Adversarial Backdoors in Deep Reinforcement Learning:

Novel Supply Chain Vulnerabilities and Detection Strategies

Sanyam Vyas

A thesis submitted for the degree of

Doctor of Philosophy

Cardiff University

August 2025

Abstract

The rapid integration of Deep Reinforcement Learning (DRL) into real-world infrastructure such as autonomous vehicles, cyber defence and healthcare has introduced critical vulnerabilities in the AI supply chain. This thesis investigates adversarial backdoors (also known as trojans) in which an adversary embeds hidden malicious behaviours that remain dormant until activated by a specific environmental trigger. Unlike traditional adversarial examples, backdoors exploit the overparameterisation of neural networks to bypass standard validation protocols while maintaining high performance on clean data.

This research challenges the unrealistic high-privilege assumptions in existing literature, which typically assume an adversary has full control of the training pipeline. Instead, it introduces three novel attack vectors that operate under substantially reduced adversarial privileges. TrojanentRL shows that backdoors can be injected in DRL by corrupting under-audited auxiliary components, specifically the Rollout Buffer, without modifying the main codebase. InfrectroRL demonstrates data-free, post-training attacks that manipulate DRL model weights after training but before deployment. InfRLhammer introduces the first inference-phase exclusive backdoor in DRL by using hardware-level Rowhammer-induced bit-flips in DRAM to subvert an agent trained under perfectly benign conditions.

To counter these threats, this thesis proposes Neural Watchdog, a real-time detection framework that identifies backdoors by monitoring internal neural activation patterns. It shifts defence from external observations, which can be bypassed by in-distribution triggers, to the agent's internal neural activations. Experiments in MiniGrid and Atari show the attacks evade state-of-the-art sanitisation defences while maintaining high clean data accuracy.

In practice, a compromised RL agent could cause catastrophic failure, from ignoring traffic cues to triggering harm or data exfiltration. Although evaluation was limited to benchmark environments like MiniGrid and Atari, the work expands the backdoor threat landscape and provides a clear pathway to extend these defences to domains such as Autonomous Cyber Network Defence and Autonomous Vehicles.

Table of contents

List of figures

List of tables

List of publications

This thesis is primarily based on the following peer-reviewed publications, which present the core technical contributions developed during the course of the PhD:

- S. Vyas, V. Mavroudis, and P. Burnap, "Towards the Deployment of Realistic Autonomous Cyber Network Defence: A Systematic Review," *ACM Computing Surveys*, 2025.
- S. Vyas, C. Hicks, and V. Mavroudis, "Mitigating Deep Reinforcement Learning Backdoors in the Neural Activation Space," in *Proceedings of the 2024 IEEE Security and Privacy Workshops (SPW)*, pp. 76–86, 2024.
- S. Vyas, A. Caron, C. Hicks, P. Burnap, and V. Mavroudis, "Beyond Training-time Poisoning: Component-level and Post-training Backdoors in Deep Reinforcement Learning," in Proceedings of the Fortieth AAAI Conference on Artificial Intelligence (AAAI-26), 2026.
- S. Vyas, A. Caron, C. Hicks, P. Burnap, and V. Mavroudis, "InfRLHammer: Inference-stage Backdoor Attack on Deep Reinforcement Learning via Weight Replacement," (*In Preparation for ICML 2026*).

The following additional publications were produced during the PhD, but are not included as part of this thesis:

- E. Shereen, D. Ristea, S. Vyas, S. McFadden, M. Dwyer, C. Hicks, and V. Mavroudis, "SoK: On Closing the Applicability Gap in Automated Vulnerability Detection," *arXiv preprint arXiv:2412.11194*, 2024. (*In Preparation for ACM Computing Surveys*)..
- W. Feng, S. Vyas, and T. Li, "Autonomous Cyber Defence by Quantum-Inspired Deep Reinforcement Learning," *SciTePress*, 2025.
- M. Meijer, S. Vyas, V. Mavroudis, and M. Juarez, "Adversarial Evasion Against Autonomous Cyber Defence Agents," *Proceedings of the Workshop on Autonomous Cybersecurity (at ESORICS)*, 2025.

This thesis is dedicated to my parents, who have stood by me and kept me motivated during the most challenging phases of this academic journey. . . .

Acknowledgements

This journey would not have been possible without the support of some truly important individuals. I am deeply grateful to my supervisory team for their insightful guidance and for giving me the freedom to pursue my PhD in the direction I envisioned.

During the most challenging phase of this degree, I had the privilege of collaborating with researchers at the AI for Cyber Defence Research Centre at the Alan Turing Institute, led by Chris and Vasilios. The weekly flash meetings and the lively post-lunch discussions with Chris, Alberto, Liz, Isaac, Myles, Stephen, Fraser, Ilya, Shae, Maddi, Dan, Ezz, and Burak not only enriched my research but also provided me with a sense of belonging and balance.

A particularly formative influence on my academic career has been the mentorship of Vasilios, both during and beyond my internship at the Alan Turing Institute. His encouragement during periods of difficulty, recognition during moments of success, and ability to instill an ambitious research vision played a pivotal role in shaping this thesis and my broader outlook as a researcher.

I was very lucky to meet some amazing people along the way. To Vageesh, Dushani, Vikalp, Shobhna and Vanshika, I will always cherish the wonderful moments spent with each one of you.

Finally, I owe the deepest gratitude to my family. To my brother and sister-in-law, I am thankful for your constant care and for the many cheerful dinners and conversations since 2023 that helped me maintain balance and perspective outside my academic work. Your presence has been a source of reassurance, and I deeply value the way you have looked after my wellbeing during this time.

To my parents, I remain profoundly indebted for your unwavering faith and unconditional support throughout my academic journey. From the very beginning, it was your encouragement that instilled in me the confidence to pursue higher education and to persevere through

moments of doubt. Your sacrifices, patience, and constant belief in my abilities gave me the resilience to complete this thesis.

This accomplishment belongs not only to me but to all five of us as a family. It is a reflection of the values you have instilled, the support you have provided, and the love that has carried me through every stage of this long journey.

Chapter 1

Introduction

1.1 The Rise of Autonomous Systems

In an era of unprecedented digital transformation of safety-critical systems, the last decade has seen a rapid expansion in *autonomous systems* deployed across safety-critical domains. Rapidly transforming verticals include transport [177, 204, 111, 118], energy [47, 242, 211], aerospace [94, 198], healthcare [77, 61] and industrial automation [191, 140]. These systems are in the process of incremental transformation towards greater system autonomy through *decision-making agents* rather than static controllers. These agents will be required to continually interpret noisy sensor streams, reason over latent systems states and select actions whose consequences unfold over extended horizons. As a result, *sequential-decision making* algorithms have emerged as a unifying lens for autonomy.

Unlike the previous wave of automation, which focused on the paradigm of computing a "single best output" as a solution, this transformation aims at choosing a "sequence of actions" that optimise the long-term objectives under high levels of uncertainty. This process naturally formalises autonomy as a control-and-inference loop operating in a dynamical world, where decision quality depends on temporal credit assignment, delayed effects, and the ability to trade off short-term costs against long-term safety and performance.

The practical importance of sequential decision-making is most visible in complex autonomous platforms, where the environment is non-stationary and tightly coupled to the agent's actions. In robotics, sequential decision-making methods have enabled adaptive control and policy learning for manipulation and locomotion, moving beyond brittle scripted behaviours [227]. In industrial systems [163], similar principles underpin adaptive scheduling [56] and closed-loop optimisation in the presence of disturbances and operational constraints. In high-impact scientific and engineering settings, sequential decision-making has also been used to manage processes governed by complicated physical dynamics, demon-

strating how learned or optimised policies can outperform hand-crafted strategies when system behaviour is difficult to specify analytically [53]. These developments collectively suggest that autonomy at scale increasingly hinges on decision-making algorithms that can operate robustly over time, rather than one-shot predictors.

However, the same assumptions that make the above examples autonomously tractable also impose sharp limitations as autonomy is pushed into open-world, adversarial, and high-dimensional settings. Several structural factors increasingly constrain classical sequential automation including combinatorial complexity, partial observability and limited adaptivity. These limitations have become more pronounced as autonomy is expected to operate continuously, coordinate across heterogeneous subsystems, and respond to rare but high-impact events. In such regimes, the central challenge is not merely producing correct actions under nominal conditions, but sustaining safe behaviour under uncertainty, distribution shift, and strategic adaptation. This has motivated a growing shift toward *learning-based sequential decision-making*, where agents can improve policies through experience and discover robust strategies that are difficult to design explicitly [121, 221].

1.1.1 Deep Reinforcement Learning for Autonomous Systems

Against this backdrop, Deep Reinforcement Learning (DRL) offers a particularly compelling route to autonomy in domains where the environment is complex, partially observed, and continuously evolving. By learning policies that optimise cumulative objectives through interaction, DRL provides a mechanism for constructing adaptive decision-makers that explicitly reason over action sequences and delayed outcomes [221].

A defining characteristic of DRL is its ability to couple sequential decision-making with high-capacity function approximation, enabling policies to be learned directly from high-dimensional observations such as images, sensor streams, or network telemetry [159, 137]. This end-to-end learning paradigm reduces reliance on manually engineered state representations and domain-specific heuristics, which often become brittle as system complexity increases [131, 214]. As a result, DRL has demonstrated strong empirical performance in domains where accurate system models are unavailable, incomplete, or prohibitively expensive to maintain.

Crucially, DRL frameworks naturally support long-horizon optimisation through the maximisation of expected cumulative reward. This allows objectives such as operational efficiency, safety margins, and system resilience to be embedded directly into the learning objective, rather than enforced through external supervisory logic [8, 75]. In contrast to classical control approaches that operate over fixed or receding horizons, DRL policies

can internalise long-term dependencies and adapt behaviour based on experience, enabling decision-making that accounts for delayed consequences and rare but high-impact events [59].

DRL also offers a degree of adaptivity that is difficult to achieve with static or model-based controllers. Through continued interaction with the environment, agents can refine their policies in response to non-stationarity, distributional shift, or changes in operating conditions [130]. This capability is particularly relevant in real-world autonomous systems, where assumptions encoded at design time may no longer hold during deployment. Empirical successes in robotics, control, and large-scale optimisation illustrate how DRL can discover strategies that are difficult to specify explicitly, especially in settings involving complex feedback loops or adversarial dynamics [215]. Despite these advantages, the application of DRL to safety-critical autonomous systems remains challenging. Learned policies may exhibit unexpected behaviour outside the training distribution or fail to generalise under rare but consequential conditions. Moreover, the opacity of deep neural policies complicates verification, interpretability, and certification [141]. These challenges have motivated a growing body of research focused on improving training stability, sample efficiency, and robustness, as well as incorporating constraints, uncertainty estimation, and human oversight into the learning process.

1.1.2 Hurdles Towards Deep Reinforcement Learning Deployment

Overall, the successful deployment of DRL in highly complex environments *typically* necessitates meticulous attention to several critical components: the simulation environment, agent architecture, training methodology, deployment infrastructure, and continuous evaluation framework. Each component presents unique challenges that must be systematically addressed. The simulation environment must faithfully capture the complexity and dynamics of real-world networks whilst balancing computational feasibility. Agent architectures require careful design to ensure appropriate state representation, action space construction, and reward formulation that aligns with security objectives [8]. The training pipeline must incorporate robust adversarial testing, out-of-distribution detection, and safety constraints to prevent unintended behaviours.

Furthermore, for safety-critical systems that require DRL such as Autonomous Cars and Autononomous Cyber Network Defence (ACND) [15], the deployment infrastructure demands particular attention to latency requirements, failover mechanisms, and human oversight capabilities. Neglecting any of these fundamental building blocks invariably results in not merely suboptimal performance, but potentially catastrophic security implications, as inadequately designed autonomous systems may create new attack surfaces or fail to respond appropriately to novel threats [178]. Moreover, the entire DRL supply chain,

from data collection and environment construction to model deployment and monitoring, requires comprehensive security hardening to prevent adversarial manipulation. Without sufficient safeguards, adversaries could potentially compromise model integrity through various ways and forms of security threats, leading to imperceptible but critical degradation in defensive capabilities that may remain undetected until exploited. Even after the attack is conducted, some may remain undetected even during post-attack forensics, due to their inherent evasiveness [152].

1.2 Backdoors In Deep Reinforcement Learning Algorithms

As mentioned in the previous section, DRL is highly susceptible to adversarial and poisoning attacks that can cause significant damage to the systems in which they are deployed. *Adversarial attacks* typically operate at inference time by introducing carefully crafted perturbations to the agent's observations in order to induce incorrect action selection. These attacks exploit local sensitivities of the learned policy and often result in immediate performance degradation or unstable behaviour. Over the last decade, researchers have made significant strides in this domain through the development of increasingly sophisticated attack models, as well as a wide range of defensive techniques, including adversarial training, input preprocessing, robust policy learning, and certified defences. As a result, adversarial perturbations in DRL are now relatively well characterised within the broader AI security literature, with a growing understanding of both their limitations and their operational impact.

Nevertheless, while adversarial attacks primarily induce transient decision errors through input manipulation, they do not fundamentally alter the underlying policy itself. In contrast, attacks that compromise the policy representation or its learned decision structure pose a qualitatively different and more severe risk.

Backdoor attacks, also referred to as trojan attacks, constitute a more stealthy threat, with comparatively limited investigation in the context of DRL. A backdoor is a targeted vulnerability deliberately embedded within a learning system, enabling adversary-controlled behaviour through the activation of a hidden trigger. In DRL settings, triggers may be instantiated as specific visual patterns, sensor configurations, or environmental states that cause the policy to transition into an alternative behavioural regime. This mechanism allows the agent to remain benign under standard operating conditions while exhibiting adversary-specified behaviour only when the trigger condition is satisfied.

The threat posed by backdoors is exacerbated in DRL due to the sequential and reward-driven nature of policy learning. Rather than causing isolated decision errors, a backdoored policy can introduce persistent deviations in the agent's decision-making trajectory, resulting

in unsafe actions, mission failure, or systematic violations of safety constraints. For instance, an autonomous driving agent may be induced to ignore traffic signals under specific environmental conditions, or a robotic system may execute hazardous control sequences in response to particular sensor states. Crucially, such behaviours may manifest over extended temporal horizons, allowing failures to arise in a delayed and non-local manner. Given the subtlety and context-dependent nature of the triggering conditions, detecting these attacks in practice is highly challenging, positioning backdoors as a significantly more severe bottleneck than other security threats in DRL.

From an attack-surface perspective, backdoors can be introduced at multiple stages of the DRL pipeline. During training, adversaries may manipulate environment dynamics, modify reward signals, or contaminate experience data to associate the trigger with malicious behavioural outcomes. Beyond training-time attacks, backdoors may also be inserted post hoc through direct modification of model parameters, corruption of pre-trained or open-source RL components, or exploitation of hardware and system-level vulnerabilities during deployment. The diversity of insertion pathways complicates assurance and underscores the extent to which backdoors represent a critical obstacle to the trustworthy deployment of DRL systems.

1.2.1 Mitigation and Defences against Existing Backdoors

Very limited research works [20, 36, 4, 248] have explored mitigation and defences against DRL backdoors in literature. Publications above have assumed highly out-of-distribution backdoor triggers and generated backdoor behaviours that are skewed towards one particular action (i.e., move paddle to the right upon backdoor trigger appearance). While these backdoor trigger is mathematically possible in the input observation distribution perspective, it's realism in the real world has not been explored in the real world. In this thesis, we explore the highly sophisticated state-of-the-art defence by Bharti et al [20] and implement a backdoor trigger that could potentially evade its safe subspace sanitisation algorithm.

This thesis also identifies several key bottlenecks in the literature, which explores how the defences sanitise the poisoned policy from backdoor triggers. One key bottleneck we identified in *all* defences was the time it took for them to defend against the backdoor policy attacks. Specifically, Authors from [20] took 4 times longer to sanitise the backdoor policy compared to the time it takes to usually train the policy. In realistic domains, where it can potentially take months to train an algorithm, sanitisation of the backdoor using this algorithm in equal ratio will take around a year! As a way to make strides towards real-time sanitisation, *this thesis utilises a method to detect the most evasive (in-distribution) backdoors in the MiniGrid environment used often in literature* [12].

1.2.2 Exposing a Wider Threat Landscape

Given the very recent popularity of DRL [157, 202], its security threats in the domain were never addressed until the 2020s. Until now, only [235, 117, 50, 193, 192] have addressed security issues of DRL. However, all attacks require excessively high privilege levels such as access to the training pipeline. In addition, all attacks require sophisticated knowledge of the area DRL is applied into, along with a thorough understanding of the code. Lastly, the attacks are also conducted in the most utilised files of the project code, which make it susceptible to detection through code auditing mechanisms. While these cases are prevalent within lab-based conditions, their occurrence in the real world could be questioned. In addition, the method of assessment of the damage caused by backdoor attacks on DRL systems should also be gauged upon.

Moreover, existing papers on DRL backdoors use the same metrics used in supervised learning, which could potentially be redundant (or less useful) in the domain of DRL. *This thesis explores the different aspects of the DRL supply chain in which the adversary could inject stealthy backdoors and cause equivalent or more significant damage. In addition, this thesis also discovers more appropriate forms of assessing the impact of backdoors in DRL that are unique to decision-making systems.*

1.3 Gaps in this Research

In order to ensure the successful deployment of autonomy in safety-critical and mission-critical systems, substantial progress must be made in securing the DRL supply chain. Nevertheless, we first elucidate how DRL offers distinctive advantages for complex sequential decision-making compared to other contemporary algorithms in a real research literature application example of ACND. We specifically highlight DRL's advantages by situating it within the broader literature of autonomous decision-making and contrasting its capabilities with alternative model-based and rule-driven approaches. This thesis then focuses on DRL backdoors, a critical security bottleneck in which malicious behaviours remain dormant during standard validation and evaluation, yet cause catastrophic system failures when activated.

We present a comprehensive analysis of the DRL threat landscape, introducing novel backdoor attack methodologies that exploit vulnerabilities across multiple stages of the learning pipeline, including the component level, post-training modification, and inference-time deployment. In addition, we also propose mitigation and real-time detection strategies aimed at improving the robustness and trustworthiness of DRL agents. Although the proposed methods are evaluated using standard benchmark environments such as MiniGrid and Atari,

the resulting insights and techniques provide a foundational framework for securing the future deployment of autonomous agents in complex real-world operational domains.

1.3.1 Assessing Limitations in DRL Backdoor Defence Literature

While the overall AI research community (researchers publishing specifically to top-tier conferences and journals along with other important papers) are vigilant on ways to detect flaws in state-of-the-art AI backdoor defences, *no works have assessed backdoor defences in DRL applications*. This is a problem because many of the defences often rely on unrealistic theoretical assumptions and prove their backdoor defence against a very small minority of backdoor triggers, potentially making them redundant against an adversary that crafts backdoor triggers that blend into the environment.

Therefore, testing the robustness of popular defences is necessary to explore methods that can allow us to discover and design more detection and mitigation-based backdoor defences.

1.3.2 Real-time DRL Backdoor Detection

Previous work and the existing defences have not sought to address one of the biggest requirements of backdoor detection (and potentially sanitisation) of poisoned DRL policies – real-time detection. All existing defences [20, 89, 36, 248], while detect/sanitise backdoor policies to a specific level, require a significant amount of time to complete their objective. In areas such as ACND, where training will require a significant amount of time due to the complexities of real-world enterprise networks, the objectives of the papers mentioned above (such as partially retraining policies) will take an unaffordable amounts of time to clean the policy from backdoor triggers and/or behaviours. In addition, future backdoor threats may also involve multiple backdoor triggers which may evade detection/sanitisation of the existing defences. Therefore, there exists a need to detect the most evasive backdoors as soon as they appear in the input observation, preventing them from (or alerting authorities about the agent) making malicious behaviour that could potentially cause significant damage to the system DRL algorithms are operated on.

In the area of decision-making algorithms such as DRL, there has been a lack of assessment of agent behaviour through neuron activation, a key metric to determine episodic transitions and long-term actions. Therefore, we address this gap through our work.

1.3.3 Expanding the Threat Model

While existing research has significantly advanced the understanding of backdoor threats in DRL, current literature remains predominantly focused on training-time attacks. These approaches typically involve modifying elements such as the reward function or the action selection process during training. Some studies extend this further by altering transition probabilities to manipulate the learned policy when a trigger is present. Although these methods have demonstrated effectiveness, they are inherently complex and often lack practicality in real-world settings due to their dependence on extensive access to the training pipeline.

This thesis addresses a critical gap in the field by investigating overlooked vulnerabilities across the broader DRL supply chain, particularly those arising prior to deployment. Notably, no prior work has systematically explored how reduced adversarial access could still yield successful backdoor attacks. Existing attack models typically assume full access to training-time code and input observations, resulting in high computational costs and limited applicability under constrained conditions.

The research presented here challenges prevailing security assumptions in DRL by introducing novel backdoor attack strategies that are not only highly effective and evasive but also operate under significantly stricter adversarial constraints. These methods demonstrate the feasibility of poisoning DRL policies outside of the traditional training-time context. In several cases, the attacks require no access to training, validation, or testing data, and involve considerably lower computational overhead. By doing so, this work highlights previously unaddressed security vulnerabilities that must be mitigated before DRL systems can be safely deployed in autonomous defence roles within enterprise environments.

