24th International Conference on Practical Applications of Agents and Multi-Agent Systems

Naples, Italy · 21–23 October 2026

PAAMS International Conference

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

Fatemeh Saberi Khomami · Julita Vassileva

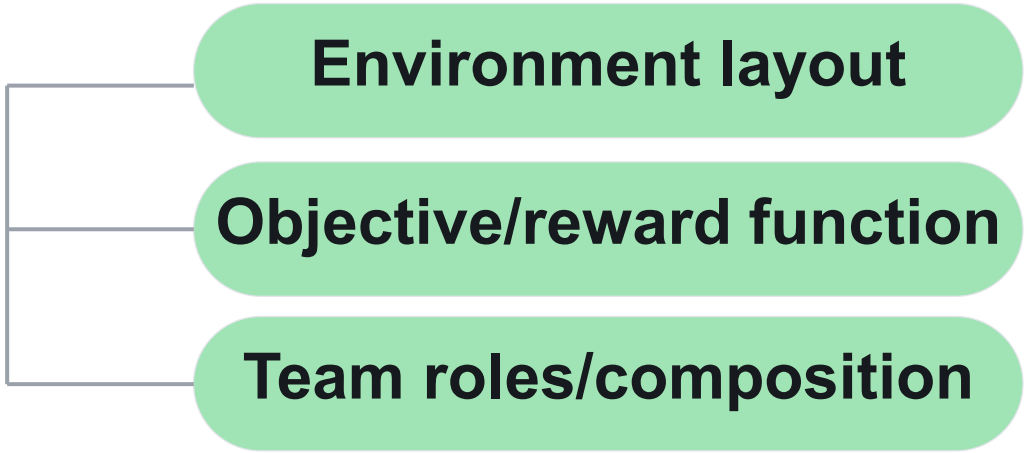
Department of Computer Science, University of Saskatchewan, Canada

MADMUC LAB

Motivation

When should a learning system stop trusting what it has learned?

Cooperative agents build their policies from past experience. In real systems, that experience can go stale when something changes.



