ISSN: 3079-6342

Robust Memory Update Mechanisms Against Poisoning Attacks in Multi-Agent Reinforcement Learning

James Thompson[1], Oliver Bennett[2], Daniel Carter[3]*

Department of Computing, Imperial College London, London SW7 2AZ, United Kingdom

Corresponding author: d.carter@imperial.ac.uk

Abstract

In multi-agent reinforcement learning (MARL), shared replay buffers and inter-agent memory exchange introduce vulnerability to memory poisoning attacks. This work proposes a confidence-weighted memory validation mechanism integrated into the experience sharing pipeline. Each memory entry is assigned a trust score derived from temporal consistency and reward deviation metrics. A Bayesian filtering process excludes anomalous transitions before propagation to peer agents. Experiments were conducted on cooperative navigation and resource allocation benchmarks with 12–24 agents. Under a 20% poisoning injection rate, baseline MARL performance dropped by 37.8%, whereas the proposed defense limited degradation to 11.4%. Convergence time improved by 23.6% compared to anomaly-blind training.The method effectively mitigates adversarial memory contamination in collaborative reinforcement learning environments.

Keywords

Multi-agent reinforcement learning; memory poisoning; adversarial defense; replay buffer security; collaborative learning robustness

1. Introduction

Multi-agent reinforcement learning (MARL) has become an important framework for cooperative decision-making in robotics, network control, scheduling, and autonomous systems. Its strength lies in distributed interaction, coordinated policy improvement, and collective adaptation across multiple decision-making agents. In cooperative environments, agents often improve learning efficiency by exchanging experiences through shared replay buffers, message-passing modules, or parameter-sharing strategies [1]. These collaborative mechanisms support faster convergence and better use of training data, but they also introduce a critical security concern [2]. Once corrupted transitions enter a shared memory pipeline, the harmful effect is no longer limited to a single learner. Instead, poisoned experience can be repeatedly sampled, propagated to peer agents, and embedded into the broader coordination process. Recent surveys have therefore identified adversarial robustness as a central challenge in cooperative MARL, especially when the compromise of one channel can influence the behavior of the whole team [3]. This concern has become more urgent with recent evidence showing that memory corruption in collaborative environments can propagate over time and require dedicated repair mechanisms, rather than being treated as a transient disturbance [4]. Recent studies have shown that MARL is vulnerable to a range of training-time and execution-time attacks. Existing work has demonstrated that mixed action and reward poisoning can substantially degrade online MARL performance even when the attacker has only limited prior knowledge [5,6]. Related studies have shown that manipulating the experience or behavior of only one agent may be enough to contaminate the