1.3.4 Inference-based Backdoors

As we address vulnerabilities in DRL supply chain up until the deployment, we explore backdoor vulnerabilities in the deployment stage of the pipeline.*Similar to the rest of the stages of the DRL supply chain, no works have explored backdoor threats during inference in DRL models. Existing backdoor attacks in literature depend solely on threats that exist through perturbations in the code prior to deployment. To fill the gaps in the literature, we explore specialised hardware fault injection techniques to explore methods to inject backdoor behaviours into deployed models given some modifications to the input observation during inference.*

While the supervised learning literature has explored fault injection techniques, its effects in the domain of DRL and decision-making agents is yet to be explored in literature. In supervised learning, the effects of a benignly trained model under hardware fault injection

could replicate a backdoor'ed model that misclassifies an input image based on a backdoor trigger. In decision-making algorithms like DRL, an activated hardware fault injection during an episode could result in the agent being forced to follow a policy that is highly suboptimal through taking an action(s) targeted by the adversary. In the domain of ACND, this security flaw must be addressed in DRL to alert the community about hardware fault injection attacks that can potentially occur on benignly DRL ACND agents, leading to potentially dangerous autonomous decisions made in the enterprise networks.

1.4 Research Questions

Given these gaps in previous literature identified above, this thesis seeks to address the following research questions which are raised in response:

1. **RQ1**: What are the key requirements for the deployment of DRL agents in safety-critical sequential decision-making tasks, as illustrated by the application example of ACND?

    (a) **RQ1.1**: Drawing on existing literature of the ACND application example, what fundamental research gaps and security vulnerabilities must be addressed to enable the reliable transition of DRL agents from controlled benchmark environments to complex real-world operational domains? Additionally, which research gap requires immediate research contributions for the successful deployment id DRL in such safety-critical domains?

2. **RQ2:** To what extent have existing DRL backdoor defences been rigorously evaluated for their credibility against *realistic* backdoor threats?

3. **RQ3:** What are the key limitations in *all* existing defences in the defences against DRL backdoors that must be addressed? And how can we implement a more effective method to detect backdoors in DRL before they induce malicious behaviours?

4. **RQ4:** What underexplored vulnerabilities exist in the DRL supply chain prior to deployment, particularly before and after the training phase?

    (a) **RQ4.1:** What attack methods within the DRL supply chain can be executed with reduced adversarial privileges compared to existing approaches in the literature?

5. **RQ5:** Are there methods in the literature that could be utilised to deploy backdoor attacks on benignly trained models solely during inference?

1.5 Contributions

This thesis contributes to advancing the secure deployment of DRL techniques within the context of Autonomous Cyber Network Defence. It identifies critical and previously unexplored vulnerabilities in the DRL supply chain, and proposes lightweight mitigation strategies that may inform both future research and practical implementations. The following is a bulleted list of all contributions and their associated publications. The researcher contributions made in our technical chapters (and their corresponding publications) are highlighted in bold.

- ***C1***: This thesis first contributes to ACND through a systematic literature review. It clarifies key terminologies and identifies subdomains such as Autonomous Blue and Red Teaming Agents and Autonomous Cyber Operations Gyms. Through an analysis based on a defined set of ACND requirements, it highlights significant research gaps and open challenges that must be addressed for the effective deployment of DRL-based ACND solutions in real-world enterprise networks.

  ***C1* was published in [230].**
  S. Vyas, V. Mavroudis, and P. Burnap, "Towards The Deployment of Realistic Autonomous Cyber Network Defence: A Systematic Review," ACM Computing Surveys, 2025.

- *C2*: This thesis identifies the challenge of securing DRL against backdoor attacks and critically evaluates a state-of-the-art defence [20], revealing its theoretical limitations through empirical experimentation.

- ***C3***: Building on the limitations identified, a lightweight and novel backdoor detection system is designed to detect highly evasive triggers within the environment, preventing the execution of malicious backdoor behaviours.

  ***C2* and *C3* were published in [231].**
  Vyas, S., Hicks, C., and Mavroudis, V. (2024). Mitigating Deep Reinforcement Learning Backdoors in the Neural Activation Space. In 2024 IEEE Security and Privacy Workshops (SPW), pages 76–86. IEEE

- *C4*: This thesis uncovers previously unexamined vulnerabilities in the DRL supply chain and develops a comprehensive threat model that encapsulates risks across multiple stages of the pipeline.

- ***C5***: Specifically, it demonstrates component-level backdoor vulnerabilities that exploit under-audited code within projects prior to training, enabling highly evasive attacks.
- ***C6***: It further reveals vulnerabilities in the post-training stage, showing how an adversary can embed an evasive backdoor into a *benignly trained* model using an optimised trigger and switch mechanism.

  ***C4, C5, C6* have been submitted for publication.**
  Vyas, S., Caron, A., Hicks, C., Burnap, P., Mavroudis, V. (2025). Beyond Training-Time Poisoning: Component-level and Post-Training Backdoors in Deep Reinforcement Learning. AAAI 2026.
- ***C7***: Finally, this thesis leverages a hardware fault injection technique called Rowhammer to implement a backdoor attack at inference time using an optimised trigger, marking the first such demonstration in DRL.

  ***C7* is in preparation** for publication.
  Vyas, S., Caron, A., Hicks, C., Burnap, P., Mavroudis, V. (2025). InfRLhammer: Deep Reinforcement Learning Trojan Insertion Through Efficient Weight Replacement Attack.

1.6 Thesis Structure

This chapter has introduced the research area, outlined the research questions, and highlighted the key contributions of this work. The remainder of this thesis is structured as follows, aiming to establish the overarching research purpose and identify gaps in development, methodology, findings, discussion, and directions for future work.

- **Chapter 2: Background**
  - This chapter analyses the key challenges associated with deploying DRL in safety-critical systems. Using ACND as an illustrative application, it synthesises existing literature through a Requirements Table and identifies adversarial backdoors as a central security bottleneck. This analysis motivates the focus on DRL supply chain vulnerabilities addressed in subsequent chapters.
- **Chapter 3: A Review on Backdoor Attacks**
  - This chapter examines backdoor vulnerabilities in AI and DRL, establishing the foundation for the research contributions in Chapters 5, 6, and 7. It reviews

existing threat models, highlights technical gaps in current attack and defence strategies, and evaluates the robustness of a key defence [20] against a baseline attack [117]. By challenging its theoretical assumptions, this chapter demonstrates the defence's failure against a more advanced and realistic backdoor.

- **Chapter 4: Methodology**
  - This chapter provides the overall methodological framework that underpins the technical chapters of this thesis. Specifically, it outlines the experimental setup, evaluation metrics, and implementation details used to systematically assess the efficacy of DRL backdoor defences and attacks presented in the following chapters.
- **Chapter 5: Beyond Existing Defence**
  - Building from the the insights of Chapter 3, this chapter proposes a novel detection system targeting the most evasive backdoors (in-distribution triggers). Experimental results demonstrate the system's high effectiveness in detecting these threats in real-time.
- **Chapter 6: Exposing a Wider Threat Landscape**
  - While earlier chapters contributed to existing DRL backdoor attack and defence research, this chapter broadens the scope by introducing a more comprehensive threat model. It proposes two novel attacks with significantly reduced adversarial privileges. The first exploits component-level vulnerabilities prior to code creation, often overlooked by developers and audits. The second is a data-free, post-training attack on a benignly trained model that is computationally efficient. Experiments on standard Atari environments validate both attacks' effectiveness and their lower access requirements compared to prior work.
- **Chapter 7: Injecting Backdoors into DRL Models During Inference Through Rowhammer**
  - This chapter further extends the threat model by exploring hardware fault injections as a novel attack vector in DRL. To date, no existing works have investigated inference-time attacks or fault injection techniques in this domain. This chapter details the implementation of a Rowhammer-based backdoor attack and evaluates its performance through experiments and ablation studies on benchmark Atari domains. The results show high effectiveness while requiring even less adversarial access than the attacks introduced in the previous chapter.

- **Chapter 8: Limitations, Future Work, Conclusions**
  - This chapter summarises the thesis contributions on identifying and mitigating vulnerabilities in the DRL supply chain. It discusses key limitations such as the benchmark environments used for experimentations. The chapter also outlines directions for future research and considers the applicability of the findings to safety-critical domains such as ACND, highlighting new security verification capabilities introduced by this work.

Chapter 2

Towards the Deployment of DRL for Safety-Critical Systems: Autonomous Cyber Network Defence

2.1 Introduction

Given the promise of autonomous systems in both research and real-world applications, including healthcare, robotics, and structural and automotive engineering, the integration of sequential decision-making algorithms has the potential to automate a significant portion of complex decision-making tasks that currently require sustained human oversight. A particularly important domain for such systems is safety-critical applications, such as Autonomous Cyber Network Defence (ACND), where rapid and adaptive responses to evolving conditions are essential for maintaining enterprise network integrity and operational continuity.

This need is becoming increasingly urgent as governments and private sectors worldwide continue to transition essential operations toward fully digitised infrastructures, thereby expanding the attack surface available to cyber adversaries ranging from individual attackers to organised hostile states. This global digital transformation, combined with a growing shortage of cybersecurity expertise [48], has rendered many existing cyber defence mechanisms inadequate against novel and sophisticated attack strategies. Consequently, there is a pressing demand for the incorporation of advanced autonomous defence architectures and techniques [112, 149] across digital infrastructures, including enterprise networks and operational technology (OT) systems. While existing technical publications and white papers have explored autonomous defence solutions, their limitations in addressing realistic and adaptive cyber threats highlight the need for more structured and systematic research to

support reliable real-world deployment. In response, this chapter presents a comprehensive systematic literature review of ACND, an area concerned with autonomous decision-making agents for networked systems tasked with mitigating a wide range of current and emerging cyber threats. To provide a holistic perspective, the chapter formally defines ACND and analyses prior work across its major sub-domains, with particular attention to the decision-making paradigms employed. This analysis is synthesised through an overarching ACND Requirements Table 2.3 (Section 2.4.2), which highlights key objectives and design considerations for autonomous defence systems. Through this process, the chapter motivates the role of sequential decision-making approaches, demonstrating why DRL emerges as a particularly suitable framework for ACND due to its ability to reason over long-term consequences and adapt to evolving dynamics within complex enterprise networks.

Broader surveys on machine learning for cybersecurity, such as Buettner et al. [25], focus primarily on datasets and methods for intrusion detection systems. While these approaches contribute to automation in cybersecurity, they do not constitute ACND, as they lack an autonomous decision-making component capable of executing network attack or defence actions.

White papers further highlight this gap. Burke et al. [26] present a comprehensive exploration of potential projects within Active Cyber Defence (AcCD), proposing multiple ML-based abstractions for automated defence, attack, and security planning. Although several of these projects overlap with ACND, the white paper does not address the development of autonomous agent training environments or provide a comparative analysis of decision-making algorithms suitable for autonomous cyber operations. Similarly, Yu et al. [246] outline a Multidisciplinary University Research Initiative (MURI) for Adaptive Cyber Defence, emphasising adaptation and adversarial dynamics corresponding to autonomous blue and red teaming. While their framework aligns with the motivations of ACND, it does not consider the critical role of simulation and emulation platforms in enabling real-world deployment.

A notable exception is DARPA's Cyber Agents for Security Testing and Learning Environments (CASTLE) program [3], which explicitly integrates RL agents for autonomous defence, assessment, and purple teaming within realistic simulated and emulated network environments. CASTLE is structured around three parallel specialisations, namely red team RL, blue team RL, and purple teaming, and is organised into phased milestones targeting real-world deployment. This program closely aligns with the motivations of this chapter, and its phased structure informs the technical requirements analysis presented herein, particularly with respect to the development of networked environments and autonomous agents.

Collectively, this body of work highlights the need for a structured and streamlined technical pathway for the real-world implementation of robust sequential decision-making algorithms within ACND. To address this gap, the domain is conceptualised as comprising two concurrent research and development thrusts: the advancement of Autonomous Blue and Red Teaming sequential decision-making agents, and the parallel development of an Autonomous Cyber Operations Gym capable of supporting the training, evaluation, and transition to deployment of sequential decision-making agents. Formal definitions of these and related concepts are provided in Section 2.2.

Overall, this chapter makes the following analytical contributions:

- It formulates a structured analytical framework for ACND by organising the literature around the functional requirements of autonomous sequential decision-making agents. This framework, expressed through a Requirements Table, explicitly decomposes ACND into two interdependent components: the algorithmic development of Autonomous Blue and Red Teaming agents, and the system-level design requirements of Autonomous Cyber Operations Gyms required to support the training, evaluation, and deployment of the sequential decision-making algorithms.

- It critically evaluates existing ACND-related publications by mapping their methodological assumptions, algorithmic choices, and experimental settings onto the proposed Requirements Table. This analysis exposes systematic imbalances in current research, particularly the emphasis on agent-level learning in isolation from the networked environments necessary for realistic autonomous operation.

- It identifies previously underexplored and practically significant research gaps arising from the misalignment between autonomous decision-making algorithms and the environments in which they are trained and deployed. These gaps are analysed in terms of their implications for robustness, scalability, and real-world deployment, thereby delineating concrete challenges and guiding future research directions for the transition of sequential decision-making ACND from controlled benchmarks to operational networked systems.

This chapter is organised as follows: We first introduce important terms used frequently in this chapter (Section 2.2). Then, Section 2.3 addresses the research methodology utilised to find relevant ACND publications. Subsequently, Section 2.4 elaborates the curated terminology of ACND and its differentiation from similar terminologies used within recent literature. This section then provides the importance of sequential decision-making systems in ACND within key national strategy publications. Lastly, the section provides a comprehensive

Requirements Table that will be used to evaluate the selected publications recognised to be as part of ACND. Section 2.5 elaborates and critiques on the autonomous defence and attack (defined as autonomous blue and red team) agents in custom ACO Gyms through the ACND Requirements in Section 2.4. Section 2.6 elaborates an exhaustive list open-source and closed-source ACO Gyms and assesses them using the ACND Requirements in Section 2.4. Section 2.5 elaborates a list of published autonomous agents within ACO Gyms and evaluates them using the ACND Requirement in Section 2.4. Section 2.8 provides a discussion identifying the open research areas and their corresponding challenges within ACND literature using the analysis conducted in the previous sections. Lastly, Section 2.9 concludes the article by summarising the area of ACND.

2.2 Key Definitions

This article comprises of several technical terminologies that are commonly used within the fields of cybersecurity and artificial intelligence. This section will define the key terminologies used within this document.

Autonomous Red Teaming: Red Teaming is a technique used within military and industry operations to uncover networked system vulnerabilities or to find exploitable gaps in operational concepts, with the overall goal of reducing surprises, improving and ensuring the robustness of the networked system [40]. In the context of this chapter, autonomous red teaming refers to an autonomous agent possessing a set of operations (to uncover vulnerabilities and exploits within the networked system) as their action space. In the context of this chapter, the overall aim of autonomous red teaming is to ensure the robustness of the autonomous blue team agent (definition elaborated below) in terms of defending the system against known vulnerabilities and exploits.

Autonomous Blue Teaming: Blue Teaming is a technique responsible for defending a networked system by maintaining its security posture against a set of mock attackers that aim to exploit gaps and vulnerabilities of the networked system. Typically the Blue Team must defend against real or simulated attacks 1) over a significant period of time and 2) in a representative operational context (e.g., as part of an operational exercise)[1]. In the context of this chapter, Autonomous Blue Teaming refers to an autonomous agent possessing a set of operations as their action space to destroy malicious processes from entering the networked system through its nodes/endpoints.

Autonomous Cyber Operations Gym: Autonomous Cyber Operations (ACO) is concerned with the defence of computer systems and networks through autonomous decision-making

[1] https://csrc.nist.gov/glossary/term/blue_team

and action. It is particularly required where the deployment of security experts to cover every network and location is becoming increasingly untenable, and where systems cannot be reliably accessed by human defenders, either due to unreliable communication channels or adversary action. ACO Gyms are networked system environments that facilitate the use of autonomous red and blue teaming agents in order to further strengthen the networked systems of the future from ever-evolving cyber-attacks [218]. ACO Gyms aim to address and reduce the 'reality gap' of potential networked systems, used in [222] by combining learning on simulations with testing in a real environments.

Simulated Network: A Simulated Network is an ACO Gym (or a part of the ACO Gym's training-testing strategy) that is designed as a finite state machine. The creation is usually completed in the form of code that includes objects that correspond to the components, agents and actions within the simulated network. [162]

Emulated Network: An Emulated Network is an ACO Gym (or a part of the ACO Gym's training-testing strategy) that is designed through a group of virtual machines (or a docker container with several network drivers), which are used to create a computer networked system [162].

Sequential Response: Sequential response, or sequential decision-making refers to algorithms that take the dynamics of the world into consideration, thus delaying segments of the problem until it is solved [69]. It is a fundamental task faced by any intelligent agent in an extended interaction with its environment which demands a set of decisions that are concerned with short and long-term decisions in order to reach a state that acts as an overall target within the environment [142]. In the context of this chapter, sequential decision-making algorithms are considered in this chapter as Autonomous Blue and Red Teaming agents due to the complexity of the network that requires navigation before a target action is taken by the autonomous agent (e.g. launching an exploit in a host within a different subnet).

Single-step Response: Single-step response algorithm refers to decision-making actions that only focus on the short-term outcomes. For example, in temporal context, the algorithm at time *t(n)* will perform calculations solely for a solution at time *t(n + 1)*.

2.3 Review Methodology

A methodology inspired by [119] was implemented to find all relevant articles for this review. In order to interpret the overall definition of ACND and the research questions for this article, an initial set of white papers [26, 122, 164] from national and international government institutions and organisations (mentioned in section 2.4.1) were utilised. Backward snowballing [114] was utilised to further find relevant similar and relevant papers. These papers

addressed the need for autonomous response solutions in networked systems within a variety of different areas, allowing us to categorise areas where autonomous response could be utilised within the existing areas of ACND terminology, specifically, Autonomous Red and Blue Teaming.

2.3.1 Research Questions

To harness the concepts proposed for ACND, research questions were developed to establish a search strategy and utilise the scrutinised literature to delineate forthcoming research trajectories and challenges. Addressing the research questions articulated herein will equip future AI and Cyber Security researchers to undertake studies within ACND, pinpointing essential research gaps necessary for publishing significant work. This requires employing the most effective algorithms in optimal ACO Gyms, specifically targeting a research gap identified in this study. Such a strategy promises to significantly streamline research and development efforts within the field of ACND and enhance the broader landscape of cyber security. The research questions (RQs) are as follows:

- **RQ1**: What is the role of Autonomous Cyber Network Defence in the current and projected cyber landscape?
- **RQ2**: What are the most promising algorithmic approaches used in Autonomous Cyber Network Defence?
- **RQ3**: What are the most suitable environments in which better sequential-decision making algorithmic approaches could be developed?
- **RQ4**: What future research directions and challenges must be undertaken to enable the real-world deployment of sequential decision-making algorithms for Autonomous Cyber Network Defence?

The first research question (answered in Section 2.4.1) investigates the importance of ACND and its projected role in safeguarding networks. We will address this question by primarily reviewing government and funding agency strategy documents (highlighted in Table 2.2), as ACND is currently predominantly supported by state-sourced funding [2].

The next two research questions aim to investigate distinct but complementary aspects of ACND. RQ2 seeks to identify the most promising algorithmic approaches by surveying past works identified as ACND literature (in Section 2.5), while RQ3 explores the most

[2]https://www.thinkdigitalpartners.com/uncategorised/2022/09/29/the-22-billion-future-of-the-uks-cybersecurity-insights-for-suppliers/

suitable environments for the development of these advanced algorithmic approaches (in Section 2.6). It aims to understand where (in terms of infrastructure, technology, and support systems) these algorithms can be best developed and refined to enhance their effectiveness in real-world applications. By addressing these two aspects, researchers can contribute to building more resilient and adaptive systems that are capable of defending against the increasingly sophisticated landscape of cyber threats.

Finally, the fourth research question aims to map out the necessary research paths and challenges (shown in Section 2.8) that need addressing to enable the effective real-world application of ACND solutions. Deploying such advanced systems in actual operational environments goes beyond theoretical research and development. The outcome of this inquiry will provide a comprehensive blueprint for transitioning ACND from a research and development phase to full-scale operational deployment, thus closing the gap between theoretical possibilities and practical usability in defending against cyber threats.

2.3.2 Search Terminology Strategy

After identifying the initial set of research questions, the next step involves searching for relevant primary studies. As elaborated in this section, RQ4 was developed only after the initial research questions were answered. In order to optimise our search for relevant papers, popular digital libraries including IEEE, ACM Digital Library, Springer Science Direct and Google Scholar were utilised. A list of strings grouped within 3 overall themes of ACND were collectively identified (shown in Table 2.1 as themes **a**, **b** and **c**). The strings from all different overall themes are then grouped together in 3 different groups of permutation combinations as an aim to identify publications in digital libraries that:

- **i**: allow us to explore and rank the performance of identified algorithmic families in Autonomous Blue Teaming (**RQ1, RQ2**).
- **ii**: allow us to explore and rank the performance of identified algorithmic families in Autonomous Red Teaming (**RQ1, RQ2**).
- **iii**: allow us to discover the best possible environments in which the most suitable algorithms could be developed, trained and tested (**RQ1, RQ3**).

2.3.3 Overall Relevant Content Extraction

Due to the area of ACND gaining popularity only recently, backward snowballing [114] and forward snowballing [237] were conducted for several searches in order to find publications

and code repositories relevant to ACND that were not listed in the search strategy. For example, with areas such as "autonomous cyber operations gym" being a recently created terminology within this area, backward snowballing aided us to identify other popular publications (with respective implementations found in code repositories) that were created before this term was officially introduced. In addition, a manual search was conducted to identify the latest ACND related papers (along with papers highlighting further potential areas within the domain) that cited the publications identified through the search strategy. Through this search strategy, a total of 132 papers were shortlisted. The papers selected were passed through another screening process based on their abstract and conclusion in order to select the papers that align to the scope of ACND, reducing the selected relevant papers to 70. Lastly, the remaining papers were then fully read and analysed as further screening step, leading to 55 papers selected for this review overall. Figure 2.1 suggests the overall steps included within this search methodology.