Two ways to get it wrong

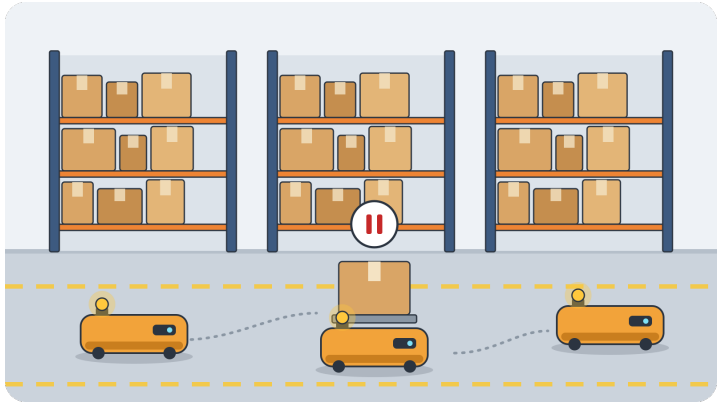
Continue as before

Stale experience is trusted too long, so learning slows or is misled

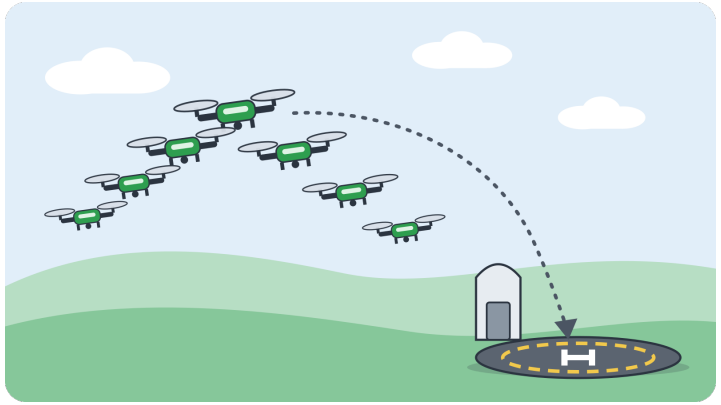
React too early or too often

Useful learned patterns are lost and training destabilizes

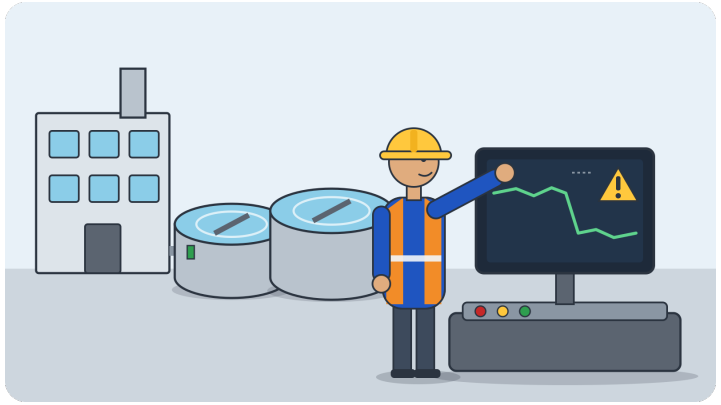
So we want to be timely and safe.



Warehouse robot team
can activate a fail-safe mode



Drone swarm
can return to base



Water treatment plant
can alert a human supervisor

Detect, then adapt

To be timely and safe, we break the response into two steps:

STEP 1 · THIS RESEARCH

Detect that a change occurred

Can trigger a fail-safe or a handover to a human

→

STEP 2

Adapt the policy

Retrain, re-explore or recover

Idea: let the rewards tell us

If the dynamics, the objective or teams changes, the team's return should change too.

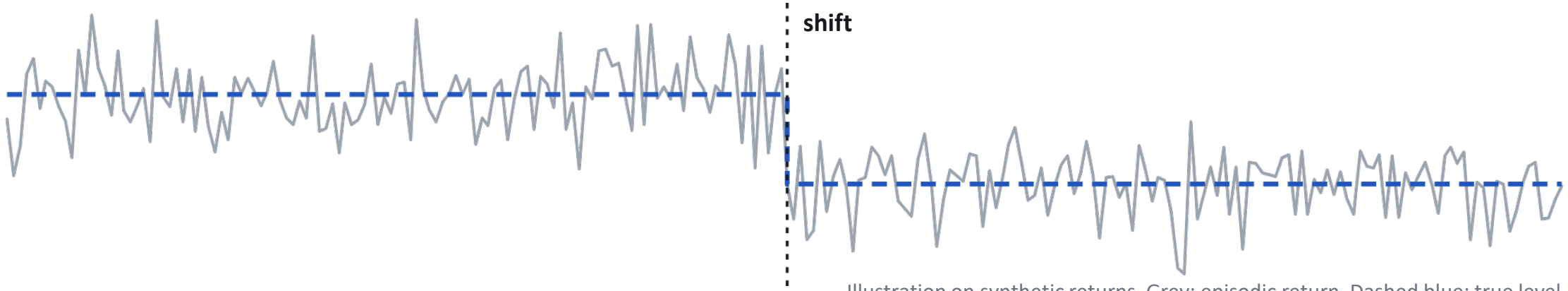
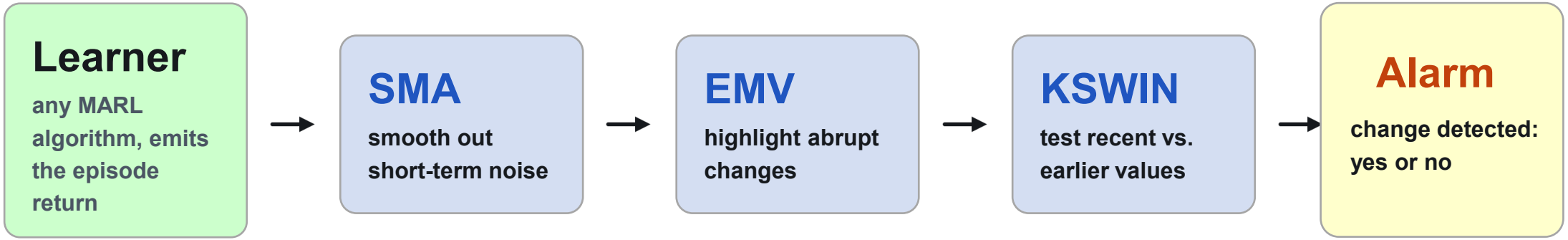


Illustration on synthetic returns. Grey: episodic return. Dashed blue: true level.

But the return is noisy: exploration, stochastic transitions and ongoing learning all add fluctuation on top of the change.