ISSN: 3079-6342

policy development of the entire cooperative group [7]. Observation poisoning has also been reported to cause clear performance degradation under subtle input manipulation, indicating that MARL policies remain sensitive even when the attack is not visually obvious or structurally large [8,9]. More recent work has extended this line of research to coordinated attack designs against cooperative MARL systems, including strategies that jointly target a primary victim and nearby assisting agents in order to amplify disruption across the team [10,11]. Taken together, these findings show that cooperative learning provides clear efficiency gains, but it also creates direct pathways through which adversarial effects can spread across multiple learners. Most existing attack studies in MARL focus on perturbing observations, actions, or rewards. Although this line of work is important, memory poisoning operates through a different mechanism and should be studied separately. In off-policy and cooperative MARL, replay memory strongly influences what each agent learns, how quickly value estimates stabilize, and which state-action patterns are reinforced during training. Even under benign conditions, replay design has been shown to affect both convergence behavior and final performance [12]. This makes replay memory a particularly sensitive target in adversarial settings. Once malicious transitions are inserted into the replay stream, their effect may persist because the same contaminated samples can be reused many times during critic updates, target estimation, and coordinated policy refinement. The danger becomes greater in cooperative MARL because agents commonly share trajectories, transmit compressed memory summaries, or train centralized critics on pooled experience. Under these conditions, poisoned memory is not merely a one-step perturbation. It can become a durable source of statistical bias that gradually reshapes joint policy learning and weakens coordination quality. A broader literature on reinforcement learning security further confirms that poisoning during training is both practical and consequential. Backdoor poisoning studies in deep reinforcement learning have shown that malicious training signals can implant hidden attack behaviors while preserving apparently normal task performance during most of the training process [13]. Other work has shown that targeted poisoning can manipulate training data in a way that increases the likelihood of specific undesirable outcomes, even when the global training objective appears unchanged [14,15]. Certified defense methods proposed for poisoning attacks in offline reinforcement learning also point to replay data and training datasets as core security targets in modern RL pipelines. Although these studies were not designed specifically for cooperative replay sharing in MARL, they make one point clear: data poisoning in reinforcement learning is not a peripheral issue. It directly affects policy reliability, convergence stability, and operational safety. Once experience quality is compromised, the resulting policy may remain vulnerable even if the attack is sparse, delayed, or difficult to detect from short-term reward trends. Despite this progress, current defense strategies still have important limitations. Many robust MARL methods are designed for state-adversarial or observation-level perturbations rather than for poisoned memory entries that have already entered the replay pipeline. Methods developed for robust variants of cooperative value decomposition can improve resilience under state perturbation, and related robust optimization schemes can reduce performance loss under adversarial input shifts [16,17]. However, these approaches generally do not verify whether stored transitions themselves are trustworthy before they are sampled again. As a result, they address the downstream effect of contamination more directly than its origin. This distinction matters in cooperative off-policy learning, where replay memory is not a passive storage unit but an active mechanism that shapes the gradient signal throughout training. Another limitation is that many current defenses rely on adversarial training or robust optimization assumptions. These methods may be effective when the perturbation type is known in advance or when the attack pattern remains relatively stable. In realistic cooperative settings, however, poisoning may be sparse, delayed, weakly structured, or intentionally mixed with normal transitions to

ISSN: 3079-6342

avoid detection. Under such conditions, policy-level robustness alone may not be sufficient. A transition that appears individually plausible may still be harmful when viewed against the recent temporal behavior of an agent, the reward structure of the task, or the consistency of shared experience across agents. This challenge becomes even more pronounced in collaborative systems where information reuse is frequent. If suspicious transitions are allowed to circulate freely through the replay mechanism, even a mild attack can accumulate into a persistent distortion of policy updates. The experimental scope of the current literature also remains limited. Several studies successfully demonstrate attack impact on standard cooperative benchmarks, but many still rely on a small number of agents, short training horizons, or simplified attack settings that do not reflect practical collaborative learning pipelines. Some investigations are conducted in single-agent RL settings and therefore cannot capture the propagation effect that emerges when contaminated experience is exchanged among multiple learners. Other studies evaluate poisoning only at the observation or reward level, without explicitly modeling repeated contamination through replay reuse and inter-agent sharing. Because of these limitations, important questions remain insufficiently explored: how poisoned transitions spread through shared memory structures, how rapidly trust in replayed experience should decay after suspicious behavior is detected, and which validation rules can remove harmful samples without unnecessarily discarding useful information. These questions are central to the deployment of secure cooperative learning systems, yet they have not been studied with enough depth at the memory-management level. These gaps point to the need for a defense that operates directly on shared memory rather than relying only on robust policy training after contamination has already influenced the update process. In cooperative MARL, an effective memory-level defense should do more than reject obvious outliers. It should evaluate whether each transition is consistent with recent temporal patterns, whether the observed reward is plausible under the current interaction context, and whether the transition is safe to propagate to peer agents through replay sharing. This requirement is particularly important when the attacker tries to remain hidden by mixing malicious transitions with valid ones. Under such circumstances, unconditional replay sharing becomes a structural weakness. A trust-aware memory filter can therefore serve as a practical complement to robust policy learning by reducing contamination before poisoned samples affect critics, value targets, and cross-agent coordination. Recent work on poisoning resilience in reinforcement learning supports this direction by showing that the quality control of stored experience is essential for reliable training under adversarial data manipulation [18]. This study proposes a confidence-weighted memory validation mechanism for cooperative MARL under poisoning attacks. Each replayed transition is assigned a trust score derived from temporal consistency and reward deviation, and a Bayesian filtering step is used to remove suspicious samples before they are propagated across agents. Unlike conventional replay usage, which treats stored experience as equally reusable once admitted into memory, the proposed framework views experience sharing as a controlled and continuously evaluated process. The contribution of this design lies not only in attack mitigation, but also in reframing shared replay as a security-sensitive component of cooperative learning. By placing the main defense at the memory update stage, the proposed method targets the point at which poisoning first enters and begins to spread through the collaborative pipeline. The purpose of this study is therefore to improve robustness against memory contamination while preserving the learning efficiency that makes cooperative MARL attractive in the first place. In this context, the work aims to clarify whether trust-aware replay validation can reduce performance degradation under poisoning, support more stable convergence than unfiltered replay training, and provide a practical foundation for secure shared-memory learning in multi-agent systems. From a broader perspective, this study contributes to the reliability and safe deployment of cooperative MARL in real-world domains