2.4 Autonomous Cyber Network Defence

Autonomous Cyber Network Defence (ACND) is a topic that has recently been mentioned within a few publications and news articles over the last decade, in light of the increasing cyber-attacks that have occurred over the last few years. To define and interpret this term, a brief review was completed.

Rege et al. [194] provided a high-level description of ACND algorithms as a decision-making system with expert-level ability inspired by how humans reason and learn, citing a publication [18] producing an autonomous blue agent within a custom networked system. Ko et al. [120] provided a terminology for ACND when elaborating the purpose of the Defence Advanced Research Projects Agency (DARPA) grand challenge [3], where it described ACND as systems that are able to self-discover, prove, and correct software vulnerabilities in real-

[3]https://www.darpa.mil/program/cyber-grand-challenge

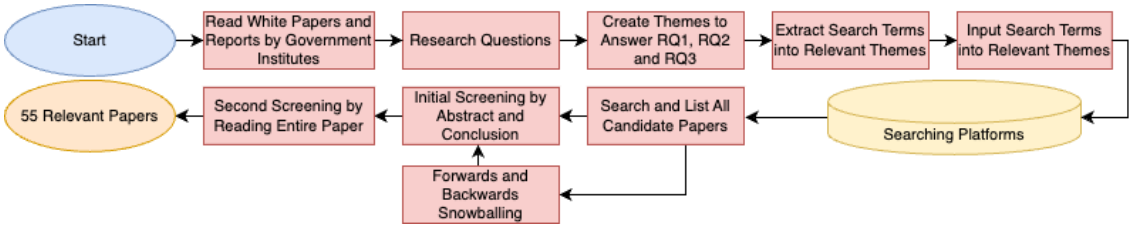


Fig. 2.1 A flow chart of the overall Autonomous Cyber Network Defence Research Methodology described in Section 3.

a. Algorithmic Approaches	b. Autonomous Blue-Teaming	c. Autonomous Red-Teaming
- "Artificial Intelligence" ‡	- "Autonomous Cyber Operations Gym"	- "Malware"
- "Machine Learning" OR "Deep Learning" ‡	- "Process Killing"	- "Process"
- "OpenAI" AND "Gym" ⋆	- "Cyber Defence" OR "Cyber Defense"	- "Penetration" AND "Testing"
- "Reinforcement Learning" ‡	- "Malware"	- "Offensive Cybersecurity"
- "Game Theory" ‡	- "Deception"	- "Autonomous Malware"
- "Generative Modelling" ⋆	- "Response"	- "Privilege Escalation"
- "Automated" OR "Automatic" OR "Autonomous" OR "Automation" ‡	- "Wargaming" OR "War-gaming"	- "Adversary Emulation"
- "Response" †	- "Cyber Resilience"	- "Wargaming" OR "War-gaming"
	- "Advanced Persistent Threats" OR "APT"	- "Red Team" OR "Red Teaming" OR "Red-teaming"
	- "Blue Team" OR "Blue-teaming" OR "Blue Teaming"	- "Reconnaissance"
	- "Cyber Threat Intelligence"	- "Autonomous Cyber Operations Gym"
		- "Cyber Defence" OR "Cyber Defense"
		- "Deception"

Table 2.1 This table outlines the overarching themes used for search terminology. **(a)** lists algorithmic approaches and terminologies that enable autonomous responses, which are essential for Autonomous Cyber Network Defence agents. **(b)** includes terms associated with Autonomous Blue Teaming, while **(c)** contains those related to Autonomous Red Teaming. Legend: ⋆ — The keyword has been individually combined with each term from column (a); † — The keyword has been individually combined with each term from column (b); ‡ — The keyword has been combined with all possible pairwise combinations of terms from both (a) and (b).

time without human intervention. In 2016, Baah et al. [13] provided a generalised overview of an ACND system. The paper described ACND as a response that begins with detection of an ongoing attack or an existing vulnerability in the network. The paper highlighted that speed and accuracy of detection are important in order to take action to mitigate threats before they can do damage to network assets or disrupt missions. It also illuminates a solution of machine learning analytics that can distinguish between suspicious and benign network activity, and automated fuzzing techniques that can discover previously unknown vulnerabilities in software. Benjamin et al. [18] define the ACND term through their project called Cognitive Support for Intelligent Survivability Management (CSISM), where the authors implement an Autonomous Cyber Network Defence decision-making mechanism with expert level ability. The ACND system observes and alerts the relevant users, and then takes defensive actions to ensure the survivability of the computing capability of the network. The authors realise that producing such an expert-level response in real-time with uncertain and incomplete information is a difficult target. However, they realise that there is a stepping-stone between the development of autonomous reasoning and learning through the use of cognitive architectures for cyber defence operations.

Burke et al. [26] from the Alan Turing Institute introduced a research initiative focusing on Active Cyber Defence (AcCD) through a white paper, which focuses on seeking increased automation within an enterprise to bolster network defenders and cybersecurity. Note, it is important to address the difference between the term AcCD and ACND lies in the inclusion of Automated Security Planners within AcCD, which are used to enhance *human decision-making*, while ACND strictly focuses on autonomous red and blue-teaming, primarily for the overall development of autonomous blue teaming agents. Overall, the paper explains that intelligent automation is essential to enable system defenders to manage the risk posed by highly autonomous future threats and attack, and defend the systems at cyber-relevant national scale. The white paper also elaborated the need for autonomous red and blue teaming. However, it only provided high-level information on the research directions within all areas without a further technical development pipeline. The use of Artificial Intelligence has been suggested within such systems as a way to intelligently understand the terrain (i.e., networked system) for detecting and responding to complex cyber-attacks with minimal errors.

Applebaum et al. [10] introduce the term Autonomous Cyber Network Defence in the context of tabular Q-learning, defining it as the use of ML to train agents that autonomously defend systems while minimising self-damage from noisy sensor data. While conceptually aligned with our view of ACND, their definition is limited in scope. It overlooks the parallel development of simulation-to-emulation environments such as ACO Gyms and omits the critical role of autonomous red agents. This paper extends the definition by incorporating both red and blue agents within an integrated training and evaluation framework.

The definition of Autonomous Cyber Operations (ACO) will also need to be addressed relative to ACND in order to clarify specific research directions within ACND as compared to ACO. Standen et al. [218] define ACO as the parallel development of autonomous red (attacker) and autonomous blue (defender) agents within a networked system that combat one another in a game-playing scenario. ACND differs from ACO through its focus on the overall development of autonomous blue agents, where autonomous red agents are particularly designed as an autonomous penetration testing agent facilitating holistic adversarial training. The development of ACO Gyms in the lens of ACND also differs to the development of ACO Gyms in that they must be designed to specifically for the development of autonomous blue teaming agents.

When compiling all the literature mentioned above, we interpret **Autonomous Cyber Network Defence** as a terminology focusing on the *autonomous decision-making agents* for cyber systems (such as enterprise network, industrial control systems) to mitigate highly complex cyber-attacks. The development of an ACND system could be conducted through a combination of different types of operations. This includes the development of autonomous

blue-teaming agents within ACO Gyms as a mode of terrain (to replicate real-world cyber systems), where autonomous red teaming agents are used to adversarially validate, develop and strengthen the autonomous blue team agents for an overall goal of their deployment within networked systems.

2.4.1 ACND Importance within National Strategy Documents

From our initial set of white papers and search strategy, we discovered that several government-based organisations have made it clear that AI will soon be forefront within cybersecurity in terms of detecting, responding to attacks within networked systems, along with creating autonomous attacks to discover vulnerabilities. Table 2.2 elaborates the importance of ACND within different countries and organisations, allowing us to decipher our first research question.

2.4.2 ACND Requirements

The North Atlantic Treaty Organisation (NATO) and US Army Research Laboratory outlined requirements for "Autonomous Cyber Agents" (also known as sequential decision-making agents) by producing a reference architecture and technical roadmap, Autonomous Intelligent Cyber-defence Agent (AICA) Reference Architecture (AICA) [122]. A specific part of the document focuses on the high-level strategic deployment and the ethical concerns on the battlefield of autonomous agents. A few requirements in AICA relevant for this chapter have been included in a domain-specific manner within the Requirements Table for ACND (Table 2.3) due to their relevance within defending digital infrastructures against cyber-attacks through autonomous defence agents. Compiling the literature utilised within the initially collected white papers, the table includes a structured format of compiled essential requirements of autonomous red and blue agents **(A)** along with ACO Gym requirements **(G)**, which will incorporate the usage of autonomous red and blue agents.

The requirements in this table are grouped into six key categories, each representing a critical sub-area that demands focused research attention. Effective *generalisation* within ACO Gyms and ACND algorithms will enhance system flexibility and robustness, enabling them to adapt naturally to changes in networked environments. Strategic *high-level decision-making* empowers agents to operate with structure, transparency, trustworthiness, and adaptability in complex, dynamic scenarios. Investigating diverse algorithmic *learning* approaches will help researchers evaluate and uncover more effective methods for training and deploying autonomous agents. Enhancing *multi-agent collaboration* in ACND will facilitate strategic

Country/Alliance	Department/Strategy	Reference to ACND
Australia	Department of Defence [217]	Suggests the need to expand cybersecurity skills and integrating AI into it. DoD is coordinating research and investment in AI capabilities to strengthen capability across the information and cyber domains.
	AI for Decision-Making Initiative 2022 [6]	Aims to develop 30 more AI-based challenges for researchers, including the TTCP CAGE Autonomous Cyber Network Defence Challenge to produce AI-based autonomous decision blue teaming algorithms for instantaneous response against cyber-attacks.
	Royal Air Force of Australia [54]	Advises continuous evaluation in which decisions can be made by machines and which must be made by humans.
Canada	National Cybersecurity Strategy [52]	Specifically mentioned the importance of defence and security applications with autonomous decision support
	Defence Research and Development [58]	The publication suggests that a combination of deep learning and RL algorithms for accurate identification of evolving threats, and then recommend or execute an appropriate course of action.
United Kingdom	Defence Artificial Intelligence Strategy [232]	Discusses the new risks from AI-Enhanced Cyber Threats which operate at speeds and at scales preventing actions by human operators in a timely manner.
	Government Cybersecurity Strategy [173]	Described AI as an emerging technology to focus on. Proposes to explore AI in the context of detecting malicious activity and in some cases to "enable autonomous response to threats"
NATO	Cooperative Cyber Defence Centre of Excellence [165]	Suggest the need for Nation States to adopt and explore AI-enabled Cyber Defence.
	NATO AI Strategy [164]	The strategy includes "collaboration on AI technologies for Cyber Defence.
United States of America	DARPA CASTLE[3]	A long term strategy to develop autonomous Red, Blue and Purple Teaming for algorithmic development of autonomous defence, autonomous attack and ACO Gyms.
	Army Research Laboratory [122]	Designed a reference architecture providing an outline on development of autonomous agents within ACND

Table 2.2 Overview of the National Strategy Papers on ACND

coordination among agents, strengthening defence capabilities and reducing the inherent asymmetry in cyber defence scenarios. To support progress, we advocate for open-source, well-documented *collaboration* across the ACND research community, helping to streamline and accelerate the broader deployment of ACND systems. Finally, ensuring agent *resilience* requires continuous exposure to a wide range of adversarial conditions throughout the training and deployment pipeline, fostering the development of more secure and reliable autonomous defenders. Overall, Table 2.3 contributes to ACND as a checklist for researchers to streamline their implementations and research contributions, which will expedite the eventual deployment of ACND operations within real-world networked systems.

Requirement	Summary
Generalisation	- **(G.1.1)** ACO Gym will need to generalise to new settings and have the ability to seamlessly add components [26] - **(G.1.2)** ACO Gym would need to be able to add different types of autonomous agents. [26] - **(G.1.3)** Networked system training-testing must promote transfer from simulation to a real world design, including aspects like matching real networked system latency operations delays within networked systems. Examples include a hybrid of simulation and emulation within training-testing strategies. [122] - **(G.1.4)** ACO Gym must have capability of scaling the network to larger sizes (additional subnets) without configuration issues [26] - **(A.1.1)** Autonomous agent will need to generalise their decisions relevant to the autonomous agent type it represents. [122] - **(A.1.2)** Autonomous agent will have to generalise and adapt to structural changes within the ACO Gyms (addition and removal of subnets and endpoints). [26] - **(A.1.3)** Autonomous red and blue agents must be designed to sustain their high performance from simulation to real-world deployment. [122]
High Level Decision-Making	- **(G.2.1)** ACO Gyms must be designed to explain their state after specific events occur within the networked system. [26] - **(G.2.2)** ACO Gyms will need to be framed into MDP/POMDP format in order to allow for effective autonomous decisions to be made [122]. - **(A.2.1)** For planning and collective response plans, sequential algorithms will need to be considered. [26] - **(A.2.2)** AICA reference architecture argues that both Game Theory and Artificial Intelligence would be suitable for implementation within ACND. [26] - **(A.2.3)** The designed autonomous agents will require a "deep" architecture to sustain their performance according to the complexity of the ACO Gyms. [3] - **(A.2.4)** Additionally, agents will need to be able to be explainable [187, 24, 115], i.e., justify their real-time decisions made in order for them to be operational within real-world networked systems.
Learning	- **(A.3.1)** AICA [122] opens up on the possibility of enabling continual learning within ACO Gyms. - **(A.3.2)** But also argues the importance of training-testing approaches.
Multi-agent Collaboration	- **(G.4.1)** ACO Gyms must be designed in a way to allow for multi-agent reinforcement learning (MARL) to operate. [122] - **(A.4.1)** Multi-Agent System representations would be required to train the autonomous agents and for action/strategy negotiation. [4]. AICA, combined with a MARL survey produced by [239], suggests utilising combinations of communication approaches and centralised training & Decentralising Execution solutions at a bare minimum. [122]
Research Collaboration	A requirement is the need to explain and collaborate with other researchers within AcCD [26] that coincides with ACND. Thus: - **(G.5.1)** ACO Gym must be open-source for researchers to contribute further to implementations. - **(G.5.2)** Documentation for ACO Gyms must be available for further development of gyms and ease of research and implementation of autonomous agents within them
Resilience	The AICA reference architecture highlights the need for resilience against differing malware samples and other algorithmic attacks. Therefore: - **(G.6.1)** ACO Gyms must be designed to allow for autonomous red agent to adversarially train the autonomous blue agent to reduce the number of incorrect actions. [26] - **(G.6.2)** ACO Gyms must be able to incorporate cyber-attacks and algorithmic attacks (e.g. backdoor attacks on DRL agents [247, 4]) plausibly curated by an adversarial insider. - **(A.6.1)** To improve performance of autonomous blue team agent (the sole purpose of ACND), adversarial training through an autonomous red agent must be encouraged. [26] - **(A.6.2)** Autonomous red agents must be provided with a wide variety of cyber-attacks (specified within the MITRE ATT&CK framework) [162] - **(A.6.3)** Autonomous red agents must be provided with a variety of algorithmic attacks [99] (such as adversarial examples) on the trained autonomous blue agents to address autonomous blue agent's algorithmic vulnerabilities. - **(A.6.4)** Autonomous blue and red agent must be able to launch deception defence and attacks respectively. [122]

Table 2.3 This table provides a list of Requirements for ACND to streamline its deployment within real networked systems.

2.5 ACND algorithms used within Custom ACO Gyms

As mentioned in the section 2.4.2, a typical ACND system comprises of a type of networked system, which possesses the provision to allow autonomous red and blue team game-playing scenarios. Recent publications within ACND have utilised autonomous decision-making algorithms such as Game Theory (GT), Machine Learning (ML) and RL for autonomous blue and red teaming within custom ACO Gyms. A comprehensive overview on the fundamentals of GT and RL can be found in [216] and [14] respectively.

ML-based solutions (along with RL-based solutions [208, 168, 195]) have also been utilised solely for quick incident and intrusion response over the years [82, 170, 228]. Specifically, Zago et al [249] utilise ML techniques to analyse, detect and react against existing and upcoming cyber threats, including botnets. The proposed approach combines unsupervised and supervised approaches to create a scalable detection and reaction framework willing to decrease the error rate as well as increasing the efficiency in terms of computational resources. The approach uses dimensionality reduction algorithms and uses publicly available datasets for intrusion detection for its implementation. While sole ML-based implementations like this allow the mitigation of specific types of attacks, they lack the ability to defend against sophisticated attacks that require a multi-step response.

An example of a threat that requires a multi-step response is a ransomware attack that has already spread partially through the network. Once the attack is detected, containment actions are first taken to prevent further spread. This step might involve disconnecting infected machines, applying network segmentation, or temporarily shutting down network access. Following from this, the focus shifts to removing the ransomware from all infected machines and restoring data from backups. This step requires careful planning to avoid reinfection and to ensure data integrity. A rapidly acting autonomous defence system can potentially address the threat sooner.

Additionally, like zero-sum GT-based solutions, their performance does not scale to larger enterprise networks due to the algorithms not being complex enough to generalise state spaces further away from the scenario in operation. Cam et al. [27] also highlight how most ML-based solutions (which include supervised and unsupervised learning algorithms) provide solutions to a single-step learning problem, a feature of the algorithm that makes it infeasible for implementing it as ACND-based solutions within networked systems. Therefore, the publications selected for this section focus on sequential response that is required for autonomous agent to stop cyber-attacks within an overall networked system.

The rest of this section provides an overview of the recent publications within autonomous response for blue and red teaming respectively within custom networked systems, and

analyses the publications based on their autonomous agents and custom ACO Gyms through the Requirements Table in section 2.4.2.

2.5.1 Autonomous Blue Team Solutions

The autonomous blue agent within a network system must be perpetually vigilant to defend the entire attacker surface in real-time, while the attacker only needs to succeed once within a single location. Due to this asymmetric scenario between cyber-attackers and defenders, the defenders with limited resources cannot afford to prepare for all possible attacks.

In this subsection, we focus on addressing Posture-related vulnerabilities (PrV), a concept introduced by Huang et al. [103] that highlights the inherent disadvantage faced by the blue team compared to the network attackers. Specifically, the blue team must continuously monitor and protect the entire attack surface from unauthorised access, while attackers only need to find and exploit a single vulnerability to succeed. Due to this disadvantage in security posture, a blue team with limited resources cannot afford to prepare for all possible attacks. Table 2.4 below evaluates the autonomous blue teaming publications along with their custom ACO Gyms.

	Autonomous Blue Team Custom Networked System Publications															
Requirements	[257]	[22]	[101]	[169]	[151]	[73]	[60]	[34]	[42]	[27]	[236]	[74]	[233]	[213]	[197]	
A.1.1	+			+	+	+	+	+	+	+	+	+		+	+	
A.1.2				+		+	+	+	+		+	+		+	+	
A.1.3								+		+	+					
A.2.1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
A.2.2	+			+	+	+	+	+	+	+	+	+	+	+	+	
A.2.3			+						+	+		+			+	
A.2.4	+	+														
A.3.1																
A.3.2	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
A.4.1					+		+									
A.6.1				+		+			+							
A.6.2	+	+														
A.6.3																
A.6.4											+	+				
G.1.1	+	+	+	+		+	+		+		+	+		+	+	
G.1.2			+	+		+	+		+					+	+	
G.1.3		+			+					+	+					
G.1.4	+	+	+	+		+	+		+			+		+	+	
G.2.1		+											+	+	+	
G.2.2	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	
G.4.1					+		+		+							
G.5.1													+			
G.5.2													+			
G.6.1			+	+			+		+							
G.6.2																

Table 2.4 Autonomous Blue Team Solutions within custom networked systems

Table 2.4 shows relevant ACND autonomous blue teaming publications within networked systems designed solely for their respective autonomous blue team agent implementations. The table highlights most publications meeting requirements A.1.1, A.1.2, A.2.1, A.2.2. This is specifically because most publications highlight the need for a sequential blue agent response [27], as opposed to single-shot blue agent responses that are not feasible to defend the systems against modern day cyber-attacks. This is further shown by all publications framing the problem as an MDP/POMDP (G.2.2), which allows autonomous agents to take sequential response through the transitioning of states, that signifies a combination of actions taken within specific nodes of a networked system. However, while the requirements of A.1.2 are met within the specific publications, they are simulation based networked system implementations, which means that the system does not completely represent the complexity of configuration changes of the real-world networked systems. This is specifically highlighted in A.1.3 requirement which is not met by most publications in Table 2.4 that only test their algorithms within simulated networked systems. Most publications did not meet A.2.3 that is required within complex networked environments for appropriate generalisation of long-term actions for the agent. Only DRL implementations were able to fill this requirement, making them more suitable. Dhir et al. [55] also suggested the use of Causal Inference algorithms [113, 182, 250, 196, 76] that could maintain their performance within ACO Gyms. Most publications in Table 2.4 also do not meet explainability requirement of A.2.4, which is essential for utilisation of any autonomous agents within Security Operations Centre (SOC) environments, in which such agents will need to be certified before they are in operation. Only 2 of the selected publications met A.4.1, in which both publications implemented autonomous response against specific cyber-attacks (i.e., DDoS, as opposed to an agent that could detect and respond to a variety of cyber-attacks). Such requirement is highlighted in the form of A.6.1 and A.6.2, which suggests the need to continually develop the knowledge base of the autonomous blue agent through adversarial training against a variety of cyber-attacks. Moreover, the lack of implementations that fill the A.6.1 requirement also hinders the development of autonomous blue agents against algorithmic attacks mentioned in A.6.3, an area in which no publications highlighted in Table 2.4 have implemented solutions for.