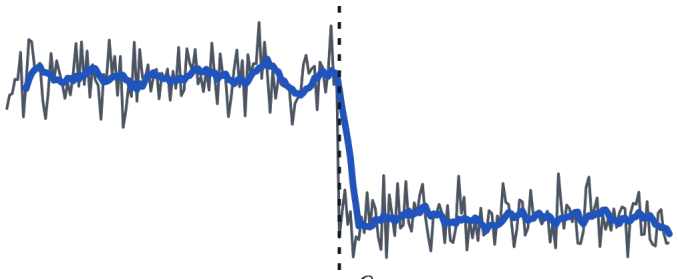
PPR: Patterns of Past Rewards

A lightweight algorithm-agnostic monitor that runs next to training and works the return.



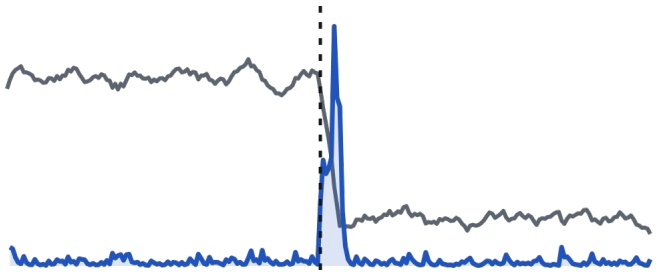
Inside PPR

STAGE 1 · SMA



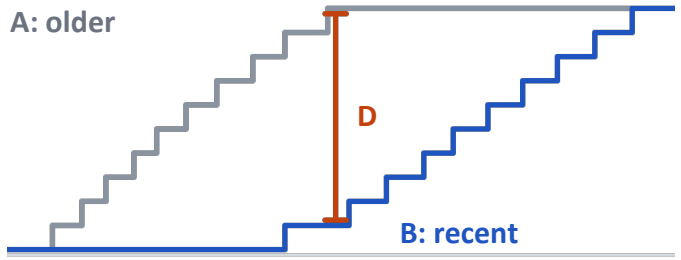
$$\bar{R}_e = \frac{1}{w} \sum_{i=e-w+1}^{e} R_i$$

STAGE 2 · EMV



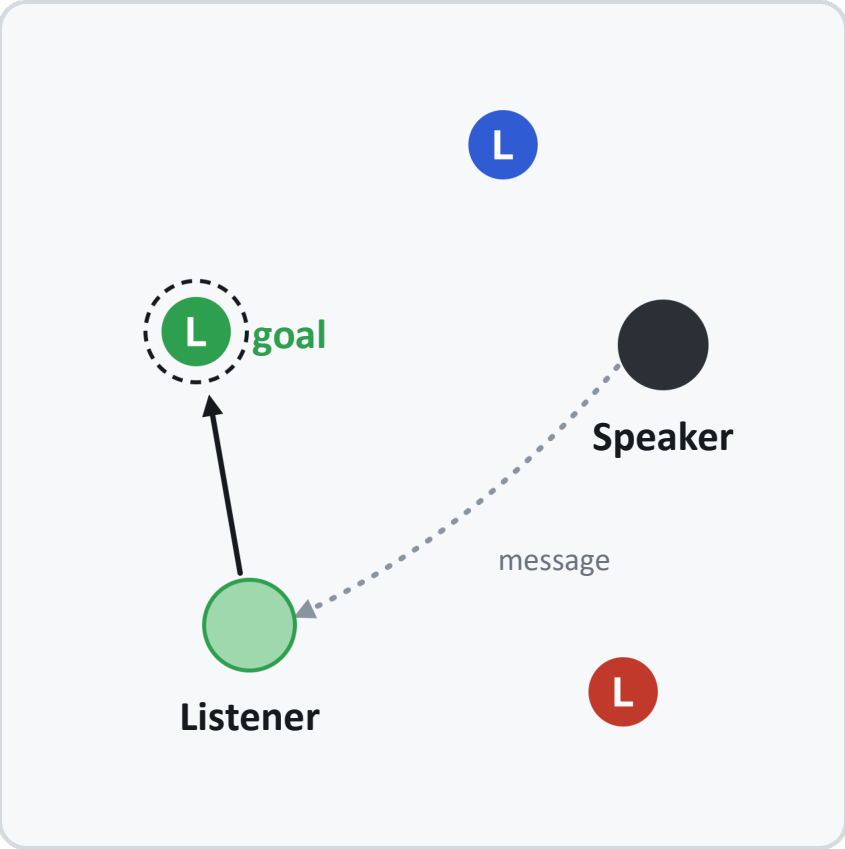
$$V_e = \beta \left(\bar{R}_e - \bar{R}_{e-1}\right)^2 + (1-\beta)\, V_{e-1}$$