ISSN: 3079-6342

where experience sharing is necessary, but the trustworthiness of shared data cannot be taken for granted.

2.Materials and Methods

2.1 Experimental Environment and Data Generation

The experiments were carried out in two cooperative multi-agent reinforcement learning environments: cooperative navigation and distributed resource allocation. These tasks are commonly used to test coordination among agents. The number of agents ranged from 12 to 24 depending on the task setting. During training, each agent generated sequences of states, actions, rewards, and next states. These transition records were stored in a shared replay buffer for policy updates. Each training run produced several million transition samples. To test robustness under adversarial conditions, poisoned transitions were inserted into the replay memory during training. The poisoning rate was controlled at several levels, with a maximum of 20% malicious samples mixed with normal experience data.

2.2 Experimental Design and Baseline Comparison

Two training pipelines were examined. The experimental group used the proposed memory validation mechanism, which evaluates each transition before it is shared with other agents. The control group used a standard MARL training process with a shared replay buffer and no validation step. Both groups used the same policy network structure, learning rate, exploration strategy, and training schedule. This setup ensured that the only difference between the two groups was the memory validation process. Experiments were repeated under different poisoning rates to examine how memory contamination affects learning performance. Evaluation indicators included cumulative reward, task success rate, and training convergence time.

2.3 Measurement Procedure and Quality Control

Training results were recorded during the learning process. The cumulative reward and task success rate were calculated over multiple episodes. To reduce the influence of random variation, each experiment was repeated ten times with different random seeds. The mean value and standard deviation were then calculated for each metric. Replay memory was also monitored during training to identify unusual reward values and abnormal state transitions. These checks helped ensure that the experimental results reflected stable learning behavior rather than random fluctuations. All experiments were conducted on the same computing platform to keep the measurement conditions consistent.

2.4 Data Processing and Model Formulation

The proposed defense method assigns a trust score to each transition before it enters the shared replay buffer. The score is calculated using two indicators: temporal consistency and reward deviation. Temporal consistency measures whether the state transition follows the expected system dynamics. Reward deviation measures the difference between the observed reward and the predicted reward value. The trust score T_i for transition i is defined as

$$T_i = \exp\left(-\lambda_1 D_{state} - \lambda_2 D_{reward} \right)$$

ISSN: 3079-6342

where D_{state} represents the inconsistency of the state transition, D_{reward} represents the reward deviation, and λ_1 and λ_2 are weighting parameters.

A Bayesian filtering rule is then applied to determine whether the transition should be accepted. The probability that a transition is valid is calculated as

$$P(valid|x_i)=\frac{P(x_i|valid)P(valid)}{P(x_i)}$$

Transitions with probability lower than a predefined threshold are removed before experience sharing.

2.5 Performance Evaluation Metrics

Several indicators were used to evaluate the effectiveness of the proposed defense. The cumulative reward reflects the overall performance of the cooperative task. Convergence time measures the number of training steps required for the policy to reach stable performance. The performance degradation ratio is used to measure the influence of poisoning attacks on learning outcomes. In addition, defense effectiveness is evaluated by comparing reward loss under poisoning conditions between the protected and unprotected training pipelines. These indicators provide a clear assessment of robustness, learning stability, and the ability of the proposed method to reduce the impact of poisoned memory entries.