Requirement A.6.4 in the context of autonomous blue teaming refers to defender agents which have the capacity to strategically launch deceptive elements that enhance the defence of a networked system through an increase in threat detection functions. Applications of Cyber Deception in literature seek to integrate high-fidelity deceptive assets into existing infrastructures with the purpose to mislead or slow down adversaries and ultimately thwart their cognitive processes. These assets are typically encapsulated inside virtual environments that resemble their physical counterparts; and have two overall aims: first, the defence of a

system through the enhancement of threat detection functions such as lures and decoys, and second, the ability to misdirect and quarantine attackers to support the gathering of Cyber Threat Intelligence (CTI). Deception-based Cyber Defence (DCD) platforms counter classic attacker-defender asymmetries by executing and maintaining preventative cybersecurity tools that, unbeknown to an adversary, obfuscate the true security posture of a network. In fact, the use of DCD is becoming an increasingly prudent choice in the mitigation of PrV(s) on the account that adversaries must 'minesweep' through a sea of supposed vulnerabilities in order to execute a successful cyber-attack. Wang et al. [236] and Ghao et al. [74] both consider the notion of combining the use of intelligent algorithms with dynamic deployment strategies in order to analyse adversary behaviour. Both solutions succeed in training a blue agent to select optimal deployment strategies but fall short of many generalisation and resilience-based requirements due to the link between the attackers with the associated environment. As previously mentioned, solutions such as [74] which incorporate DRL typically meet the high-level decision-making requirement A.2.3. The use of DRL in this instance is sensible because the authors are aware of the impact that general attacker-defender scenarios have on the space complexity of typical RL algorithms. This is because Deep Neural Networks (DNNs) are introduced to make policy-based deployment decisions without the need to manually engineer the state space. In the context of ACND, determining a reward path through the trial and error of all possible states can often converge to computational intractability as the scale of the network environment grows; thus, by harnessing the predictive element of a DNN, knowledge becomes generalised by approximating each Q value rather than storing and looking up every distinct state. The authors in [74] utilise online learning to update defence models with newly collected attack information, although this is of a 'non-continual' variety, meaning continual learning techniques have not been implemented to address concerns regarding catastrophic interference, thereby failing to meet requirement A.3.1. Leveraging the approximations of DRL, Li et al. [132] proposes an optimal defensive deception framework by creating System Risk Graphs (SRG) which model adversary actions. The attack models are then used to train a DRL agent to generate optimal deployment strategies within micro-service architectures. Incorporating defensive deception into container-based cloud environments is sensible as, like the diversity and scale of typical OT networks, the virtualisation of technology and the dynamism of container services exposes a glut of additional attack vectors to an already overwhelming issue. Through the intelligent deployment of deceptive assets, the expanding threat surface can be maintained and prevented. The authors highlight the issue of scalability when modelling network environments and threat models as high-dimensional input spaces, implementing a DRL framework that scaled up to 60 nodes. In a different light, Walter et al. [233] draws attention to the prospect of augmenting ACND environments with defensive

cyber deception components by adapting the source code of an existing open-source ACO Gym called CyberBattleSim [223]. This paper falls short of many requirements as the solution does not necessarily create a dedicated blue agent. Instead, the aim of the paper was to gain insight by observing the impact of active cyber deception on attacker behaviours which can ultimately inform autonomous blue teaming agents.

In terms of the requirement of networked systems within the publications mentioned in Table 2.4, G.1.1 and G.1.2 were met within most simulated networked system publications. However, as mentioned previously, simulated systems do not represent the real-world systems accurately, hence the reason why very low number of mentioned implementations are able to meet the G.1.3 requirement. Similar to the requirement A.4.1, G.4.1 is an area in which networked systems will need to be developed in order to facilitate the inclusion of autonomous agents. Areas of research development also include G.4.1 and G.6.1, in which networked systems will need to be designed to allow such requirements. Overall, through our analysis, the most optimal autonomous blue teaming algorithm seen within literature is DRL due to it's ability to meet most algorithmic requirements in the ACND Requirements Table 2.3, answering our second research question. Using the analysis, it can be understood that many more requirements can be met by DRL algorithms, however, they (like all other algorithms) are highly dependent on the ACO Gym they are developed within.

2.5.2 Autonomous Red Team Solutions

The existing literature on autonomous red teaming solutions can be split into three categories: assistance to security analysts with attack planning, penetration testing or red teaming "automation", and red agent research conducted in gym environments. The later categories relate closely to ACO goals/objectives, whilst the former is an intermediary step towards it.

The attack path planning category utilises scanning information outputted from penetration testing tools such as Nmap or Nessus to design a POMDP (G.2.2) representing a corporate network. The Common Vulnerability Scoring System (CVSS) scores [5] from vulnerability scans are then utilised to define the transition probabilities. [72] also utilised the CVSS scores to inform the rewards (landing on the host as an administrator for instance). Researchers then utilise RL/DRL algorithms (A.2.2) on these environments to reach set objectives (while adding negative penalties at each step to avoid loops). For example, [72] and [41] utilised this approach to generate action plans to assist a human expert in reaching testing objectives with the DQN algorithm (A.2.3). Finally, it should be noted that tools

[5] https://nvd.nist.gov/vuln-metrics/cvss

such as Bloodhound [6] offer attack path planning focusing on Active Directory weaknesses, without utilising ML.
To automate penetration testing, one can extend the DRL game defined in the paragraph above to incorporate actions of penetration testing or red teaming tools (A.6.2). In fact, [255] did so to automate penetration testing with the Metasploit framework [7], whereas [148] utilised the PowerShell Empire framework [8] to automate post exploitation activities. Furthermore, researchers have analysed specific tasks of red teaming and attempted to automate them. For example, [123] automated privilege escalation through RL. One could envision multiple cells of the MITRE ATT&CK matrix [9] being automated in this fashion, such as defence evasion as seen in [64]. Overall, there is a need for a system to continuously evolve the autonomous red teaming agent to append new types of attacks within its action space. A combination of open-source red-teaming such as Atomic Red Team [10], ATTPwn [11], Infection Monkey [12] and APTSimulator [13] follow the MITRE ATT&CK matrix and can be used by DRL agents to execute specialised scanning and attack techniques.

Given that research into RL for autonomous red team solutions can be abstracted into simulated environments (described in further detail in ACO Gyms, G.1.3), the literature also comprises of such research (making it the most relevant algorithm for autonomous red teaming aswell). For example, [219] build DRL agents in the Network Attack Simulator Gym. The authors trained agents in five different scenarios of varied sizes and complexity, which were built with the PPO and DQN algorithms. They trained them on smaller scenarios to see how they performed in the larger ones at testing time, where PPO seemed to generalise slightly better. Given the exponential growth in action sets, researchers have begun analysing the use of Hierarchical RL in this setting, in fact [225] did so in the CyBORG Gym environment [218] where they proposed a Hierarchical DQN algorithm. Research in the open-source gyms are summarised through the list of requirements in Table 2.5.

Finally, it should be noted that GT Models (A.2.2) have also been explored (an example is provided by [45]), but in this case they are utilised to aid decision makers, such as in cyber war-gaming.

[6] https://github.com/BloodHoundAD/BloodHound
[7] https://www.metasploit.com/
[8] https://github.com/EmpireProject/Empire
[9] https://attack.mitre.org/matrices/enterprise/
[10] https://github.com/redcanaryco/atomic-red-team
[11] https://github.com/Telefonica/ATTPwn
[12] https://github.com/guardicore/monkey
[13] https://github.com/NextronSystems/APTSimulator

Autonomous Red Team Custom Networked System Publications			
Requirement	[148]	[123]	[72]
A.1.1			
A.1.2			
A.1.3		+	
A.2.1	+	+	+
A.2.2	+	+	+
A.2.3	+	+	+
A.2.4			
A.3.1			
A.3.2			
A.4.1			
A.6.1			
A.6.2			
A.6.3			
A.6.4			
G.1.1	+		+
G.1.2			
G.1.3	+	+	+
G.1.4	+		+
G.2.1			
G.2.2	+	+	+
G.4.1			
G.5.1			
G.5.2			
G.6.1			
G.6.2			

Table 2.5 Autonomous Red Team solutions within custom networked systems

2.6 Autonomous Cyber Operations Gym

As shown in the previous section, the lack of common open-source ACO Gyms prevent the possibility for an independent, accelerated development of sequential decision-making algorithms as autonomous blue and red agents (and ACO Gyms). This section aims to answer the third research question and provides a detailed overview of literature that have recently developed ACO Gyms along with the autonomous agents developed and published within literature and websites. Such ACO Gyms are simulated and/or emulated networked systems designed specifically for the development of sequential decision-making algorithms as autonomous blue and red team solutions. Given the availability of several resources, different publications have produced different strategies for training and testing environments, algorithm development type, and the types of cyber-attacks. That said, the emerging trends within this domain suggests the preference of Deep Reinforcement Learning approaches.

2.6.1 Training strategies

The most common approach to training and testing involves validating agents within the same environment used for training, whether simulated or emulated. This limits the ability to assess generalisation (i.e., requirements A.1.1, A.1.2 in Table 2.3) and prevents agents from fully leveraging the complementary strengths of different environments—scalability in simulation and realism in emulation—thereby falling short of requirement G.1.3.

Several research papers have strived to make progress in the domain of generalisation. For example, [219] built DRL agents in the Network Attack Simulator Gym [Schwartz]; a simulated environment to conduct research in autonomous penetration testing. Autonomous agents were trained in five different scenarios (encompassing subnets, hosts, vulnerabilities) of varied sizes and complexity, where the authors adopted both the PPO and DQN algorithms. After training the autonomous agents on scenarios of lower complexity, the impact on performance in larger complexity scenarios was experimented with, where the PPO provided superior generalisation. The cutting-edge platforms built to conduct research in ACND designed by [162], [218] or [133] all involve a simulated environment to train agents in a time efficient manner. In addition, emulations of the environment can be spun up on cloud providers with services running, actual malware performing malicious actions and autonomous blue agents with abilities to close ports or remove infections (mapping to the action spaces of the simulation). Another approach involves "real world" testing after training is performed in a simulated environment. One example worth mentioning are task specific agents, for example, [123] enumerated all possible privilege escalation techniques from the MITRE ATT&CK matrix and built an agent with DQN to perform this task. In order to speed

up the learning process, they trained their agent in simulated environment built with Python and then conducted their testing in the "real world" (a Windows Virtual Machine). They measured its performance based on how many steps were needed to escalate privileges, for some cases/vulnerabilities, the autonomous agent outperformed human experts.

2.6.2 Existing Autonomous Cyber Operations Gyms

For the acceleration of sequential decision-making algorithmic research within the domain of autonomous red and blue teaming agents within networked systems, open-source networked systems, or Autonomous Cyber Operations Gym (ACO Gyms) will be required. The provision of ACO Gyms will allow researchers to streamline their focus on meeting the autonomous agent based requirements in Table 2.3. In addition, this allows researchers to also focus on developing more open-source ACO Gyms that meet the networked system requirements in Table 2.3. Below is a review of existing environments which are designed for sequential decision-making algorithms used within cybersecurity research. The review begins with providing an overview of the existing open-source ACO Gym environments, and then delves into other closed-source emulated (and other simulated) ACO Gym environments that have been published within literature. Each part compares ACO Gyms amongst the other open-source/closed-source ACO Gyms using the ACND Requirements table (2.3) for ACO Gyms. Overall, this section provides a key analysis on how well sequential decision-making algorithms are/can be utilised within ACO Gyms.

Open-source Gyms

Firstly, The Cyber Battle Sim [223] (CBS) environment is created for training autonomous red agents that focus on the lateral movement phase of a cyber-attack in an environment that simulates a fixed network with configured vulnerabilities. The red agent utilises exploits (specific code that remotely accesses a network and gain elevated privileges, or move deeper into the network) for lateral movement while a pre-defined blue agent aims to detect the red agent and obstruct access. The CBS environment can define the network layout and the list of vulnerabilities with their associated nodes. In CBS, the modelled cyber assets capture OS versions with a focus to illustrate how the latest operating systems and up-to-date patches can deliver improved protections. The implementation can also be extended due to its design for autonomous blue agent training. In fact, [233] have implemented this by incorporating blue teaming deception into the environment. The developers ensured sufficient complexity exists in the environment to abstract the cells of the MITRE ATT&CK matrix for vulnerabilities (to be exploited by red agents to get rewards). Overall, the documentation is sufficient to

create new scenarios/networks, tweaking reward functions (values of compromised services and costs of exploitation) and adding vulnerabilities to services. While this allows users to extensively experiment with the environment, the code only exists for implementation within a simulated domain, thus, questioning the realism of the environment.

The Gym IDS Game [90] is a simplistic Markov game built upon the OpenAI gym environment. The attacker has two types of available actions:

- Reconnaissance action
- Attack of type 1...m

The defender also has two types of actions at his disposal:

- Monitoring action
- Defensive action of type 1...m

Different scenarios exist for either training a blue or red agent (or both). Unfortunately, the gym environment is overly simplistic and only provides a simulated environment, meaning that, like CBS, it also provides low realism. Similarly, to the Gym IDS Game described above, the Gym Threat Defence gym [153] is also a simulation-based system with a POMDP set-up. However, in this case, the authors have designed it as a purely defensive game where the defender has four different available actions.

- No action
- Blocking a service
- Disconnecting a machine
- Performing action 2 and 3 in parallel

One can define the probabilities of detection for each node, the attack probabilities, the spread probabilities, and the initial state.

Similar to the environments mentioned, the Optimal Intrusion Response Gym [91] is a Markov game built upon the OpenAI Gym libraries. The environment comprises of a simulated enterprise network with 6 subnets, with several hosts, each comprising of an IDS. Unfortunately, the game is overly simplistic for our use case as the defender can only select from two actions.

- ”Stop” will block the gateway. This will degrade the IT service and has a cost associated with it. However, it will also ensure the infection is contained.

- “Continue” is a non-action.

After doing some simulations/tests, [91] discovered that the blue agents they trained are more likely to “Stop" earlier when facing a stealthy attacker than against a noisier one.

The Network Attack Simulator environments [Schwartz], is purely built for training autonomous red agents (as there is no blue agent) to test AI systems in penetration testing tasks. This environment is built upon OpenAI gym and allows the ability to create scenarios by defining the number of hosts, services, the observability mode (fully observed for instance) and the asset criticality of the hosts in question. Finally, one can decide the vulnerabilities present on the network and define the cost of actions (cost of a subnet scan for instance). The red agent can select from seven different action types: Exploitation, Privilege escalation, Service scan, Operating system scan, Subnet scan, Process scan and No action. The goal of the project is to train red agents in performing penetration tests against simulated scenarios, while no blue agent interferes with the environment. Recently, the environment was extended to, NASimEmu, which included both simulation and emulation [108] [14]. The agent [15] that is developed within simulation can be seamlessly deployed within emulation. Novel inclusion within this include dynamic scenarios that represent prototypical situations, e.g., typical university or corporate networks. In these scenarios, some attributes are fixed (network topology, OSs, services and exploits), while some are left to change (network size and hosts' configuration).

The CyBORG environment [218] is designed specifically for training blue agents. However similarly to CBS, it can simply be extended for red teaming use cases. The environment allows training and testing in simulated and emulated environments respectively. The simulated environment comprises of an agent interacting with a scenario modelled in a finite state machine (FSM), in which each state represents systems and networks. An action satisfying a respective pre-condition is required to move from one state to another. The state also provides specific details such as the creation and deletion of individual files, or the making or breaking of network connections. All combined, an ideal training environment is generated for both the defender and adversarial agent. Once the autonomous agent is trained, it can be tested in the emulator, which comprises of AWS virtual machines to create a high fidelity cybersecurity environment in which the autonomous agent interacts with. The purpose of the environment is to act as a platform for research in ACND, whereby challenges are open to the public. Namely, the TTCP Cage Challenge 1, 2, 3 and 4. The challenges are enterprise network environments with ascending complexity (in terms of the observation and action space for

[14] https://github.com/jaromiru/NASimEmu

[15] https://github.com/jaromiru/NASimEmu-agents

the red and blue agent). In Table 2.6, all CybORG challenges have been added encapsulated into one column to address the overall contributions provided by the contributors.

In the TTCP CAGE Challenge 2, which is an extension of CC1, the action sets for the blue agent are exhaustive.

- Remove - removes malware from a host.
- Restore - if malware has elevated privileges it cannot be removed, and the host must be restored from backup (with a cost associated with it).
- Analyse - monitoring does not always detect infection (5/100 times) but performing an analysis on the host will always detect it.
- Decoy service - sets up a decoy service on a specific host to delay and detect red agent activity (there are 7 different services available).
- No action - Monitoring occurs regardless of other actions.

Scenarios can be defined in YAML files (i.e network topology and asset criticality). In addition, the project comes with varying red agents utilising different strategies. Finally, the documentation is exhaustive and details the high-level desired actions of an autonomous blue agent. On top of this simulated environment, CAGE Challenge 2 extends to an emulation (which is closed source), which can be spun up on AWS to validate the trained agents.

TTCP CAGE Challenge 3 [85] requires participants to develop autonomous defences for a network of drones, pre-compromised by malware during manufacturing, to establish a necessary communication network. The challenge is set within the CybORG environment, focusing on a scenario with 18 drones at constant risk from dormant firmware malware, operating in a 100x100 area with a 30-unit communication radius and a maximum 100-unit bandwidth. Teams alternate in discrete steps to achieve their aims, with the environment automatically providing offensive (red) and neutral (green) teams, and researchers guiding the defensive (blue) team. The green team, representing one agent per drone, simulates ground operative bandwidth demand, while blue and red teams vie for drone control, totaling 18 active agents. Drone movements and network structure, dictated by a randomised swarming algorithm, remain constant, allowing researchers to focus on combating malware through software command and control tactics as a distinct challenge. The reward function, accessible at every timestep through the standard OpenAI gym interface, motivates the creation of Multi-Agent Reinforcement Learning (MARL) agents, evaluating their defensive performance by averaging scores over 1000 episodes, each up to 500 steps. The optimal score in the challenge is 0, indicating flawless message delivery, with -9000 as the minimum, reflecting complete message failure for an episode.

A recently released TTCP CAGE Challenge 4, the network architecture is divided into four sub-networks, including two deployed networks, the Headquarters (HQ) network, and the Contractor network, all interconnected via the internet. The deployed networks are further segmented into two security zones - a restricted and an operational zone, whereas the HQ network is organised into three security zones: a Public Access Zone, an Admin Zone, and an Office Network. The Contractor network, in contrast, comprises a singular UAV control zone. To foster the creation of sophisticated agents, the composition of each security zone will be variable, with 1-6 servers and 3-10 user hosts, each equipped with 1 to 5 services, ensuring a dynamic and unpredictable environment. The network is defended by five network defenders (MARL): two per deployed network across security zones, one for the entire Headquarters, and none for the Contractor network, which remains undefended. The red team starts with access to the Contractor network, seeking to expand its reach. Red agents can multiply each turn either through opened phishing emails by the Green team or compromised service interactions, with a limit of one Red agent per zone capable of existing on multiple hosts. Although the blue team can eliminate Red's presence in a network, the red team retains a permanent foothold in the Contractor Network.

Yawning Titan [46] is a highly abstracted graph-based gym for training blue agents. The action spaces for both the blue and red agents do not map to realistic ones expected for cyber defence. Instead, it appears that the gym has been created to efficiently test and validate approaches/algorithms. The graph-based design also suggests it's true purpose is to explore computationally expensive approaches involving generalisation A.1.2 as networks can be defined as functions where the YAML file determines the behaviours and spaces. Table 2.6 has been used to summarise all open-source ACO gyms that can be experimented with.

Researchers from the KTH Royal Institute of Technology and DARPA have jointly developed an open-source platform named The Cyber Security Learning Environment (CSLE), as described in [172]. This framework features network simulation capabilities that facilitate the generation of Markov Decision Processes (MDPs) and enable the rapid learning of security strategies through the training of DRL algorithms for autonomous blue team operations. These strategies can be assessed within an emulated system that offers a realistic setting for evaluation without disrupting the workflow of the targeted system. CSLE includes comprehensive documentation for implementing autonomous blue team strategies within both simulated and emulated environments, enhancing its effectiveness for scalability and realism respectively.

Autonomous Cyber Operations Gym (Open-source)									
Requirement	CBS	GIG	GTD	OIR	CybORG	NaSim	YT	CSLE	PrimAITE
G.1.1							+	+	+
G.1.2				+	+			+	+
G.1.3					+			+	
G.1.4						+	+	+	+
G.2.1	+				+	+		+	+
G.2.2	+	+		+	+	+	+	+	+
G.4.1					+				
G.5.1	+	+	+	+	+	+	+	+	+
G.5.2	+	+		+	+	+	+	+	+
G.6.1					+			+	+
G.6.2									

Table 2.6 Open-source Autonomous Cyber Operation Gyms currently utilised by literature). We compare each gym environment through the Requirements Table.

Researchers from QinetiQ released PrimAITE [16] [17], which is an environment that provides the ability to model a customised networked system, while replicating real-world networked system intricacies (e.g., representation of connections, IP addresses, ports, OS's and services) in a way done by a static CybORG environment. The gym environment, made through OpenAI gym, is specifically incorporated to allow DRL functionalities as Autonomous Blue Teaming agents.

