant held by the second author (RGPIN-2021-03521).

Declaration of generative AI in scientific writing

During proofreading of this article, ChatGPT (OpenAI) was used to assist in identifying spelling, grammar, and clarity issues. All manuscript text was written and thoroughly reviewed by the authors, who take full responsibility for its content.

References

1. Abdoos, M., Mozayani, N., Bazzan, A.L.C.: Traffic light control in non-stationary environments based on multi agent Q-learning. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 1580–1585 (2011)
2. Albrecht, S.V., Stone, P.: Autonomous agents modelling other agents: A comprehensive survey and open problems. Artificial Intelligence **258**, 66–95 (2018)
3. Banerjee, T., Liu, M., How, J.P.: Quickest change detection approach to optimal control in Markov decision processes with model changes. In: 2017 American control conference (ACC). pp. 399–405. IEEE (2017)
4. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of Markov decision processes. Mathematics of Operations Research **27**(4), 819–840 (2002)

5. Canonaco, G., Restelli, M., Roveri, M.: Model-free non-stationarity detection and adaptation in reinforcement learning. In: ECAI 2020: 24th European Conference on Artificial Intelligence. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 1047–1054. IOS Press (2020)
6. Hadoux, E., Beynier, A., Weng, P.: Sequential decision-making under non-stationary environments via sequential change-point detection. In: Learning over multiple contexts (LMCE) (2014)
7. Hamadanian, P., Nasr-Esfahany, A., Schwarzkopf, M., Sen, S., Alizadeh, M.: Online reinforcement learning in non-stationary context-driven environments. In: International Conference on Learning Representations (2025)
8. Khetarpal, K., Riemer, M., Rish, I., Precup, D.: Towards continual reinforcement learning: A review and perspectives. Journal of Artificial Intelligence Research **75**, 1401–1476 (2022)
9. Krnjaic, A., Steleac, R.D., Thomas, J.D., Papoudakis, G., Schäfer, L., To, A.W.K., Lao, K., Cubuktepe, M., Haley, M., Börsting, P., Albrecht, S.V.: Scalable multi-agent reinforcement learning for warehouse logistics with robotic and human co-workers. In: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 677–684. IEEE (2024)
10. Li, M., Shi, C., Wu, Z., Fryzlewicz, P.: Testing stationarity and change point detection in reinforcement learning. The Annals of Statistics **53**(3), 1230–1256 (2025)
11. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 6382–6393 (2017)
12. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. In: AAAI Conference on Artificial Intelligence (2017)
13. Nekoei, H., Badrinaaraayanan, A., Sinha, A., Amini, M., Rajendran, J., Mahajan, A., Chandar, S.: Dealing with non-stationarity in decentralized cooperative multi-agent deep reinforcement learning via multi-timescale learning. In: Proceedings of The 2nd Conference on Lifelong Learning Agents. Proceedings of Machine Learning Research, vol. 232, pp. 376–398. PMLR (2023)
14. Papoudakis, G., Christianos, F., Rahman, A., Albrecht, S.V.: Dealing with non-stationarity in multi-agent deep reinforcement learning (2019), `https://arxiv.org/abs/1906.04737`
15. Peng, B., Papadimitriou, C.: The complexity of non-stationary reinforcement learning. In: Proceedings of The 35th International Conference on Algorithmic Learning Theory. Proceedings of Machine Learning Research, vol. 237, pp. 972–996. PMLR (2024)
16. Raab, C., Heusinger, M., Schleif, F.M.: Reactive soft prototype computing for concept drift streams. Neurocomputing **416**, 340–351 (2020)
17. Salkham, A.a., Cahill, V.: Soilse: A decentralized approach to optimization of fluctuating urban traffic using reinforcement learning. In: 13th international IEEE conference on intelligent transportation systems. pp. 531–538. IEEE (2010)
18. Steinparz, C., Schmied, T., Paischer, F., Dinu, M., Patil, V., Bitto-Nemling, A., Eghbal-zadeh, H., Hochreiter, S.: Reactive exploration to cope with non-stationarity in lifelong reinforcement learning. In: 1st Conference on Lifelong Learning Agents (2022)
19. Zhang, Z., Yuan, L., Li, L., Xue, K., Jia, C., Guan, C., Qian, C., Yu, Y.: Fast teammate adaptation in the presence of sudden policy change. In: Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence. Proceedings of Machine Learning Research, vol. 216, pp. 2465–2476 (2023)