3.Results and Discussion

3.1 Impact of memory poisoning on learning performance

The experiments show that memory poisoning has a clear effect on cooperative MARL performance. When 20% poisoned transitions were inserted into the replay buffer, the baseline method showed a performance drop of 37.8%. In contrast, the proposed defense limited the drop to 11.4%. This result shows that replay memory validation can reduce the spread of corrupted transitions among agents. The structure of shared replay learning is illustrated in Fig. 1, where experiences collected by different agents are stored in the same memory pool. Earlier studies reported that shared replay buffers improve learning efficiency, but they also allow poisoned samples to affect many agents if no validation step is used [19,20]. The results here show that filtering suspicious transitions before sharing can reduce this risk.

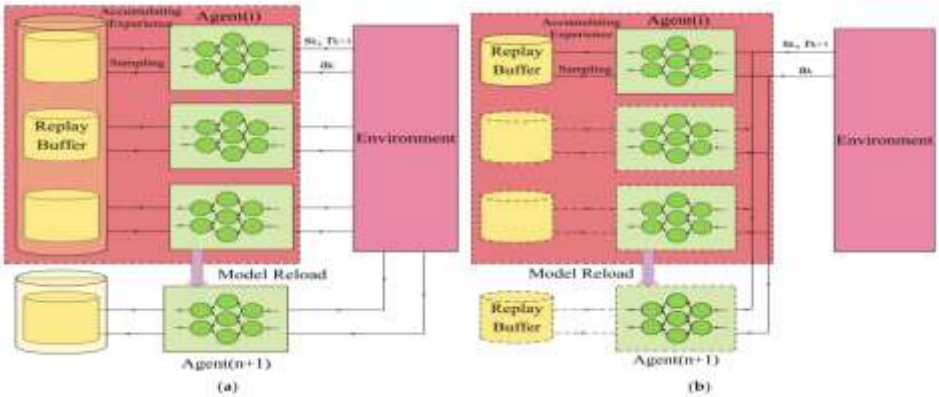

Figure 1. Shared replay memory structure used for experience exchange among agents in cooperative MARL.

ISSN: 3079-6342

3.2 Influence on convergence speed

The proposed method also improved training convergence under attack. Compared with the baseline method, convergence time improved by about 23.6%. Poisoned memory often causes unstable learning because incorrect reward signals or state transitions are repeatedly sampled during training. By removing suspicious transitions before replay, the proposed method reduced the number of harmful samples used during policy updates. Earlier studies have shown that replay buffer design and experience selection strongly affect MARL learning speed even under normal conditions. The present result confirms that replay quality becomes even more important when adversarial data are present [21,22].

3.3 Effect of trust-based filtering

The improvement in robustness mainly comes from the trust-based filtering mechanism. Each transition receives a score based on temporal consistency and reward deviation. Transitions with low scores are removed before they are shared with other agents. The process is illustrated in Fig. 2, where replay entries are evaluated before reuse. This design allows the learning system to distinguish suspicious transitions from normal experience [23]. Many previous studies on MARL security focus on observation or action attacks, while fewer studies examine validation of replay memory itself. The present results show that memory-level screening can provide an effective defense when attacks target the experience-sharing pipeline.