Closed-source Gyms

The rest of the ACO Gyms have been analysed in Table 2.7 through the ACND Requirements shown in Table 2.3. While the ACO Gyms highlighted are not open-source, they can provide important insights within the ACND community, particularly for researchers who can take inspiration when designing or making modifications to the existing ACO gyms. For example, no open-source ACO Gyms currently available have recognised the need of incorporating algorithmic cyber-attacks (G.6.2) within the action space of autonomous red agents. In addition, many closed-source gyms mention the need to scale the size of the network without configuration issues (G.1.4), an area which only one open-source gym

[16] https://github.com/Autonomous-Resilient-Cyber-Defence/PrimAITE

[17] https://www.qinetiq.com/en/news/qinetiq-releases-primaite-software-to-support-evolution-of-cyber-defence-agents

	Autonomous Cyber Operations Gym (Closed-source)									
Requirement	[71]	[155]	[70]	[23]	[199]	[200]	[133]	[162]	[62]	[3]
G.1.1	+		+	+	+			+		+
G.1.2			+		+	+	+	+	+	+
G.1.3		+		+	+	+	+	+	+	
G.1.4	+			+	+	+			+	+
G.2.1	+	+		+		+	+	+		
G.2.2						+	+	+		
G.4.1										
G.5.1										
G.5.2										+
G.6.1					+	+		+		
G.6.2								+		

Table 2.7 Closed-source Autonomous Cyber Operation Gyms currently utilised by literature. We compare each gym environment through the Requirements Table.

implements and emphasises on. This feature within ACO Gyms incorporates enhanced realism within networked systems as networks and hosts in a corporate environment are non-stationary. Lastly, closed-source environments like [133] have provided more comprehensive cyber-attacks using the MITRE ATT&CK framework for autonomous red agents, allowing more open-source gyms to implement the features within their environments. Overall, similar to open-source gyms, closed-source gyms also provide us with key developments and research areas within ACND and can be utilised to further enhance ACO Gyms in the future. Specifically, their novel implementations could be treated as an open problem for future ACO Gym creators, leading to incremental progress towards the realism of ACO Gyms.

2.6.3 Combined Analysis of all ACO Gyms

As shown in Table 2.6, most authors have recognised the requirement of the seamless addition and removal of nodes and components (G.1.1). Authors also meet the requirement of the adding autonomous agents (G.1.2) that are able to generalise their decisions along with understanding the structural changes within the ACO Gyms (A.1.1 and A.1.2 respectively). Moreover, all publications have also understood the requirement of AI-based sequential decision-making autonomous red and blue agents (A.2.1 and A.2.2 respectively), and have structured the ACO Gym as an MDP in order to facilitate such agents. However, while such ACO Gyms are highly scalable (G.1.4) and allow the development of relevant autonomous

agents, the environments utilised in all implementations are simulations of real networked systems, highlighting the lack of open-source emulated/real-world ACO Gyms (G.1.3). This results in the lack of "real-world" experience of autonomous agents, which will essential for utilisation within current networked systems.

While the rest of the analysis apply to those of autonomous agents, the design of the current state of the ACO Gyms could be used to assess the quality of autonomous agents that could be designed within the ACO Gyms. Overall, only one ACO Gym (CybORG Cage Challenge 3 [218, 85]) has recognised the need for autonomous multi-agent algorithms (A.4.1) as autonomous blue team solutions. Along with Cage Challenge 3, Malialis et al. [151] and Eghtesad et al.'s [60] publications (specifically focusing on using DRL for defending against DDoS attacks) environments could be a potential inspiration for structuring the ACO Gyms to facilitate multi-agent autonomous red and blue teaming collaboration (G.4.1). Very few ACO Gyms facilitate adversarial training (G.6.1 and A.6.1), which could potentially be utilised to strengthen the autonomous blue agent against a variety of cyber-attacks (A.6.2). In addition, no ACO Gyms currently open-source have recognised the need of incorporating algorithmic cyber-attacks (A.6.3) within the action space of autonomous red agents against autonomous blue agents. Inspiration can be taken from a closed-source ACO Gym [162] to incorporate algorithmic attacks such as evasion and poisoning of autonomous agents such as DRL algorithms. While the rest of the analysis apply to those of autonomous agents, the design of the current state of the ACO Gyms could be used to assess the quality of autonomous agents that could be designed within the ACO Gyms. Overall, only one ACO Gym (CybORG Cage Challenge 3 [218, 85]) has recognised the need for autonomous multi-agent algorithms (A.4.1) as autonomous blue team solutions. Along with Cage Challenge 3, Malialis et al. [151] and Eghtesad et al.'s [60] publications (specifically focusing on using DRL for defending against DDoS attacks) environments could be a potential inspiration for structuring the ACO Gyms to facilitate multi-agent autonomous red and blue teaming collaboration (G.4.1). Very few ACO Gyms facilitate adversarial training (G.6.1 and A.6.1), which could potentially be utilised to strengthen the autonomous blue agent against a variety of cyber-attacks (A.6.2). **Till now no ACO Gyms currently open-source have recognised the need of incorporating algorithmic cyber-attacks (A.6.3) within the action space of autonomous red agents against autonomous blue agents. Inspiration can be taken from a closed-source ACO Gym [162] to incorporate algorithmic attacks such as evasion and poisoning of autonomous agents such as DRL algorithms**. This constitutes a significant bottleneck within ACND, as algorithmic attacks have been extensively studied in supervised learning domains, requiring comparatively minimal additional effort for an adversary to adapt and construct analogous attacks against DRL-based systems.

2.6.4 Other Deployed Approaches

Several studies have focused on employing datasets and environments to enhance the detection and analysis of attacks. These environment-centric publications have been distinguished in this section from previous discussions, as they do not engage the use of wrappers for sequential decision-making algorithm frameworks, such as DRL. Such frameworks are crucial for autonomously addressing malicious alterations within the environment.

Researchers and engineers at Splunk created an open-source tool named Attack Range [18], designed for developing and testing the effectiveness of detection systems by simulating attacks in both cloud and local testbed environments. The detection development platform solves three challenges within the detection engineering domain, these include:

- The user being able to build a small lab infrastructure replicating a production environment
- Utilising attack simulation from different engines to generate highly realistic attack data
- Streamlined integration into Continuous Integration/Continuous Delivery pipeline to automate the detection rule testing process

The work therefore, allows the possibility to scale an implementation of a plethora of cyber attacks within the MITRE ATT&CK framework, and test the effectiveness of various detection methods. However, given the lack of autonomous response and sequential decision-making algorithmic frameworks implemented within this environment, the work lies outside the scope of ACND research.

Landauer et al. [125, 126] developed simulations of user attack scenarios and shared multiple labelled datasets to assess and compare the efficacy of Intrusion Detection Systems (IDSs) based on their detection accuracy. Moreover, the simulation includes a transformation engine that can automatically generate testbeds with capabilities for parallel operations. The creation of these testbeds involves a level of abstraction, enhancing reproducibility, flexibility, and usability. The datasets within these testbeds are structured to analyse multi-step attacks on a host, with each step of the attack being logged and labelled. Landauer et al. [124] have further enhanced the realism of their work by creating the Kyoushi Environment, a testbed that emulates a small enterprise network. This environment utilises complex state machines to mimic typical user activities and to introduce additional multi-step attacks. The data is automatically generated and labelled according to the configuration of the testbed.

[18] https://github.com/splunk/attack_range

The Kyoushi Environment is publicly accessible [19]. Although this testbed is one of the most effective setups for intrusion analysis, it currently lacks an autonomous response framework to counter multi-step attacks, thus relying heavily on numerous human operators to mitigate malicious threats within the network. Consequently, it remains outside the scope of ACND research. To date, no efforts have been made in this work to include wrappers or interfaces for integration with existing sequential decision-making frameworks such as DRL.

Chadha et al. [32] developed CyberVAN, a high-fidelity cyber environment specifically designed to counter evolving cyber threats. This tool is widely utilised by cybersecurity professionals for the effective evaluation and validation of cybersecurity technologies. CyberVAN offers a highly realistic representation of network environments, closely approximating the deployment of actual networks. It supports scalability, efficiently managing tens of thousands of varied cyber components such as hosts, routers, switches, firewalls, and communication infrastructures including Wi-Fi, LTE / 5G, and satellite networks. Additionally, CyberVAN is user-friendly, featuring advanced functionalities for the creation, implementation, and preservation of cyber scenarios used in experiments, operational planning, validation, and training. Despite its realism, CyberVAN relies on human analysts for threat mitigation and lacks the integration of sequential decision-making algorithms, thus not aligning with the scope of ACND.

As observed above, several projects explore the application of real-world approaches in DRL for cybersecurity research by developing flexible and abstract testbeds capable of adapting to various network environments. However, the current deployed state-of-the-art does not yet include the integration of sequential decision-making algorithms capable of quickly and universally detecting and mitigating multi-step attacks. Despite this, the methodologies used in developing real-world detection systems can inform the deployment of Autonomous Blue and Red Teaming algorithms within realistic networked systems, especially since their development is currently confined to simulation and emulation environments. That said, potential applications such as CSLE [172] and DARPA CASTLE [3] have successfully shown integration of DRL within their environments, paving the path for future researchers to make specific contributions. However, till now, *even real-world deployed approaches have not explored the issue of algorithmic adversarial attacks (A.6.3).*

2.7 ACND Algorithms within open-source ACO Gyms

Out of the open-source ACO Gyms mentioned in the previous section, several autonomous decision-making algorithms have been utilised for training and testing as autonomous agents.

[19] https://github.com/ait-aecid/kyoushi-environment

The ACO Gym creators and autonomous blue and red team agent developers have recognised the need for DRL-based solutions within the domain due to their nature of sequential response. While many of the requirements are met through the use of DRL-based solutions, this section suggests several gaps that still exist within the design of the autonomous agents through currently published implementations. Such gaps will require being met before the algorithms can be deployed into real-world networked systems for cybersecurity. Out of the current ACO Gyms, only two open-source ACO Gyms have been utilised in the publications of autonomous red and blue agents. In addition, many algorithms have been developed and are released open-source to promote research and development within the domain. CybORG [218] released four challenges with simulated networked systems with varying ACO Gym complexity in terms of the actions and observation spaces. The challenges focus on the development of autonomous blue agents, while the development of autonomous red agents (comprising of two different types of cyber-attacks) is also possible. NaSim [Schwartz] authors made their code open-source for the development of autonomous red agents and a few publications and implementations have utilised the simulated networks for the development of such agents.

2.7.1 Autonomous Blue Team Solutions

Of the two ACO Gyms discussed, CybORG has published results for its challenges [2], ranking the RL-based algorithms used in Cage Challenge 1 [84] and Cage Challenge 2 [83], with results from Cage Challenge 3 forthcoming [85]. These rankings are based on performance metrics defined by the organisers. A variety of teams employed different approaches and implemented diverse strategies through their autonomous agents. This article selects the highest-performing methods from these challenges and assesses them against the ACND requirements presented in Table 2.3.

From Cage Challenge 1, Team Mindrake [66] won the challenge and produced a Hierarchical RL algorithm that included proximal policy optimisation [202] with curiosity. The hierarchical [95] component of the algorithm is utilised through a controller to take relevant action according to the type of adversary that is deployed against the autonomous agent (B_line and Meander APT agent). Models are pretrained against both adversaries separately from the training phase and are then tested by the same adversaries at random episodes. The curiosity component allows exploration within the environment in the training phase via intrinsic reward [179], improving the reward achieved by nearly double. While the autonomous agent was victorious within the challenge, it does not meet the requirements A.1.3, A.2.4, A.3.1, A.4.1, A.6.3 and A.6.4. This is primarily due to the availability of the actions that could be taken amidst the two adversaries, along with the variety of attacks that could be conducted by the adversaries. Additionally, the environment [84] cannot facilitate

A.4.1. Similarly, the other three submissions also met the same requirements as the winners of the challenge. From Cage Challenge 2, the team from Cardiff University (with GitHub code [20]) won the challenge and also produced a Hierarchical PPO similar to Team Mindrake in Cage Challenge 1. However, the team utilised the availability of deception within the 2nd challenge through the selection of decoys (when required within the scenario) in a greedy manner. Using the ACND Requirements, the autonomous agent was not able to meet the requirements A.1.3, A.2.4, A.3.1, A.4.1 and A.6.3, but met the requirement of using deception due to its availability within Cage Challenge 2. Bates et al. [15] utilise Cage Challenge 2 to study the effectiveness of reward shaping and intrinsic agent curiousity on the performance of their autonomous blue agent. While the autonomous agent met the same requirements as the implementation above, the authors managed to improve sample efficiency, which is an area critical within ACND when applied to emulated domains. From Cage Challenge 3, Hicks et al. [96] won the challenge by utilising a MARL PPO with curriculum learning [17] to efficiently manage large action spaces, meeting the key requirement of multi-agent collaboration (A.4.1).

As shown in the first two challenges, variations of hierarchical PPO agents have shown most optimal performance (also suggested and algorithmically proven in [238]) as compared to other approaches. While the autonomous agents are able to generalise the moves of the two adversaries, the environment in which they were trained on did not comprise of many different types of cyber and algorithmic attacks (A.6.2, A.6.3) for the autonomous agents to generalise a greater pool of algorithmic attacks. To meet these requirements within this ACO Gym, future implementations could modify the ACO Gym to increase their cyber and algorithmic attack capabilities to assess the quality of generalisation of the autonomous agents against a greater pool of attacks. In contrast, no autonomous agent implementations in both challenges provided any form of explainability (A.2.4) regarding their incoming actions that they will take. The third Cage Challenge aimed at resolving the requirement of MARL agents (A.4.1), leading to several implementations submitted to this challenge using the algorithm to deal with the challenge of large action spaces. However, the environment (and hence, the agents) lacks the adversarial training of a variety of cyber and algorithmic attacks (A.6.2, A.6.3).

[20] https://github.com/john-cardiff/-cyborg-cage-2

2.7.2 Autonomous Red Team Solutions

Unfortunately, unlike for the Autonomous Blue Team Solutions, no public challenges have been proposed. As a result, research has been conducted in different gyms and under varying configurations. Therefore public comparable benchmarks are lacking.

Autonomous Red Teaming Solutions, as shown in Table 2.8 have so far largely been performed through Reinforcement Learning in ACO gym environments such as CyBORG [218], Network Attack Simulator [Schwartz] and CyberBattleSim [223], or in emulators or custom representations of IT networks. This intuitively makes sense as the problem is perfectly modelled for a Reinforcement Learning game (exploring a POMDP). Similarly to Autonomous Blue Teaming solutions, the Proximal Policy Optimisation algorithm has shown to be the most successful approach.

One example worth noting, involves research conducted in the CyBORG gym by [218] which presents the only known example of transferring a simulated red agents into an emulation. Researcher implemented DQN agents in the CyBORG simulator. They then validated the autonomous agents in the CyBORG emulator (G.1.3). Most of the autonomous agents successfully transferred to the emulator. Those which didn't likely failed due to over fitting to the observation in the simulator (moving from a discrete to continuous timed observations).
Another example from the Nasim gym, presents the first example of scaling generalisation (G.1.1) was conducted by [219]. They implemented Deep RL agents trained in small scenarios and validated on larger ones at testing time. Their research suggested that the Proximal Policy Optimisation algorithm seemed to generalise slightly better than other algorithms.

However, it remains an open-question if such algorithms are the most appropriate, indeed there appears to be a lack of research on casual approaches in Autonomous Red Teaming Solutions, even though these have recently been shown to be promising for the Blue Teaming side [9].

2.8 Discussion

The main purpose of this chapter is to identify an imminent research area for DRL-based sequential decision-making integration, ACND, within cybersecurity in order to mitigate cyber-attacks in the future. Autonomous response to cyber-attacks will need to be addressed through the research and development of autonomous red and blue teaming agents that are sequential in the nature of their decision making. The development of such algorithms could be accelerated through a parallel research and development within the area of ACO Gyms.

	Autonomous Red Team			
Papers	[225]	[218]	[167]	[219]
A.1.1				
A.1.2				
A.1.3		+		
A.2.1	+	+	+	+
A.2.2	+	+	+	+
A.2.3	+	+	+	+
A.2.4				
A.3.1				
A.3.2				
A.4.1				
A.6.1				
A.6.2				
A.6.3				
A.6.4				
Gym	CyBORG	CyBORG	Nasim	Nasim

Table 2.8 Autonomous Red Team solutions within open-source Gyms

While recent advancements have developed the research area in particular directions, more challenges have been identified using the ACND Requirements (on existing literature) in this chapter for the future development within the mentioned areas. Over 50 publications were analysed and compared through the ACND Requirements in Table 2.3. While the development of ACO Gyms and autonomous red and blue agent comprise of separate research and development strategies, the progress of one area is heavily dependent on the other, justifying the reasoning of having common research challenges. Since more challenges may exist in the specific requirement addressed, it is encouraged for researchers to build on this document to further address and develop areas within ACND that could further catalyse its development into industrial use.

The direct association of ACND Requirements in Table 2.3 with the publications identified as part of ACND has highlighted evident open problems and their corresponding challenges that must be addressed for the further development of ACND systems prior to their integration into real-world applications. This section answers the fourth research question and delineates the identified areas for further research and development, connecting them to the specific requirements outlined in the ACND Requirements.

2.8.1 AI-based Attack Robustification of Autonomous Blue Agents (A.6.3, G.6.1, A.6.1, G.6.2)

This area focuses on enhancing the robustness of DRL algorithms against poisoning and evasion attacks, which target the algorithmic functions of autonomous agents. Such attacks could originate from an insider adversary or a supply chain cyber-attack that alters the training code. To date, there has been scant research on these types of attacks targeting DRL algorithms. Nonetheless, it is clear that future cyber-attackers will likely exploit these methods through DRL and neural network-based research in various domains [209, 256, 11, 39, 12, 117, 50]. Although a few defences have managed to cleanse such poisoned models successfully [20, 89], they are prone to being bypassed by more sophisticated poisoning techniques [12]. Thus, the overarching challenges for this open problem include defending against algorithmic poisoning and evasion attacks in baseline AI environments, as well as the implementation and defence against these attacks within an ACO Gym, which would involve more advanced and context-specific AI attacks. If this open problem remains unaddressed, future networked systems may be at risk of algorithmic attacks that could seize control of autonomous blue agents and eventually, the entire network.

2.8.2 Continual evolution of action space for the Autonomous Red Agents (A.3.1, G.6.1, A.6.1, A.6.2, A.6.3)

Autonomous red agents action spaces are constantly evolving. Indeed, new services are often added which may have vulnerabilities tied to them. In addition, "every year new exploits are found for software and so in order to be useful Autonomous penetration testing agents will need to be able to handle a large growing database of exploits." [206]. Overall, this open problem aims to develop autonomous red teaming agents and ACO Gyms that can continuously add new types of cyber-attacks autonomously. While this open problem is reliant on other open problems *8.1 and 8.6*, the development and addition of cyber-attacks autonomous red agents based within a continual learning setting are yet to be explored. The

development of such system when utilising the current ACO Gyms requires the challenge of utilising different DRL algorithms that can continually add new sequential actions, while the challenge within ACO Gyms would be to convert the discrete configuration used in most gyms, into a continuous environment. Failure to implement on these challenges will keep the autonomous blue agent outdated from latest cyber and algorithmic attacks.

2.8.3 Explainable RL (A.2.4)

Explainable RL is more complicated than XAI, in fact "explainability for an RL agent, while clearly a subset of XAI and with similarities to IML (Interpretable ML), has distinct characteristics that requires its explicit separation from current XAI and IML research" [51]. Indeed, the first difficulty for XRL is due to the long-time horizons which determine the decisions/actions to take. The second one relates to the models not being built off labelled training data (which would simplify explainability). Therefore, this open problem currently relies on the development of AI research advancements, which can then permeate into the ACND domain. The challenge here involves the development of explainable and interpretable DRL algorithms within baseline AI environments, and then transferring their operations into ACO Gyms. Further inspiration could be taken from relevant survey papers and implementations [156, 212, 7, 146, 183, 78, 154, 224, 185, 175, 147]. Failure to address this challenge will lead to the autonomous blue agent not being certified by industrial employees within networked systems since the trust towards the agent will be low [147].

2.8.4 Multi-agent RL (G.4.1)

Another research area within autonomous blue teaming for ACND is the utilisation of multi-agent RL algorithms. This will be particularly more beneficial within enterprise networks environments which are highly complex. While [218] authors have proposed the implementation of multi-agent RL within their third and fourth Cage Challenge [21] [22], more research areas could emerge with increased research within this domain. Using single autonomous blue teaming agents will be useful, however, mistakes made by the agent within non-work hours will not be addressed unless there is another agent that evaluates the first agent and alerts it if a wrong decision is made.

[21] https://github.com/cage-challenge/cage-challenge-3
[22] https://github.com/cage-challenge/cage-challenge-4

2.8.5 Robustification of Deception Techniques in Autonomous Blue Agents (A.6.4)

It is highly important to highlight the necessity for research areas which utilise deception technology for ACND purposes. Their inclusion within ACO Gyms will allow the introduction more complex and proactive defensive deception techniques in order to study their effects in misdirecting and disrupting adversaries along the cyber kill chain. This is an open problem since existing literature rarely considers the complexity of this challenge, underlining the infancy of deception as a tool for ACND. Research that falls into this category [74, 236, 233] typically prioritise the use of honey-x methods [181] or 'lures' to analyse adversary behaviours through intelligent deployment strategies. This research challenge could make use of a useful framework for the challenge to encourage diversity within deceptive assets is the MITRE ENGAGE matrix, which identifies numerous deception techniques that can be leveraged at different areas of ACND to optimise adversary engagement [23]. Failure to address this challenge deflects from the key purpose of deception as adversaries can weaponise on the homogeneity of decoys and thus magnify the asymmetry that is ever-present between blue and red agents 2.5.