STAGE 3 · KSWIN



$$D = \sup_x |F_A(x) - F_B(x)| > \tau_\alpha$$

Testbed: cooperative Speaker–Listener



Speaker

sees the goal colour, cannot move, sends a continuous 3-d message

Listener

sees the landmarks and the message, moves with continuous actions

Shared team reward

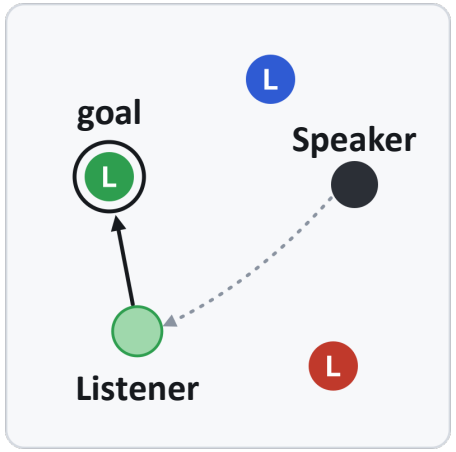
−(distance from listener to goal)², every step

Messages have no built-in meaning: the pair must invent **colour → landmark** communication convention through reward.

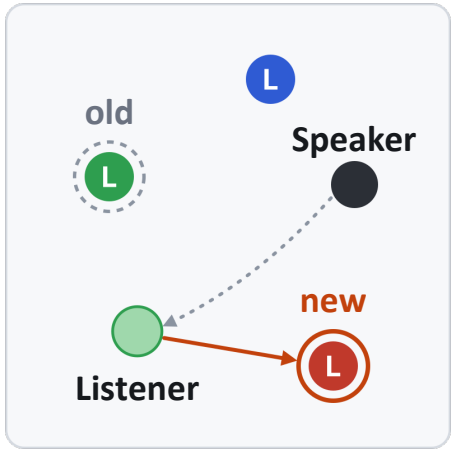
Testbed: two controlled shifts

1 Reward-function change

BEFORE



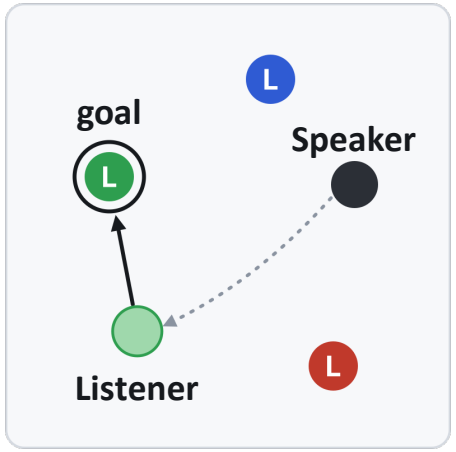
AFTER



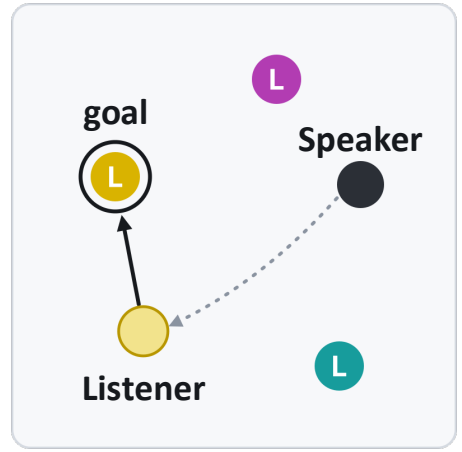
New reward target:
farthest from the original goal.