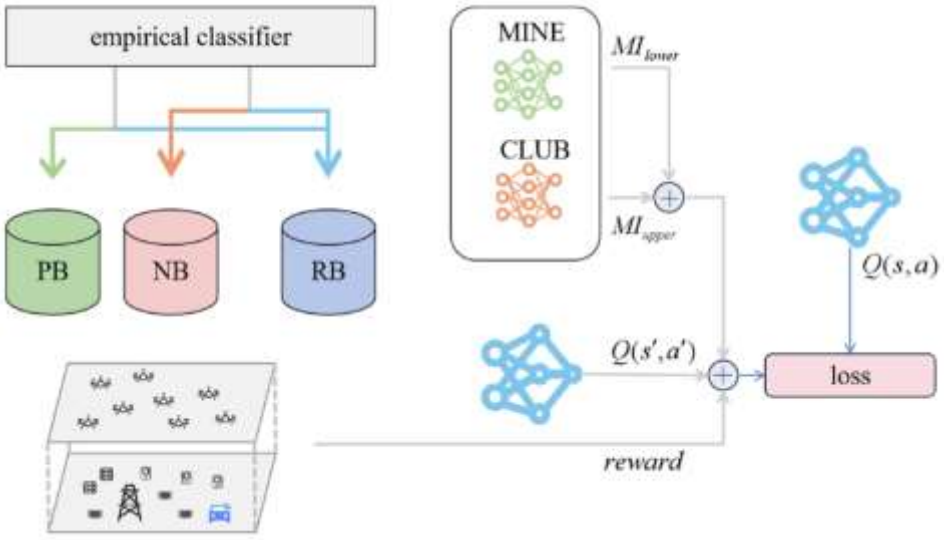

Figure 2:Screening process of replay buffer samples before reuse during multi-agent training.

3.4 Comparison with previous work and limitations

Compared with earlier research, this study focuses directly on replay buffer security in cooperative MARL. Most previous defenses improve robustness through policy design or adversarial training, while the replay pipeline itself is rarely examined. The current results show that protecting the memory stage can reduce performance loss under poisoning attacks. However, several limitations remain. The experiments were conducted on benchmark cooperative tasks with 12–24 agents, and more complex environments should be examined in future work. In addition, the validation rule uses temporal consistency and reward deviation only. Other signals, such as agent agreement or critic uncertainty, may help detect poisoned

ISSN: 3079-6342

samples more accurately. Further work is required before the method can be applied to larger cooperative learning systems.

4.Conclusion

This study examines a defense method against memory poisoning in multi-agent reinforcement learning. The method adds a trust-based validation step to the replay memory process. Each transition is checked using temporal consistency and reward deviation before it is shared with other agents. Experimental results show that the method reduces the influence of poisoned samples during cooperative learning. When 20% poisoned transitions were inserted into the replay buffer, the performance loss decreased from 37.8% in the baseline method to 11.4%. Training convergence also became faster because corrupted experiences were removed before repeated sampling. These results show that replay memory validation can improve the robustness of cooperative reinforcement learning systems. The method offers a practical way to protect experience sharing in multi-agent environments where agents depend on shared memory for policy learning. However, several limitations remain. The experiments were carried out on benchmark cooperative tasks with a limited number of agents. More complex environments and larger agent groups should be studied in future work. In addition, the current validation rule uses only temporal information and reward signals. Other indicators, such as agreement among agents or uncertainty in value estimates, may help detect poisoned transitions more accurately.

References

[1] Qiu, Y. (2024). Estimation of tail risk measures in finance: Approaches to extreme value mixture modeling. arXiv preprint arXiv:2407.05933.

[2] Nazir, A., He, J., Zhu, N., Anwar, M. S., & Pathan, M. S. (2024). Enhancing IoT security: a collaborative framework integrating federated learning, dense neural networks, and blockchain. Cluster computing, 27(6), 8367-8392.

[3] Du, Y. (2025). Research on Deep Learning Models for Forecasting Cross-Border Trade Demand Driven by Multi-Source Time-Series Data. Journal of Science, Innovation & Social Impact, 1(2), 63-70.

[4] Liu, H., Xu, D., Ma, Q., Xu, S., & Qiu, D. (2026). Memory Poisoning Propagation and Repair Mechanism in Multi-Agent Collaborative Environments.

[5] Erdem, M., & Üre, N. K. (2025). Learning to Balance Mixed Adversarial Attacks for Robust Reinforcement Learning. Machine Learning and Knowledge Extraction, 7(4), 108.

[6] Kwon, K. B., Mukherjee, S., Hossain, R. R., & Adetola, V. (2025). Security Risks of AI/ML With a Focus on Reinforcement Learning: A Review and Perspectives From Grid Applications. IEEE Transactions on Emerging Topics in Computational Intelligence.