2.8.6 Realism of ACO Gyms (G.1.3, A.3.1, A.3.2, G.1.4, G.1.1, G.1.2)

Another open research area within the ACO gyms is the lack of realism of most of the environments that currently exist. A metric to classify the quality of the training-testing (or continual learning [96]) strategy as a research area is particularly important. Additionally, researchers generally would require building simulated environments and then transfer the learned policies to the real world (Sim-to-Real Transfer), this is often done in the case of robotics as pointed out by [254]. Environments such as CyBORG [218] attempt to address this challenge by supporting both simulation and emulation, however, both implementations comprise of areas which do not represent real networked systems (i.e., latency delays in simulation and network scalability in emulation). In addition, IT and OT networks, unlike traditional RL tasks, are continual and ever-changing environments which contrasts with most RL tasks. Moreover, networks and hosts in a corporate environment are non-stationary, whereas video games in which RL have been used would not expect an agent to perform well on an entirely new map [104, 19, 229]. Lastly, it should be noted that some networked systems like enterprise networks are multi-party network, which have a hierarchy of access levels depending on the user, future ACO Gyms should focus on designing such systems which incorporate this. The challenge here looks at developing new ACO Gyms to aid

[23] https://engage.mitre.org/

continual learning, an everchanging configuration and incorporating access restriction while training the DRL agents to maintain generalisability [176]. Such issues must be addressed, else the agent will not recognise the environment when implemented within real-world networked systems, leading to unwarranted actions being taken.

2.8.7 Realism of Deception Techniques (A.6.4)

Deception fidelity is often overlooked and introduced as a part of a constraint or assumption in current literature. As virtualisation of physical assets becomes more commonplace in context of network emulation, the implementation of Deception-based Cyber Defence (DCD) platforms must have the capability to model and simulate physical processes to maintain system fidelity and not alert attackers of its use. However, it is difficult to strike a balance between system fidelity and a sizable attack surface, particularly when considering the complexity and scale of some networked systems such as corporate networked system and OT environments, where researchers must find methods to emulate devices in convincing ways without replicating the network in its entirety. This open problem requires new methods for creating decoy profiles for assets which embody the attributes of the network component. To solve this challenge, researchers can also look to deceptive techniques which already consider or enhance the fidelity of integrated-lures. 'Honeyshills' [97] are an example as they use real components or systems and configure them to communicate with decoys to further give the impression of realism. These encourage suggestions for scaling deception methods within simulation-based networks and ultimately the move towards the emulated domain. Failure to address these challenges may result in the exposure of deception to the attacker, nullifying the precedence of deception over an attacker's inadvertence to its use. Such a contradiction cancels-out the symmetric advantage that's provided by correctly implementing deception technology.

2.8.8 Impact of Incorrect Action (G.6.1, G.1.3) [68]

The issue of incorrect actions also leads to a wide open research gap within the ACND literature for autonomous decision-making agents. The impact of such actions could lead of a plethora of issues within a corporate organisation. Examples range from minor actions such as blocking benign user hosts from joining the network to major actions such as the deletion of mission-critical documents conducted due to a lack of data diversity within the data used for training the autonomous blue agents. Therefore, research challenges include appropriate evaluation and metrics for the maximal reduction of "damage control" done by the agent. Additionally, explainable approaches [156, 212, 98] must be prioritised for superior forensic

evaluation of the autonomous agents. Not addressing this area will result in the autonomous blue team agent potentially eliminating important processes within the network, which could lead to high monetary losses.

2.8.9 Action and Observation Spaces (G.2.1 ,G.2.2, A.2.3)

Existing research in ACND significantly reduces the action and observation spaces by abstracting the action spaces to a point where they may no longer be usable in the "real world". Indeed, in a cybersecurity setting where agents may be deployed on thousands of hosts (in a single corporate network), each with huge action sets (kill any process, move/quarantine any file, change any firewall setting etc.) and essentially a continuous observation space, it would be challenging to sufficiently explore the space in training. This challenge applies to autonomous red agents also as "applying conventional DRL to automate penetration testing would be difficult and unstable as the action space can explode to thousands even for relatively small scenarios" where "each action in autonomous penetration testing can have very different effects such as attacking hosts in different subnets or different method of exploits" [225].

2.8.10 Development of new ACO Gyms (G)

In the current landscape of ACND, the availability of open-source ACO (Autonomous Cyber Operations) Gyms for researchers to test their autonomous blue and red team agents is markedly limited. This scarcity presents an open problem, urging more collaboration among AI and Cybersecurity professionals to propel advancements in ACND by developing new ACO Gyms. The challenges associated with this open problem involve gaining proficiency with the OpenAI Gym framework and leveraging the foundational code present in existing ACO Gyms as a basis for further development. Additionally, researchers are encouraged to utilise the Table 2.3 and the open problems above for constructing the networked system and incorporating the suggested research enhancements for the ACO Gym outlined in both the table and the open problems. Inspiration could also be taken from the existing OpenAI benchmark games and custom Gyms used for different applications. Researchers can also make use of the Kyoushi Environment and incorporate host-based datasets for every machine as a way to improve realism of ACO Gym development.

2.9 Conclusion and Thesis Scope

This chapter advances the understanding of a key DRL application, Autonomous Cyber Network Defence (ACND), by elucidating its terminology through research articles, government strategic reports, and cybersecurity training organisations. This clarification of terms facilitated the identification of specific ACND sub-areas, namely, Autonomous (Blue and Red) Agents and Autonomous Cyber Operations (ACO) Gyms, thus guiding the creation of ACND Requirements, a set of criteria used to evaluate the relevant literature. Through an extensive literature review on autonomous blue and red teaming algorithms within ACO Gyms it was uncovered that Deep Reinforcement Learning (DRL) so far has outperformed Game Theoretic and conventional Machine Learning approaches. DRL's advantage lies in its ability to handle sequential decision-making for achieving short-term and long-term objectives. Moreover, an in-depth assessment of both open and closed-source gyms, along with their implementations of autonomous teaming, was conducted. These evaluations, guided by the ACND Requirements, pinpointed areas ripe for further research.

To leverage DRL's capabilities in practical cybersecurity applications, further advancements are necessary in autonomous agent technologies and ACO Gym environments. Our findings have pinpointed specific challenges and gaps in the current field, including improving the robustness of defences against autonomous blue agents, enhancing the realism of ACO Gyms, minimising the repercussions of erroneous actions, and refining ACO Gym designs. Additionally, critical issues with DRL defenders need addressing, such as safeguarding against adversarial policies targeting blue agents, enhancing the explainability of blue agents, and refining multi-agent systems. Tackling these unresolved problems is vital for the progression of autonomous agents from controlled simulations to real-world networked environments, ultimately steering future research and development efforts in ACND.

Overall, these ten gaps and challenges are critical to the successful deployment of DRL within ACND as an application domain. A dominant challenge, encompassing eight of the ten identified gaps, lies in developing a principled understanding of the ACND problem space through appropriate Markov Decision Process (MDP) formulations and the corresponding design of DRL algorithms. While limited progress has been made toward addressing Requirement A.6.4 [92], *no existing work, at the time of writing this thesis, has directly addressed* ***DRL-based backdoor attacks*** *associated with Requirement A.6.3*. This gap is largely attributable to the fact that research in this area remains underdeveloped even within standard benchmark DRL environments. Given the severity of the threats posed by Requirement A.6.3 [162] in real-world ACND settings, foundational investigation within benchmark domains is a necessary prerequisite for meaningful progress in operational environments.

Accordingly, **this thesis contributes toward addressing the gap of *AI-based Attack Robustification of Deep Reinforcement Learning Agents*, with a specific focus on backdoor vulnerabilities in DRL systems (Requirement A.6.3)**. Existing literature has largely confined the study of DRL backdoors to traditional benchmark environments such as Atari. **This thesis extends that body of work by introducing novel and realistic attack and defence methodologies that have not previously been explored.**

By advancing the study of DRL security within controlled experimental settings, this work establishes a foundation upon which future research can translate these contributions to simulated Autonomous Cyber Operations Gyms, thereby supporting progress toward Requirement G.6.2.

Chapter 3

Backdoor Vulnerabilities in Deep Reinforcement Learning: A Survey and Case Analysis

3.1 Introduction

A key bottleneck to the real-world deployment of DRL agents lies in their supply chain security. While adversarial examples have been extensively studied within the DRL community, leading to the development of appropriate robust defence mechanisms [251, 79, 106, 245, 30, 220, 63, 136], comparatively limited attention has been directed towards backdoor attacks and data poisoning in DRL. As a result, defences against backdoor attacks are still underdeveloped.

In order to support the practical deployment of DRL agents for applications such as ACND, it is essential to strengthen the security of the DRL supply chain and increase awareness of the adversarial threat landscape (Requirement A.6.3 in Table 2.3). Foundational research in this area should first be conducted using standard DRL benchmarking environments, such as Atari, before transitioning to more complex and domain-specific simulations relevant to ACND.

This chapter begins with a high level overview of adversarial attacks and then delves deeper into backdoor attacks along with their formal characterisation within the broader field of AI. It then extends this formalism to the domain of DRL, offering insights into the nature of existing DRL backdoor threats and the corresponding defence strategies. Following this, the chapter presents an experimental evaluation of a prominent DRL backdoor defence proposed by authors from [20], critically examining its theoretical assumptions and broader

applicability within the field. Finally, the chapter concludes by outlining the high-level scope of the thesis and contextualising the technical contributions presented in the following chapters.

Overall, this chapter answers the second research question, **RQ2**, and makes the second contribution, **C2**, by critically evaluating a state of the art defence through experimental assessment of its theoretical limitations.

3.2 Reinforcement Learning

Reinforcement Learning (RL) is a subset of machine learning focused on teaching agents to attain an "optimal policy" for maximum performance in a given environment through trial and error. This method rewards or penalizes actions based on their outcomes, a strategy Sutton and Barto [221] term as "hedonistic" for its focus on maximizing environmental signals. The advent of Deep Reinforcement Learning (DRL) has markedly advanced RL agent capabilities by combining RL's strategic decision-making with deep learning's representation prowess. DRL enables agents to learn intricate policies for decision-making through environmental interaction, effectively mapping states to actions to optimise long-term rewards. Whereas traditional RL approaches like Monte-Carlo or tabular Q-Learning excel in achieving optimal behaviour, they often lack computational efficiency and struggle with extensive state and action spaces. Conversely, DRL demonstrates its robust potential in managing complex challenges, from gaming to robotics, as showcased in groundbreaking efforts like Mnih et al.'s Deep Q-Network (DQN) [158], marking a significant evolution in the discipline.

3.2.1 Proximal Policy Optimisation

PPO is a popular policy gradient method [243] that builds on the policy gradient framework. It refines the Trust Region Policy Optimisation (TRPO) algorithm [201] by simplifying it while retaining its efficiency. In policy gradient methods, the gradient of the objective function guides policy improvement. This objective (depends on policy π and parameters θ) optimises the expected rewards across trajectories (Equation 3.1) and leverages the advantage function to assess action benefits (Equation 3.2) within the environment it is operated on $\mathscr{E}$.

$$J(\pi, \mathscr{E}, \theta) = \mathbb{E}_{\tau \sim \pi_\theta} R(\tau) = \sum_{\tau} P(\tau; \theta) R(\tau) \tag{3.1}$$

$$\nabla_\theta J(\pi, \mathscr{E}, \theta) = \mathbb{E}\pi_0[\nabla\theta \log \pi_\theta(s,a) A_{\pi_\theta}(s)] \tag{3.2}$$

PPO ensures moderate policy updates using an actor-critic structure. The actor selects actions, while the critic evaluates them, facilitating balanced updates (Equation 3.3). The critic's evaluations help refine the actor's decisions, promoting a more effective and efficient learning process. This streamlined approach underscores PPO's adaptability and performance across diverse RL applications.

$$L_t(\theta) = \hat{\mathbb{E}}_t[-c_1 L_t^{VF}(\theta) + c_2 S\pi_\theta], \tag{3.3}$$

3.3 Adversarial Threat Landscape In DRL

Adversarial threats against RL and DRL agents are best positioned in a lifecycle taxonomy that distinguishes (i) inference-time evasion, (ii) training-time poisoning, and (iii) supply-chain compromises, with backdoors sitting at the intersection of (ii) and (iii) but behaving qualitatively differently from transient evasion. A representative entry point is inference-time observation manipulation, where small, carefully crafted perturbations induce incorrect action selection and rapid reward degradation in deep policy and value-based agents [102, 139, 16]. Lin et al. further refine this class by contrasting "always-on" (uniform) perturbations with strategically-timed attacks that perturb only a small subset of steps to reduce detectability, and enchanting attacks that steer the agent toward an adversary-chosen trajectory state [139]. Complementing evasion, a large training-time threat surface exists in which an attacker shapes what the agent learns by corrupting the learning signal or interaction process, for example through reward poisoning or environment poisoning. Zhang et al. formalise reward-poisoning as bounded perturbations to the reward stream and show that adaptive attackers, which condition their manipulations on the learner's progress, can be substantially more efficient than non-adaptive ones [253]. Rakhsha et al. broaden this view by showing that poisoning rewards and transition dynamics can be cast as "policy teaching," where an attacker forces the victim to learn an attacker-specified target policy under cost and stealth constraints [188]. Xu et al. further demonstrate that environment poisoning can transfer across settings, including black-box regimes where the attacker has limited knowledge of the victim learner [240]. On the defence side, the same taxonomy helps distinguish approaches aimed at improving robustness to evasion (for example, adversarially trained or disturbance-aware policies [184]) from approaches that attempt to provide explicit robustness guarantees against adversarial state perturbations [145].

Within this taxonomy, backdoor (also known as trojan) attacks are particularly dangerous because they are not primarily "noise at inference" but a conditional policy compromise that can remain invisible under standard validation while enabling reliable, attacker-controlled

behaviour when a trigger condition appears. This mirrors the classic supervised-learning backdoor supply-chain risk identified by BadNets, where models retain strong clean performance yet misbehave on trigger-bearing inputs [87], including stealthier variants where poisoned samples remain label-consistent and hard to spot by inspection or simple filtering [226]. DRL backdoors amplify these concerns through sequential effects: once triggered, a compromised policy can execute coherent multi-step deviations (not isolated wrong actions), potentially persisting across many timesteps and creating delayed, non-local failures. TrojDRL demonstrates that DRL trojans can be implanted with extremely low poisoning rates and in-band reward manipulation while preserving benign behaviour on normal inputs, and also shows that defences designed for classification do not transfer cleanly to RL settings [117]. BackdooRL extends the threat model toward more complex multi-agent or competitive RL and explores triggering mechanisms that do not require arbitrary modification of the victim's raw observations, which broadens feasibility beyond simple Atari-like assumptions [235]. Offline RL introduces an additional supply-chain risk because policies are trained from shared datasets; BAFFLE shows that poisoning offline datasets can implant persistent backdoors that survive fine-tuning on clean data and remain difficult to detect with common defences [80]. Finally, BadRL highlights the operational attractiveness of backdoors by demonstrating sparse, targeted poisoning strategies that maintain clean performance while reducing the attacker's footprint and increasing stealth [50].*Given the severity of backdoor attacks and the limited progress on effective defence strategies, this thesis seeks to strengthen the secure deployment of DRL and prevent such vulnerabilities from becoming a bottleneck in potential application domains such as ACND.*

3.4 Backdoor Attacks in AI

Backdoor attacks in AI represent a class of integrity violations, typically introduced via data poisoning and adversarial manipulation. In such attacks, a model is intentionally altered to exhibit malicious behaviour only under specific conditions, while maintaining normal performance otherwise. These attacks embed a covert functionality, known as a *backdoor* behaviour, which remains inactive unless a carefully crafted *trigger* is present in the input.

The adversarial modifications associated with backdoor attacks frequently evade detection during standard validation, testing, and deployment, rendering the malicious behaviour invisible to developers, system operators, and end-users alike. This stealth, coupled with the difficulty of detection using conventional evaluation protocols, renders backdoor attacks particularly dangerous.

Unlike adversarial example attacks that introduce minor perturbations at inference time, backdoor attacks typically arise through the injection of poisoned data during training. These poisoned samples induce conditional behaviour mappings within the model that do not align with the natural data distribution. Importantly, such mappings remain dormant until activated by the trigger, thus bypassing traditional performance metrics.

Backdoors are especially insidious as they exploit the overparameterisation and generalisation capabilities of deep neural networks. Even a relatively small proportion of malicious training data (often less than 5%) can be sufficient to embed robust trigger-response mechanisms. Triggers may vary widely, ranging from visual artefacts, to enterprise network traffic patterns, or even logical feature combinations. An effective backdoor prioritises stealth and high precision of activation.

Notably, [87] introduced one of the first practical demonstrations of backdoor attacks in deep neural networks, where the insertion of a small set of maliciously modified training samples resulted in models that behaved normally on clean data but misclassified inputs containing a specific trigger. While this was demonstrated in the context of supervised learning, similar vulnerabilities are applicable to DRL, which heavily relies on deep neural networks.

3.5 AI Backdoor Attack Formalism

Formally, let s be a clean input and β be a trigger perturbation. The triggered input $\tilde{s}$ is defined as:

$$\tilde{s} := s + \beta \tag{3.4}$$

The adversary $\mathscr{A}$ introduces the trigger into the input data through a targeted transformation:

$$\mathscr{A}(s, m, \Delta) = (1 - m) \circ s + m \circ \Delta \tag{3.5}$$

Here, m is a binary mask matrix with values in $\{0,1\}$, specifying which elements of the input should be altered, and Δ represents the trigger content to be injected. The Hadamard product $\circ$ ensures that only selected parts of the input are modified, enabling precise and sparse poisoning.

The attack's success is typically assessed by the drop in performance when the trigger is present, while maintaining high performance on clean data. Let $J(\pi, \mathscr{E}, \theta)$ denote the expected output quality of a model π in environment $\mathscr{E}$ with parameters θ. The adversary's goal is to induce a significant divergence in the model's performance when a backdoored

input is provided:

$$J(\pi, \mathscr{E}, \theta) - J(\tilde{\pi}, \tilde{\mathscr{E}}, \theta) \gg \varepsilon \tag{3.6}$$

In this formulation, $\tilde{\pi}$ and $\tilde{\mathscr{E}}$ represent the model and environment under the presence of the trigger, and ε quantifies the adversarial impact threshold. The model must therefore simultaneously exhibit benign behaviour under normal inputs and targeted misbehaviour under the adversary's chosen trigger. This represents a dual requirement that defines the essence of a successful backdoor.

3.5.1 DRL Backdoor Attack Formalism

To formalise the objective of a DRL backdoor attack, we expand the overall backdoor attack formalism above and cater it to DRL in terms of the attacker's objective.

Let π^* denote a normally-trained (benign) policy, which serves as the baseline. The expected reward for a policy π in a given environment $\mathscr{E}$ is defined as:

$$R(\pi, \mathscr{E}) = \mathbb{E}_{\mathscr{T} \sim p(\mathscr{T}|\pi, \mathscr{E})} \left[\sum_t r(s_t, a_t) \right] \tag{3.7}$$

The attacker aims to learn a backdoored policy π_e such that its performance in the clean environment $\mathscr{E}$ is nearly indistinguishable from the baseline policy. Formally, this is captured as:

$$|R(\pi^*, \mathscr{E}) - R(\pi_e, \mathscr{E})| < \varepsilon_1 \tag{3.8}$$

The second part of the objective concerns the poisoned environment $\tilde{\mathscr{E}}$, in which the trigger is present. In this case, the attack aims to maximise the discrepancy in expected reward between the standard and backdoored policies:

$$\max \left(R(\pi^*, \mathscr{E}) - R(\pi_e, \tilde{\mathscr{E}}) \right) \tag{3.9}$$

To ensure that the performance degradation in the poisoned environment is indeed caused by the backdoor and not due to any inherent sensitivity in the standard model, it is also expected that the baseline policy performs consistently regardless of the presence of the trigger:

$$|R(\pi^*, \mathscr{E}) - R(\pi^*, \tilde{\mathscr{E}})| < \varepsilon_2 \tag{3.10}$$

This formulation collectively captures the stealthiness and effectiveness requirements of a successful backdoor attack. All attacks currently in literature aim to meet this requirement.

3.6 Backdoor Attacks in DRL

Even though DRL utilises neural networks as their function approximators, only a few publications in the literature [117, 235, 50, 193, 192] have exposed vulnerabilities in the DRL pipeline to backdoor attacks. Till now, all DRL backdoor literature follow the convention of training-time poisoning.

While backdoor attacks in supervised learning have been extensively studied, their implications for DRL remain comparatively underexplored. DRL poses unique challenges due to its interactive learning paradigm, where agents acquire policies through continuous interaction with an environment. Unlike classification tasks, where outputs are independent across examples, DRL decisions influence future state distributions and rewards. Consequently, the design and success criteria for backdoor attacks in DRL differ substantially.

Backdoor attacks in DRL aim to train agents that behave normally under typical circumstances but exhibit attacker-specified actions when a trigger pattern appears in the environment state. For instance, [117] introduced one of the earliest studies on untargeted backdoor attacks in DRL, demonstrating that consistent insertion of trigger patterns during training can establish a robust trigger-to-action mapping. Such behaviour, when activated, can severely degrade policy performance or cause catastrophic failure. Authors from [12] followed an identical methodology, but implemented a trigger that completely blends into the environment (in-distribution trigger) and causes the policy to deviate into a malicious action whenever the in-distribution trigger appears in the "line-of-sight" of the agent.

Subsequent works, such as [235], have investigated trigger patterns derived from opponent actions in competitive settings, while [247] explored sequential trigger embeddings in partially observable environments. These methods often rely on handcrafted rules to identify poisoned states or utilise sustained reward manipulation over consecutive time steps to nudge the agent's policy in a malicious direction.