2 Landmark-colour change

BEFORE



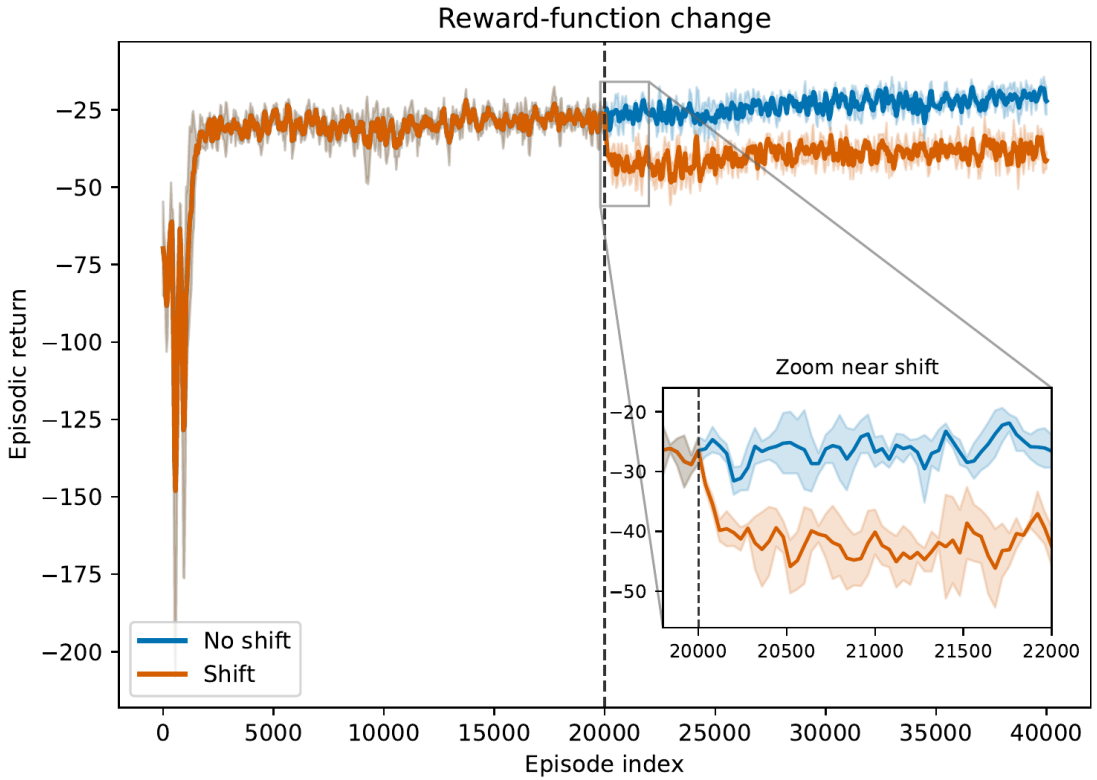
AFTER



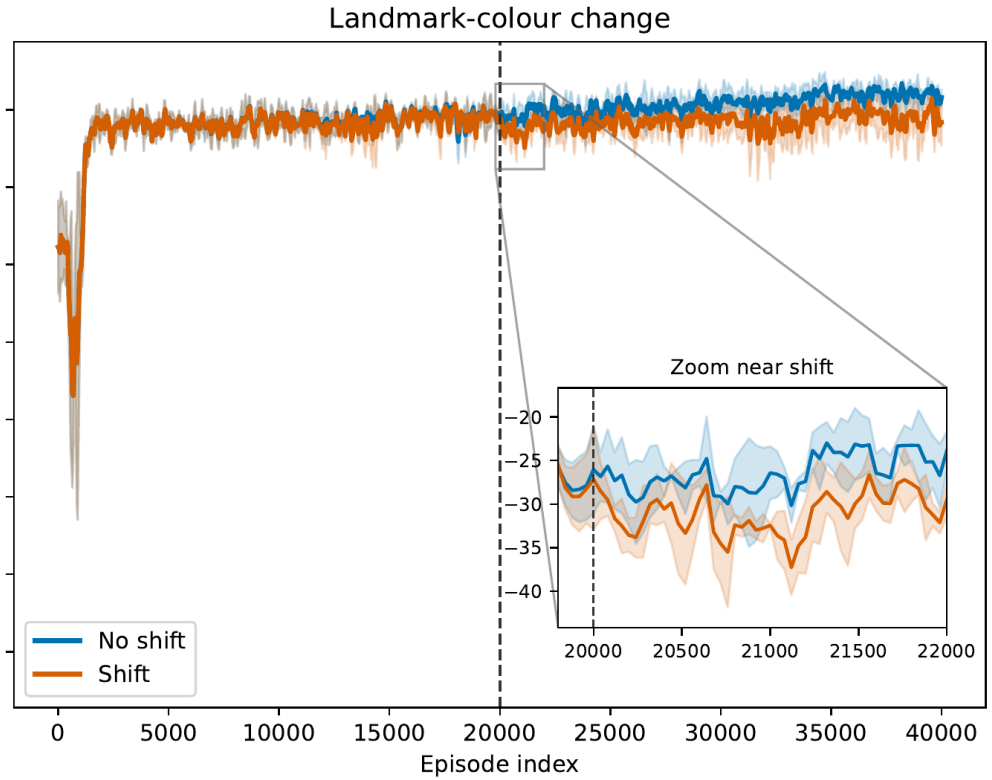
New colours.
Goal still specified by colour.