[7] Liu, S., Feng, H., & Liu, X. (2025). A Study on the Mechanism of Generative Design Tools' Impact on Visual Language Reconstruction: An Interactive Analysis of Semantic Mapping and User Cognition. Authorea Preprints.

[8] Mohan, A., & Schön, T. (2026). Towards Robust Agents: A Survey of Adversarial Attacks and Defenses in Deep Reinforcement Learning. IEEE Access.

ISSN: 3079-6342

[1] Ma, Q., Yue, L., Xu, S., Shi, Y., & Liu, H. (2026). Web Agent Agentic Reinforcement Learning Decision Model Under Multi-Cost and Failure Risk Constraints.

[2] Brandl, B., Dyer, C. B., Heisler, C. J., & Otto, J. M. (2006). Enhancing victim safety through collaboration. Care Management Journals, 7(2), 64.

[3] Qiu, D., Xu, D., & Yue, L. (2025, December). Reinforcement Learning-Augmented LLM Agents for Collaborative Decision Making and Performance Optimization. In 2025 7th International Conference on Frontier Technologies of Information and Computer (ICFTIC) (pp. 1337-1342). IEEE.

[4] Jiang, M., Dennis, M., Parker-Holder, J., Foerster, J., Grefenstette, E., & Rocktäschel, T. (2021). Replay-guided adversarial environment design. Advances in Neural Information Processing Systems, 34, 1884-1897.

[5] Chen, H., Li, J., Ma, X., & Mao, Y. (2025, June). Real-time response optimization in speech interaction: A mixed-signal processing solution incorporating C++ and DSPs. In 2025 7th International Conference on Artificial Intelligence Technologies and Applications (ICAITA) (pp. 110-114). IEEE.

[6] Schwarzschild, A., Goldblum, M., Gupta, A., Dickerson, J. P., & Goldstein, T. (2021, July). Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In International Conference on Machine Learning (pp. 9389-9398). PMLR.

[7] Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., ... & Goldstein, T. (2022). Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2), 1563-1580.

[8] Li, T., Liu, S., Hong, E., & Xia, J. (2025). Human Resource Optimization in the Hospitality Industry Big Data Forecasting and Cross-Cultural Engagement.

[9] Ali, M. N., Soliman, M., Mahmoud, K., Guerrero, J. M., Lehtonen, M., & Darwish, M. M. (2021). Resilient design of robust multi-objectives PID controllers for automatic voltage regulators: D-decomposition approach. IEEE Access, 9, 106589-106605.

[10] Gu, X., Yang, J., Tian, X., & Liu, M. (2025). Research on the Construction of a Human-Machine Collaborative Anti-Money Laundering System and Its Efficiency and Accuracy Enhancement in Suspicious Transaction Identification.

[11] Kim, W., Shin, Y., Park, J., & Sung, Y. (2023). Sample-efficient and safe deep reinforcement learning via reset deep ensemble agents. Advances in neural information processing systems, 36, 53239-53260.

[12] Yang, Y., Leuze, C., Hargreaves, B., Daniel, B., & Baik, F. (2025). EasyREG: Easy Depth-Based Markerless Registration and Tracking using Augmented Reality Device for Surgical Guidance. arXiv preprint arXiv:2504.09498.

[13] Kumari, L., Wang, S., Zhou, T., & Bilmes, J. A. (2022). Retrospective adversarial replay for continual learning. Advances in neural information processing systems, 35, 28530-28544.

ISSN: 3079-6342

[14] Bai, W., Wu, Q., Wu, K., & Lu, K. (2024). Exploring the Influence of Prompts in LLMs for Security-Related Tasks. In Workshop on Artificial Intelligence System with Confidential Computing (AISCC 2024)(San Diego, CA). USA. https://dx. doi. org/10.14722/aiscc.

[15] Roseline, J. F., Naidu, G. B. S. R., Pandi, V. S., alias Rajasree, S. A., & Mageswari, N. (2022). Autonomous credit card fraud detection using machine learning approach☆. Computers and Electrical Engineering, 102, 108132.