A distinguishing characteristic of backdoor attacks in DRL lies in the dynamic nature of data generation. In online RL, poisoned training steps not only impact the current learning update but can also influence future policy trajectories due to feedback loops in agent-environment interactions. This amplifies both the potency and risk of detection of poisoning strategies. To address this, recent approaches such as [81], [193] and [192] advocate for sparsity in poisoning, ensuring minimal yet effective contamination. Work from [50] also aligns with this objective, prioritising single-step poisoning of strategically selected states based on estimated attack value, thereby minimising detection risk while maximising long-term disruption.

Overall, existing approaches inject backdoor behaviour by manipulating rewards, actions, and/or state observations. The central objective across these studies is to minimise the number

of poisoned observations in training required to induce a successful backdoor, referred to as the *Attack Sparsity Rate*, while simultaneously maximising the frequency of backdoored actions upon the presence of an adversary-defined trigger, termed the *Attack Success Rate*. Equally critical is ensuring that the backdoored agent maintains performance comparable to a benign policy when evaluated on clean, unaltered inputs (this is known as the *Clean Data Accuracy*). Collectively, these attributes contribute to the construction of a backdoored DRL agent that is both highly effective and difficult to detect, even through manual inspection.

3.6.1 Threat Model

All DRL backdoor attacks in literature currently possess a very similar threat model. Specifically, all DRL backdoor attacks require access to the training pipeline in which they are able to make manipulations to the codebase at some point before (and during) the training process. However, prior knowledge and assumptions about the user methodology varies.

TrojDRL [117] first laid the critical foundations of DRL-based backdoors by stating the key assumptions of their attacks. Specifically, they assume that the attacker; 1) cannot change the architecture of the policy and value networks, 2) cannot change the DRL algorithm used for the training of the agent, and 3) can only change the states, actions and rewards that are communicated between the agent and the environment. The authors also stated critical objectives of the attacker that are also commonly used in all DRL backdoor attack literature.

The core objective of the overall backdoor attack is dual in nature. On one hand, the attacker seeks to train an agent that is indistinguishable from a normally-trained model in terms of performance when no trigger is present in the input. On the other hand, the presence of a backdoor trigger should cause the agent's performance to degrade as significantly as possible. Based on objective, the authors propose two threat models; Strong Attacker, and a Weak Attacker.

A Strong Attacker considers the scenario in which the training of the agent is outsourced to an external service provider, such as a cloud platform or a third-party offering pretrained models. The adversary, in this case, resides on the provider's side and possesses complete control over the training process. This includes full access to, and the ability to manipulate, the state, action, and reward signals at every timestep of training. Given this comprehensive level of access, the attacker can stealthily embed a backdoor into the model. This model represents a high-risk, high-capability adversary and is thus termed a *strong attacker*.

A Weak Attacker focuses on a more constrained adversary; one who has no direct access to the model or its internal decision-making processes. Instead, the attacker can only influence the training process through tampering with the environment. This setup corresponds to scenarios where the environment itself is adversarially crafted or manipulated. As the model

is trained locally by the client, the attacker must rely solely on altering the states and rewards perceived by the agent, without the ability to modify actions. As a result, the emphasis shifts towards ensuring high attack stealth, as the training procedure is observable by the user. This model reflects a more practical yet still potent adversarial setting and is referred to as a *weak attacker*.

Under both threat models, a *targeted attack* refers to a scenario in which the adversary aims to train the agent to execute a specific target action a_e when a particular input state s_t is modified with a chosen trigger pattern Δ and binary mask λ, as described in Equation 3.5. The primary goal is to embed this malicious behaviour while preserving high performance on clean, unaltered inputs.

To maintain stealth, the adversary seeks to poison as few states as possible (minimising the *Attack Sparsity Rate*) and ensures that any reward modifications remain within the typical reward distribution. This minimisation reduces the likelihood of detection by human oversight or statistical anomaly detection methods, while still achieving the desired malicious outcome when the trigger is encountered.

In contrast, an *untargeted attack* does not aim to elicit a specific action, but rather to induce arbitrary or harmful behaviour upon the presence of the trigger. In control-critical applications (such as autonomous driving), even random deviations can lead to catastrophic outcomes. For instance, the insertion of a backdoor might cause the model to produce erratic steering behaviour when triggered.

As with targeted attacks, the attacker in an untargeted setting strives to poison as few training samples as possible and to ensure that the model maintains high utility on clean data. This stealthy injection of unreliable behaviour can be equally, if not more dangerous due to its unpredictability.

Authors from [235] extend the above Strong Targeted Attacker threat model in which the user outsources the training to a malicious developer or downloads a pretrained policy from the developer. However, the authors only modify the action space and refrain to modify the states to evade visual detection.

The BadRL [50] backdoor attack follows the Strong Targeted Attacker methodology as it injects trigger patterns into the observation data while overriding selected actions to force malicious behaviour that induces a forced, modified (positive) reward. The authors of this paper also implement a black-box attack where the attacker does not know the clean MDP environment either. However, the authors suggest that the attacker has access to the simulator to interact with the clean environment for an arbitrary number of rounds.

Authors from [193] propose SleeperNets, which also follows the Strong Targeted Attacker methodology as their proposed adversary has full access to the training machine. This

Attack Name	Threat Model	Attacker Knowledge	Code Access	Replay Buffer Access	Model Weights Access	Environment Access	Knowledge of $P(s'\|s,a)$	Modifies State	Modifies Action	Modifies Reward
SleeperNets	Training (Outer-Loop)	White-box	•	•		•	•	•		•
Q-Incept	Training (Outer-Loop)	White-box	•	•		•	•	•	•	•
TrojDRL	Training (Inner-Loop)	White-box		•		•		•	○	•
BadRL	Training (Inner-Loop)	White-box		•		•	•	•	○	•
BACKDOORL	Training (Full Control)	White-box	•	•	•	•	•		•	

Table 3.1 Unified analytical taxonomy of DRL backdoor attacks combining adversarial knowledge assumptions, supply-chain access, and Markov Decision Process manipulation. The symbol • denotes capabilities required by all variants of an attack, while ○ indicates multiple strategies, some of which require the corresponding capability. All listed attacks result in policy-level compromise rather than transient inference-time perturbations.

includes the modifications of the state, reward along with the knowledge of the transition function. The overall breach in the states and rewards are made before the policy of the algorithm is evaluated i.e., outer-loop. The same authors also propose Q-Incept [192] which also follow the Strong Targeted Attack threat model in which the adversary is able to infiltrate the end user's system and manipulate RAM values. The authors also leverage the same outer-loop attack and alter the states, actions and rewards of the agent stored in the trajectory before the agent uses them for policy optimisation. All existing attacks in literature are summarised in table 3.1 with their attacker capability matrix.

3.6.2 Backdoor Defences

Despite the significant threat of backdoor attacks in DRL, very few defences [20, 4, 36, 248] have been proposed. In addition, most defences require very specific (and often unrealistic) assumptions to detect and/or sanitise such attacks. Till now, only three defences [20, 36, 248] have initiated to generalise their defence for a variety of environments. Nevertheless, this subsection will summarise all current DRL backdoor defences to provide the reader with insights on the existing state of the literature.

Work from [4] introduced an attribution analysis-based algorithm to detect DRL backdoors, exploiting advantage prediction sensitivities to observation changes. They used Jacobian matrices to identify trigger-affected inputs, showing success in IARPA's TrojAI. However, the approach lacks practicality due to the extensive training requirement of the challenge that exhausts the number of backdoors the classifier visualises during training. Realistically, this may not be possible because of the scarcity of backdoored samples during inference.

PolicyCleanse [89] for backdoor detection and mitigation in Competitive RL (CRL), using reward reversal to identify and counteract opponent-triggered backdoors. It generates a

backdoored policy to mimic potential triggers, reverses the reward function, and evaluates the target for malicious behaviour. Triggers are mitigated by training the victim with benign and pseudotrigger episodes. While the methodology of the solution may be able to provide a holistic DRL backdoor defence in a variety of environments, the authors have only shown its effectiveness for CRL, and not single-agent DRL, making it outside the scope of this research.

Authors from [20] propose a "*Provable Defence*" sanitisation mechanism against the TrojDRL [117] attack. Their defence is conceptually a wrapper method that operates by projecting observed actions onto a "safe subspace" which is estimated from a small number of interactions with a clean (non-triggered) environment. The method relies on several key assumptions about the attack and the policy. The adversary is assumed to use a "subspace trigger", meaning they can only inject triggers only in a specific eigenspace of the state space (out-of-distribution triggers). The authors also theorectically define a safe subspace of the agent state observation to effectively detect any anomalous states that may appear (through the backdoor trigger). The authors estimate this safe subspace by collecting $\tilde{2}1000$ samples from a "clean environment" and through assumptions, filter out the trigger component from the observed state, even if that trigger's influence has propagated to the future state. The authors prove their methodology in Atari Breakout under the assumption that their collected samples will only comprise of the clean observations and that all backdoor attacks exist outside their safe subspace.

Till the end of 2023, only the above DRL backdoor defences were published, with [20] being the most accredited in literature solution due to it being published at a top-tier AI research venue. In 2024, [36] and [248] were also published in top-tier AI research venues and promised high potential for backdoor detection.

BIRD (Backdoor Identification and Removal for DRL), proposed by [36], is a defence mechanism designed to detect and eliminate backdoors from pretrained, potentially poisoned DRL policies. Unlike earlier approaches, BIRD is generalisable, scalable, and does not require access to the original training process or prior knowledge of the attack—such as the trigger, poisoned actions, or reward functions. Instead, it relies on analysing the distinctive behaviours of backdoored agents.

BIRD comprises three main stages: trigger restoration, backdoor detection, and backdoor removal. In the restoration phase, it formulates an optimisation problem to identify perturbations in the state space that induce high-value actions. This is based on the insight that backdoored agents typically associate specific triggers with elevated value estimates due to poisoned training rewards. To regularise the optimisation and reduce noise, BIRD promotes

small and dense perturbations—consistent with common trigger designs—and employs a Beta-distribution-based generative model for stable and compact trigger restoration.

For backdoor detection, BIRD measures the drop in the agent's actual reward when the restored trigger is applied in a clean environment. A substantial performance decline is interpreted as evidence of a backdoor, as benign agents should not be affected by such perturbations.

To remove the backdoor, BIRD identifies the top L most activated neurons under the restored trigger—presumed to encode backdoor-specific behaviour—and reinitialises them. The agent is then finetuned in a clean environment with continued trigger exposure. To preserve clean performance, a Kullback-Leibler divergence regularisation term is applied between the original and updated policy. BIRD has been shown to be effective and robust across a variety of DRL environments and backdoor types.

SHINE, proposed by [248], is a defence method designed to shield pretrained DRL agents from backdoor attacks during the testing phase, even when the presence of a backdoor is unknown. It is specifically tailored for use in deployment settings and does not require access to the original training process or clean environment. SHINE consists of two primary stages: *trigger restoration* and *backdoor shielding*.

In the trigger restoration phase, SHINE collects agent trajectories from a potentially poisoned environment and applies a two-stage explanation strategy. First, step-level analysis is conducted to identify time steps associated with performance failures. This is followed by feature-level analysis to pinpoint influential state features likely responsible for activating the backdoor trigger.

Once the trigger is inferred, the backdoor shielding phase retrains the agent using a carefully constructed objective function that enhances performance in the poisoned environment while bounding performance degradation in clean settings. This retraining approach ensures robustness to the trigger without sacrificing the agent's original capabilities.

SHINE has been evaluated against three types of DRL backdoor attacks—perturbation-based single-agent attacks, adversarial two-player attacks, and cooperative multi-agent attacks—across seven benchmark environments. Experimental results demonstrate that SHINE significantly outperforms prior defences in both identifying triggers and mitigating their effects, achieving strong robustness in poisoned environments while maintaining clean performance.

Despite the demonstrated effectiveness of existing backdoor defence mechanisms, their practical deployment remains significantly constrained. Most current defences are tailored to specific environments, particularly the Atari domain, and demand considerable computational resources, rendering them unsuitable for broader or real-time applications. This limitation

is particularly critical in safety-sensitive settings, such as autonomous driving, where DRL agents must make rapid decisions in dynamic environments. Within the research lens, these defences have not yet been tested by other researchers, reducing their generalisation credibility. Therefore, the next section experiments with one of the three main defences [20] to measure its credibility and ability to defend against more sophisticated backdoor triggers.

3.7 Assessing A State-of-the-Art DRL Backdoor Defence

As mentioned in the previous section, the lack of testing of DRL backdoor defences is a significant bottleneck towards the real-world usage of DRL backdoor defences. Given that this practice is much more common in the wider AI literature [234], we aim to extend this practice within DRL security by testing [20]'s backdoor defence.

Due to the scarcity of defence methods specifically targeting backdoor attacks in DRL [139, 235], authors from [20] proposed an unsupervised sanitisation technique for pretrained agents. Their approach estimates a clean empirical subspace using the covariance matrix and eigen-decomposition of state observations collected in a clean environment. Incoming observations are projected onto this subspace, and any components lying outside it are filtered and replaced with alternatives within the identified "safe" subspace.

This method was evaluated on an Atari environment known as *Breakout*, a classic arcade game in which the player controls a paddle that moves horizontally along the bottom of the screen. The objective of the game is to bounce a ball upward to break a wall of bricks positioned at the top; each brick hit by the ball disappears, awarding points to the player. The game ends when the player fails to prevent the ball from falling below the paddle after a limited number of attempts. In the context of our experiments, a backdoored policy in *Breakout* is defined as one where, upon encountering a small out-of-distribution trigger, the agent consistently moves the paddle towards the right corner, irrespective of the ball's actual position.

Overall, the performance of the backdoor defence technique against the backdoor in Atari Breakout was shown to depend on factors such as the size of the collected sample set and the dimensionality of the projected subspace. As mentioned, at the time of conducting experiments for this chapter, [20]'s method remained the only known technique in the literature capable of sanitising single-agent backdoored DRL policies without requiring any prior knowledge of the training environment or the trigger mechanism.

In order to test the credibility of the DRL backdoor defence literature, this section evaluates the robustness of their defence by challenging the core algorithmic assumptions that underpin it. In particular, we propose an alternative form of stealthy backdoor triggers,

referred to as in-distribution triggers, that may bypass their sanitisation process by invalidating key theoretical assumptions.

3.7.1 Key Assumptions

The sanitisation technique proposed by the authors [20] relies on three foundational mathematical assumptions. In what follows, we outline these assumptions and critically assess their implications for both attackers and defenders. We conclude by using the theory of one assumption and challenging the theory of another through our proposed adversarial strategy.

Assumption 1 pertains to the distribution of states visited under the optimal policy in a clean, non-triggered environment. It assumes that clean and benign states are concentrated within a bounded region of the state space, specifically within what is termed the "smallest $D-d$ eigen-subspace", denoted by $E^{\perp}$. This subspace is defined by the eigenvectors corresponding to the lowest eigenvalues of the empirical state covariance matrix. The assumption implies that, when projected into this subspace, clean states do not exhibit large variations and are effectively bounded in magnitude. This is considerably a reasonable assumption for many real-world environments, where physical or operational constraints naturally limit the variability of observed states. Furthermore, the state distribution is assumed to be K-subGaussian, implying that the probability of extreme values decays rapidly, providing formal guarantees on the behaviour of the clean state distribution.

From the perspective of an adversary, this assumption creates an opportunity: states that fall outside the bounded region of $E^{\perp}$ are rarely visited during normal execution. This allows the attacker to embed malicious behaviours in these less frequently encountered regions without affecting the agent's performance on standard tasks, thus increasing the stealthiness of the attack.

Assumption 2 constrains the form of the backdoor policy introduced by the adversary. Specifically, it assumes that the policy is L-Lipschitz smooth, meaning that small changes in the input state lead to proportionally small changes in the output action. This smoothness condition, common to policies trained using gradient-based methods such as neural networks, ensures that the backdoor policy behaves similarly to the optimal policy within regions of the state space commonly visited during clean execution. Consequently, the policy appears well-behaved under standard evaluation, while reserving malicious behaviour for regions less frequently explored.

Assumption 3 characterises the behaviour of the trigger function and the resulting triggered states perceived by the agent. Central to this assumption is the notion of a *subspace trigger*: the trigger function modifies observations by adding components that lie exclusively within $E^{\perp}$, the region outside the identified safe subspace E. This effectively restricts the

trigger to operate in directions deemed atypical in the clean state distribution. The trigger function may also be adaptive, depending on the agent's historical states, actions, or rewards. Furthermore, the assumption imposes a boundedness constraint: the triggered state (i.e., the original state plus the trigger) is assumed to be B-bounded in expectation. This ensures that, on average, the injected perturbations do not produce excessively large values, which could otherwise render the attack easily detectable.

This third assumption is integral to their defence mechanism itself. The sanitisation algorithm works by identifying the safe subspace E from clean data and projecting each incoming observation onto it. Since the trigger is assumed to be confined to $E^{\perp}$, the projection step is designed to eliminate its influence. The boundedness assumption further supports the theoretical guarantees of the sanitisation procedure by limiting the adversary's ability to exploit extreme values.

3.7.2 Sanitisation Technique

The core technique proposed in [20] aims to neutralise adversarial triggers by projecting the observed state onto an estimated *safe subspace* prior to feeding it into the potentially backdoored policy. This approach comprises two primary phases: *sanitisation* and *deployment*.

Sanitisation phase. During this phase, the defender interacts with the provided policy in a clean environment—one that does not contain any backdoor triggers. A small number of state observations are collected from these interactions, which are assumed to represent the normal behavioural trajectory of the agent under clean conditions. The defender then computes the empirical covariance matrix of these collected states and performs an eigendecomposition (or Singular Value Decomposition). The principal eigenvectors obtained from this analysis are used to define the estimated safe subspace, which characterises the typical states encountered by the policy in a benign environment.

Based on this estimated subspace, a linear projection operator is constructed. The sanitised policy is then defined as a modification of the original backdoored policy, where every input state is first projected onto the safe subspace before being passed to the policy network.

Deployment phase. In this phase, the agent operates using the sanitised policy. At each time step, the agent receives an observed state from the environment. This observation may correspond to a clean state or may have been modified by the adversary to include a trigger component. Before the policy processes this input, the sanitisation mechanism intervenes. The observed state is projected onto the safe subspace using the operator constructed during the sanitisation phase. The resulting projected state—purged of any components lying outside

the safe subspace—is then passed to the original backdoored policy to determine the agent's action.

This defence relies on a crucial assumption (Assumption 3 in [20]): that the adversarial trigger lies outside the estimated safe subspace. By projecting the observed state onto this subspace, the defence mechanism effectively removes the trigger component, thereby neutralising its influence on the policy's decision-making. As a result, the input fed into the backdoored policy closely resembles one that would arise under clean conditions, allowing the agent to act in a manner that approximates optimal behaviour, even in the presence of backdoor triggers.

3.7.3 Challenging the Defence Assumptions

In this work, we specifically focus on leveraging the theoretical underpinnings of **Assumption 1** while challenging **Assumption 3** within our defined threat model. This threat model is identical to that adopted by [20]. We hypothesise that an adversary could design *in-distribution triggers* (as introduced in Section 5.3.1)—malicious patterns or configurations that lie entirely within the distribution of clean state subspace E—to activate backdoored behaviours. Such triggers, while realistic (suggested by [12]) stand in direct contradiction to Assumption 3, which presumes that adversarial triggers can exist only outside the safe subspace.

These in-distribution triggers would not violate the bounded support constraint laid out in Assumption 1 and would be statistically indistinguishable from naturally occurring, clean state observations. As a result, we hypothesise that such triggers would evade sanitisation by projection-based defences, which operate under the assumption that harmful components are geometrically separable from the clean distribution and can thus be filtered out by subspace projection.

This analysis reveals a fundamental limitation in the defence framework proposed by [20]: its dependence on geometric separability between clean and triggered states. If the adversary embeds triggers within the support of the clean state distribution, this assumption may not hold, thereby undermining the efficacy of the defence. Our findings underscore the need for more robust detection and sanitisation methods that account for triggers embedded within the clean subspace, rather than solely relying on subspace exclusion.

In the original experiments by [20], the backdoor trigger for the Atari *Breakout* environment consisted of a conspicuous 3×3 white pixel square located in the top-left corner of the screen. This modification does not align with the game's natural palette or design aesthetics (see Figure 3.3b), and therefore clearly lies outside the clean observation distribution.

Fig. 3.1 Visual representation of applying the sanitisation algorithm proposed by Bharti et al. [20] in the Atari *Breakout* environment, under two different backdoor trigger conditions: (a) the sanitised state where the algorithm successfully removes the out-of-distribution 3×3 white square trigger in the top-left corner of the screen; (b) the original triggered state with the 3×3 white square; and (c) the failed sanitisation of our in-distribution trigger, which appears as a missing tile and is not removed by the projection-based defence.

In contrast, using the theory developed in Section 5.3.1, we empirically evaluate the defence algorithm's guarantees by introducing an *in-distribution* backdoor trigger into the *Breakout* environment. As shown in Figure 3.1c, our trigger appears as a missing tile within the game's tile array—a modification that adheres more closely to the statistical distribution of clean observations, yet is practically infeasible under the actual game mechanics. Nevertheless, it fulfills the definition of an in-distribution trigger.

Specifically, this backdoor trigger is introduced at regular intervals in the environment, and the agent is rewarded for exhibiting unsafe behaviours whenever the trigger is present.