Shift effects on learning

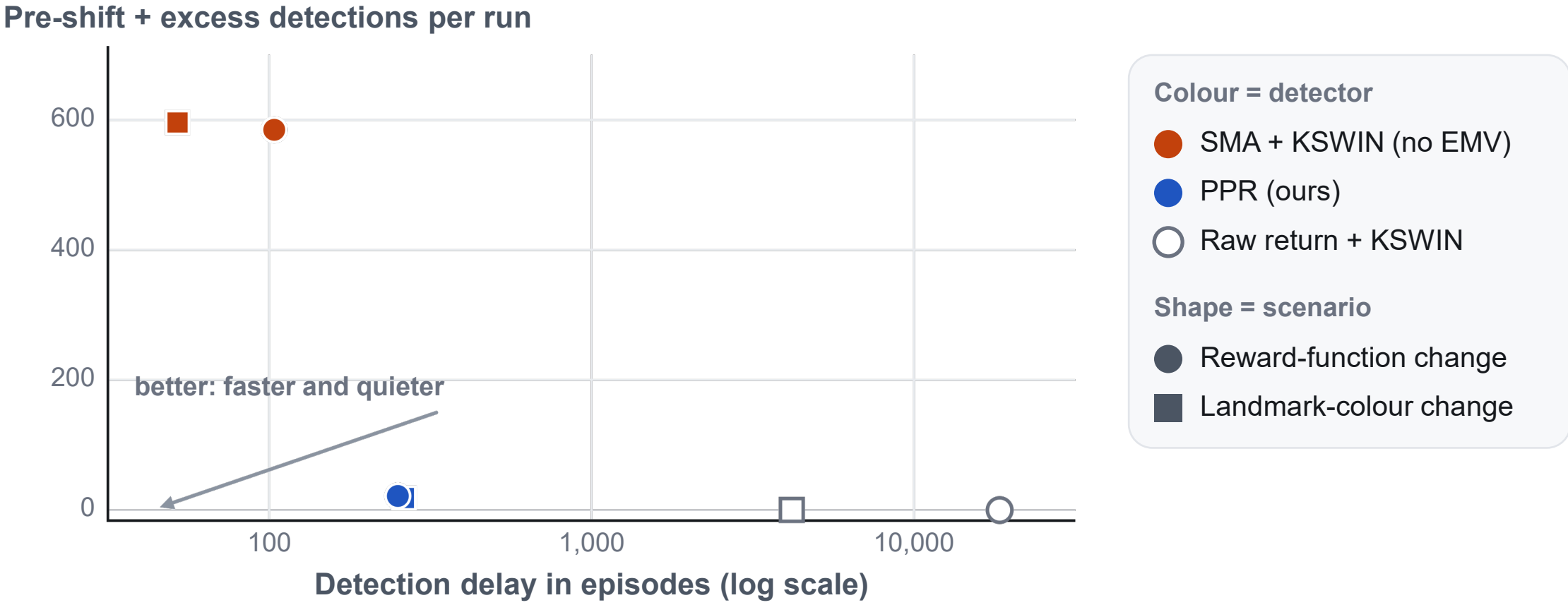
Reward change: a clear drop



Colour change: subtle drop



Results: the sensitivity–stability trade-off



Takeaways

1. Detect before you adapt! reward-derived signals can flag task shifts.
2. Preprocessing matters! raw returns miss shifts, smoothing alone floods alarms, PPR balances both.
3. PPR is lightweight and algorithm-agnostic.

Thank you — questions?

f.saberi@usask.ca · MADMUC Lab, University of Saskatchewan

Scan for the paper

University of Saskatchewan

Supported by an NSERC Discovery Grant (RGPIN-2021-03521)