3.7.4 Experimental Setup

The backdoor trigger implementation closely follows the methodology established by authors from [117], which is widely regarded for producing strong and highly targeted backdoor attacks. This methodology has also been adopted by authors from [20], who built directly upon the original implementation of TrojDRL. The TrojDRL codebase, which has been made publicly available by the original authors [1], serves as the foundation for our implementation. Both authors conducted their experiments using the environment configuration and benchmarking protocols defined by [43].

The original implementation was developed using TensorFlow and Keras libraries, which are well established tools for deep learning research. Consistent with the setup used in prior work, all experiments in this thesis were conducted on hardware equipped with NVIDIA Tesla V100 graphics processing units, commonly used in DRL and deep learning experiments.

[1] https://github.com/pkiourti/rl_backdoor?tab=readme-ov-file

3.7.5 Results and Analysis: Simple Trigger vs In-distribution Trigger

Using the authors' publicly available defence implementation[2], we executed their sanitisation algorithm on both the originally proposed "simple trigger" and our proposed *in-distribution trigger*. The results empirically support our hypothesis and demonstrate a key limitation in the defence framework.

As shown in Figure 3.2, the results for the simple trigger (represented by the green line) indicate that the sanitisation algorithm successfully neutralises the backdoor once approximately 21,000 samples are collected. This sample size appears sufficient to accurately estimate the theoretical safe subspace of the Atari *Breakout* environment. Once this subspace has been constructed, the projection-based defence is able to remove the out-of-distribution trigger, restoring or even exceeding the agent's performance in the clean environment (represented by the blue line). Notably, the algorithm's performance in this setting is consistent, as reflected in the low standard deviation of episodic returns.

More importantly, however, we observe that our in-distribution backdoor trigger successfully evades the sanitisation algorithm of [20]. This is evidenced by the orange line in Figure 3.2, which shows no significant increase in average empirical return, regardless of the number of samples used to estimate the safe subspace. The orange line remains consistently below or equal to the red line, which depicts the performance of the backdoored policy (using TrojDRL's [117] trigger) without any sanitisation applied. These results empirically demonstrate that our in-distribution trigger lies within the estimated safe subspace and is therefore not removed by the projection-based defence. Consequently, the backdoored agent continues to perceive the trigger and execute malicious actions in response.

To further validate this observation, we conducted an ablation study involving variations in the dimensionality of the estimated safe subspace, as implemented using the Support Vector Dimension algorithm. Figure 3.3 shows that, for the simple trigger (green line), subspace dimensions of around 20,000 or higher are sufficient to restore performance to levels comparable to clean actor-critic agents in the Atari *Breakout* environment. In contrast, the orange line representing our in-distribution trigger remains largely unaffected by changes in dimensionality. Although a slight improvement in performance is observed beyond 25,000 dimensions, there is no definitive indication of enhanced sanitisation efficacy beyond 28,000 dimensions—the maximum tested by the original authors.

Taken together, Figures 3.2 and 3.3 provide strong empirical validation of our hypothesis. Specifically, they demonstrate that in-distribution triggers—when deliberately designed to lie within the empirical support of the clean state distribution—are not effectively neutralised by the sanitisation algorithm proposed in [20]. This result confirms that Assumption 3,

[2]https://github.com/skbharti/Provable-Defense-in-RL/tree/main?tab=readme-ov-file

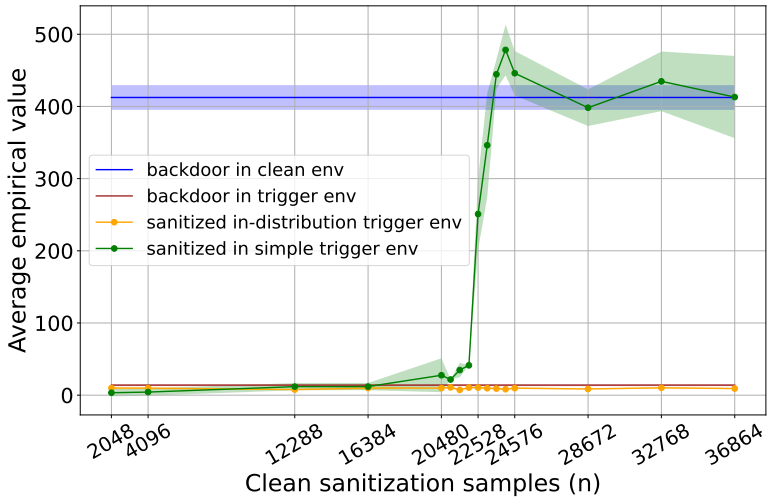


Fig. 3.2 Graph comparing the effectiveness of Bharti et al.'s [20] sanitisation algorithm against sample size, with agent performance baselines in clean (blue line) and simple trigger scenarios (red line). The algorithm's effect on neutralising a simple trigger is shown by the green line, while its impact on our in-distribution trigger is illustrated by the orange line. The results show that our in-distribution trigger eludes neutralisation by their algorithm, highlighting its inability to detect subtle triggers.

which underpins the theoretical foundations of their method, does not hold in the presence of carefully constructed in-distribution triggers.

We observed that the execution time of the sanitisation algorithm spanned several days, exceeding even the duration required to train the DRL agent itself. As DRL agents are increasingly anticipated for deployment in high-stakes applications—such as autonomous vehicles operating on public highways—a delayed or ineffective defence against backdoor attacks can have catastrophic consequences. For instance, a backdoor trigger embedded within the in-distribution space of the environment could cause the agent to take a malicious action, such as a sudden leftward swerve, potentially resulting in life-threatening accidents.

This highlights a critical limitation of the current approach: its unsuitability for both rapid- and delayed-response scenarios when dealing with complex backdoor threats. An ideal solution in such safety-critical domains must be capable of detecting in-distribution backdoor triggers in real-time and sanitising them before they influence the agent's behaviour.

At present, no existing methods in the literature (including those developed with high levels of engineering scrutiny [36, 248]) meets the *practical* requirements for real-world and real-time backdoor defence. Therefore, the need for immediate, reliable, and adaptive detection countermeasures in dynamic environments is therefore both evident and urgent before crafting sanitisation techniques.

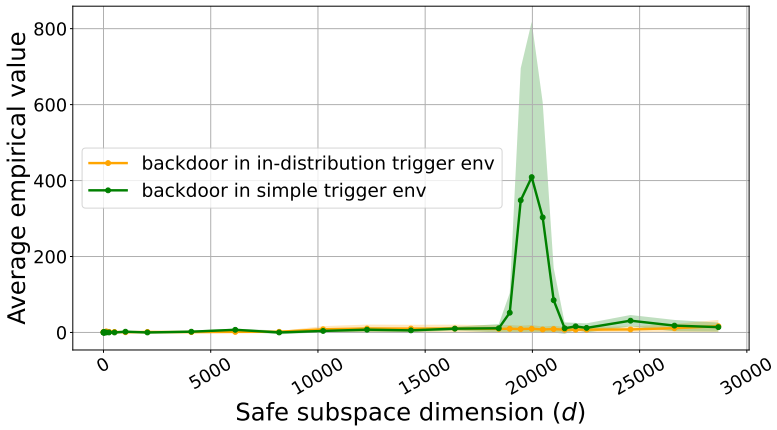


Fig. 3.3 The graph shows the impact of Bharti et al.'s [20] sanitisation algorithm on agent behaviour with increasing empirical safe subspace dimensions across 32,768 samples. The green line shows how the algorithm retains the performance of the agent when the safe subspace has 20,000 dimensions, while simultaneously neutralising a simple backdoor trigger. The orange line depicts its performance when (unsuccessfully) attempts to neutralise our in-distribution trigger. This highlights that the in-distribution trigger is within the algorithm's safe subspace and evades the defence.

3.8 Discussion

Till now, backdoors are slowly gaining traction in the AI community. However, publications on DRL backdoors have been limited. Overall, only 4 publications at top-tier venues delved into backdoor attacks, while only 3 publications at top-tier venues have implemented sophisticated defences. In order to propel Autonomous Blue Teaming solutions using DRL, researchers and practitioners will urgently need to expose more vulnerabilities in DRL backdoors along with creating sophisticated defences.

3.8.1 DRL Backdoor Attacks

Of the three main DRL attacks proposed in this chapter, the authors have expanded the methods of backdoor injection within the commonly assumed threat models of the literature. While these works represent significant initial progress in the DRL domain, a key limitation in literature is their restricted threat model assumptions. Existing studies focus solely on training-time attacks, overlooking the possibility of successful backdoor injection both prior to and after the training phase (i.e., during code project initiation, post-training and inference-time). This narrow scope has a cascading effect on current defence mechanisms, which are designed specifically to counter only the existing class of DRL backdoor attacks. Therefore,

researchers must aim to expose a wider threat landscape in the DRL backdoor literature to allow for potentially more holistic DRL backdoor defences in the future.

3.8.2 DRL Backdoor Defences

From the current DRL backdoor defence literature, all works have made significant assumptions and have empirically (and in some cases theoretically) demonstrated effectiveness against certain backdoor attacks. However, until now, no authors have critically challenged the foundational assumptions of these defences through more sophisticated theoretical or empirical contradictions. This chapter advances DRL backdoor research by addressing the second research question, **RQ2**, and making the second contribution, **C2**. This is achieved by critically evaluating a key state-of-the-art DRL backdoor defence in light of its theoretical assumptions. As the experiments in this chapter were conducted in 2023, demonstrating the defence's ineffectiveness at that time highlighted a significant research gap in the DRL backdoor defence landscape.

That said, even in 2025, a major bottleneck in existing defence approaches lies in their reliance on resource-intensive sanitisation techniques, which often requires access to extensive trajectories, offline retraining, or large-scale inference pipelines. These methods are not designed to operate in real-time, making them impractical for deployment in scenarios where immediate action is required to mitigate the consequences of a detected backdoor. Furthermore, the rarity of adversarial trigger occurrences in real-world deployments, such as those embedded in the self-driving car context, may lead to a false sense of security. For example, a self-driving car with a backdoored policy might perform adequately in most conditions, passing safety evaluations and regulatory benchmarks. However, once the trigger is encountered through an adversary (planted as a roadside cue or a traffic artefact) the agent could exhibit catastrophic behaviour. In such high-stakes scenarios, it becomes imperative not only to detect the presence of a backdoor promptly but also to sanitise the affected policy *in real-time* to prevent harm to passengers and others.

Consequently, there is a pressing need to shift the research focus towards developing efficient, real-time backdoor detection frameworks that can operate in online settings. These detection systems must serve as a prerequisite to any sanitisation procedure, ensuring that remediation efforts are only applied when a genuine threat is detected. Without such real-time capabilities, the practicality and safety assurances of DRL systems in real-world deployments remain limited.

3.9 Thesis Scope

Building upon the open challenges and research gaps identified in the previous section, this thesis focuses on enhancing the attack robustness of autonomous blue agents through DRL-based methods. In particular, this thesis contributes towards the fulfillment of requirements (A.6.3) and (G.6.2), which highlight the importance of robustness of autonomous blue teaming agents against algorithmic attacks.

Despite ongoing progress in DRL, the literature has not adequately addressed supply chain attacks or proposed robust defence mechanisms, even within the controlled confines of simulated and gamified environments. By advancing solutions aligned with requirement (A.6.3), this work aims to lay a strong foundation for future research addressing the broader goals outlined in requirement (G.6.2).

The central security threat addressed in this thesis is backdoor attacks in DRL algorithms. These attacks pose a significant risk to AI systems deployed in safety-critical domains, with the potential to cause far-reaching consequences. In the context of ACND, a DRL-based autonomous blue teaming agent compromised via a backdoor could appear to function normally while secretly facilitating adversarial objectives (such as bypassing network defences or enabling undetected intrusions), when activated under specific adversary-controlled conditions [162]. This thesis, therefore, investigates and mitigates such backdoor vulnerabilities, contributing to the development of more secure and trustworthy autonomous blue agents.

The next chapter introduces the methodology used for the experimentations, detailing the experimental setup, environment configurations, evaluation metrics, and the rationale behind the design choices. This provides the foundation for validating the backdoor attack and defence mechanisms proposed throughout the thesis.

Chapter 4

Experimental Methodology and Approach

4.1 Introduction

In Chapters 2 and 3, we introduce a potential DRL application safety-critical domain of ACND and provide a focused review of a key research gap, DRL backdoors. To test the current DRL backdoor defence literature, we evaluate the credibility of one of the existing state-of-the-art research solutions and empirically demonstrate its limitations by challenging the foundational assumptions upon which it is based. Although the analysis presented in Chapter 3 involves an experimental evaluation of an existing DRL backdoor defence, it is intentionally positioned before this methodology chapter. This structural decision reflects the role of that analysis as a critical motivating study rather than a component of the proposed methodology. By empirically demonstrating the limitations of a widely cited defence [20], the study in Chapter 3 exposes foundational weaknesses in both current defences and, by extension, the attacks they are designed to counter. These findings directly informed the problem formulation and the design objectives of the methodology developed in this chapter.

This chapter presents the methodological framework that underpins the main contributions of the technical chapters that follow. In particular, it outlines the experimental setup, evaluation metrics, and implementation framework used to systematically assess the efficacy of DRL backdoor defences and attacks. This unified methodology ensures consistency and rigour across the various experimental contributions presented in the subsequent chapters.

Overall, this thesis seeks to make substantive progress in the field of DRL security by developing effective backdoor detection strategies and proposing novel backdoor attacks that significantly broaden and reshape the existing threat landscape. Through a combination of

theoretical inquiry and empirical validation, the work contributes to a more comprehensive understanding of both offensive and defensive mechanisms in DRL backdoors, thereby laying the groundwork for the development of more secure and robust autonomous systems.

4.2 Key Research Gaps and Objectives

As discussed in Chapter 3, the current landscape of DRL backdoor research lacks robust defence mechanisms grounded in effective backdoor trigger detection strategies. In particular, all three existing DRL backdoor defences [20, 36, 248] rely heavily on input observations to identify or eliminate the backdoor trigger. While this approach facilitates the detection of out-of-distribution triggers, it is important to recognise that, in real-world settings, backdoor triggers are more likely to fall within the distribution of benign input observations, as also noted by [12].

In parallel, the current body of work on DRL backdoor attacks is predominantly centred around unrealistic training-time compromises. Although such attacks have demonstrated high effectiveness and stealth, their dependence on significantly elevated adversarial privileges limits their plausibility in practical scenarios.

Therefore, given that the specialised research gaps have been identified, this thesis addresses several key research contributions in the domain of DRL backdoor attacks and defences. Specifically, in the context of DRL backdoor defences, the objective is to design a real-time detection system capable of identifying the most concealed backdoors before they lead the agent to perform malicious actions in the environment in which they are deployed. This proactive detection capability is intended to serve as a foundation for future researchers to develop more sophisticated DRL backdoor sanitisation techniques.

In parallel, with regards to DRL backdoor attacks, the aim is to expand the current threat landscape by introducing backdoor attacks that demonstrate the following characteristics: (1) reduced adversarial access requirements, (2) improved attack success rates and effectiveness, and (3) enhanced stealth. While effective trigger design is an important part of backdoor attacks, the key contributions of our attacks lie in their adversarial access requirements. We encourage researchers to explore effective and holistic DRL backdoor trigger designs that can lie more within the distribution of the environment.

Nonetheless, to support our technical research contributions, this chapter adopts conventional experimental setups as established in the literature, thereby enabling meaningful advancements in both strands of DRL backdoor research.

4.3 DRL Backdoor Defence Experimentation

4.3.1 Experimental Setup

The experimental framework for DRL backdoor defence builds upon the work of [12], who introduce in-distribution backdoor triggers designed to blend seamlessly into the environment. Unlike conventional backdoor research that employs static or visually conspicuous triggers (commonly used in environments such as Atari) in-distribution triggers are crafted to be both visually and behaviourally indistinguishable from benign elements, thereby increasing their evasiveness.

To enhance variability and robustness in evaluation, we adopt the MiniGrid environment, which is a gamified 7×7 gridworld environment provides a diverse set of configurations and task variations suitable for fine-grained analysis of policy behaviour under stealthy backdoor conditions. Our experiments are conducted using the PPO algorithm [203], implemented in PyTorch via the RL Starter Files framework[1]. This setup follows the implementation used by [12], enabling fair comparison and reproducibility in our analysis (more details provided below). All experiments are conducted on NVIDIA Tesla V100.

In alignment with prior work, we replicate a similar class of in-distribution backdoor triggers and introduce a lightweight detection framework aimed at identifying poisoned agents. The primary objective is to evaluate the efficacy of our system in distinguishing between benign and backdoored agents without requiring access to the poisoned environment or poisoned trajectories.

Reproducibility Details

We trained a PPO using the torch-ac across several MiniGrid LavaWorld variants, which require partial observability and discrete control. Each agent was trained for 80 million time steps with a learning rate of 0.0224, linearly annealed to zero throughout training to ensure stable convergence. The discount factor was set to $\gamma = 0.99$, and an entropy regularization coefficient of 0.02 was applied to encourage exploration. Training was performed using 32 parallel environments, each generating rollouts of 80 steps per update. Optimization employed the Adam optimizer ($\varepsilon = 10^{-5}$), with a batch size of 2560, four epochs per PPO update, clip ratio $\varepsilon = 0.2$, GAE parameter $\lambda = 0.95$, and a value loss coefficient of 0.5. To prevent instability, we used global gradient clipping with a threshold of 3.0 across all updates.

The policy followed the default MLP architecture from the rl-starter-files PPO configuration for MiniGrid, consisting of two fully connected layers with 64 hidden units each

[1] https://github.com/lcswillems/rl-starter-files

and ReLU activations, followed by separate linear heads for the actor (policy logits) and critic (state value). Observations were derived from the partial egocentric view provided by MiniGrid, encoded as flattened one-hot representations of grid cells. The action space was discrete, corresponding to the available movement and rotation actions in each environment.

We used MiniGrid-LavaWorld to test robustness and generalization of the backdoor behavior. Environments followed the Gym-MiniGrid API, with standard success rewards (+1 upon reaching the goal), shaping penalties proportional to the time-step length, and termination triggered either by agent death (contact with lava) or episode timeout (100 steps).

All experiments were initialized with a fixed random seed (42) to ensure reproducibility across environment resets, parameter initialization, and sampling order. Training used the feed-forward PPO variant (no recurrence) provided by torch-ac, which synchronizes gradient updates across all environments after each rollout. The only variation across environments was the poisoning rate, which ranged from 0.020% to 0.025% of the collected samples to maintain stability while embedding subtle adversarial triggers.

4.3.2 Threat Model

Our threat model assumes a defender who has no access to poisoned environments or to any trajectories generated under backdoored policies. The only available data consists of 10,000 episodes of benign agent behaviour. This setting aligns with the conventional assumptions made in existing DRL backdoor defence literature [20, 36, 248].

However, unlike previous defences, which focus predominantly on detecting conspicuous or out-of-distribution triggers, our method targets evasive, in-distribution triggers designed to evade such detectors. The novelty of our defence lies in its ability to identify subtle policy-level inconsistencies caused by these stealthy backdoors, without relying on access to poisoned data or pre-specified trigger patterns.

4.3.3 Assessment Metrics

To evaluate the quality of our detection system, we use the F1 score as a primary metric, as it provides a more nuanced assessment of model performance than accuracy alone. Specifically, the F1 score offers a balanced measure by accounting for both precision and recall, making it particularly valuable in scenarios with imbalanced class distributions. This is especially relevant in our context, where benign samples may significantly outnumber poisoned samples, and high accuracy can be misleading if false negatives are overlooked.

In addition, we employ the Receiver Operating Characteristic (ROC) curve and the corresponding Area Under the Curve (AUC) to further assess detection performance across

various classification thresholds. The ROC curve provides a graphical representation of the trade-off between the true positive rate and false positive rate, offering a more comprehensive understanding of the model's discriminatory power. A higher AUC indicates that the detection system is capable of distinguishing between backdoored and benign agents across a wide range of thresholds, thereby supporting more robust and adaptable deployment in real-world scenarios.

4.3.4 Limitations

Despite the contributions made in Chapter 6, our experimental setup presents two key limitations. First, although we collect a substantial dataset of 10,000 episodes (approximately 50,000 steps) to train and evaluate our DRL backdoor detector, the experiments are conducted in a single MiniGrid environment. Nonetheless, it is important to note that MiniGrid offers significant environmental variability, and our setup includes over ten distinct spatial configurations of the in-distribution backdoor trigger. This variability partially mitigates the limitation by ensuring that the detector is exposed to diverse contextual instances of the trigger. The second limitation concerns the lack of backdoor attack and defencetesting within application domains such as ACND environments. Specifically, ACO Gym environments like those provided by the CAGE CybORG frameworks. While Chapter 2 outlines the benefits of these environments for simulating realistic autonomous blue teaming and hence, DRL security) scenarios, we do not incorporate them into our experiments due to the absence of established DRL backdoor benchmarks in this domain. Consequently, our detector remains untested in high-fidelity ACND and other safety-critical domain settings, which could provide stronger validity for DRL as autonomous blue teaming agents. That said, future work can focus on extending the proposed detection framework to ACO environments, particularly as benchmark suites and attack taxonomies for DRL backdoors mature within that domain.

4.4 DRL Backdoor Attack Experimentation

4.4.1 Experimental Setup

The experimental framework for implementing DRL backdoor attacks builds upon prior work in the field [117, 50, 193, 192]. Existing DRL backdoor attack research primarily leverages widely adopted reinforcement learning libraries, such as RLlib [135, 136], to maintain reproducibility and integration with standard training pipelines. In line with this practice, our training-time backdoor attacks are implemented using RLlib [136], ensuring methodological consistency with established approaches.

For attacks targeting already trained DRL agents, we employ RLZoo [57], which provides access to pretrained, high-performing models across standard benchmarks using PyTorch and Stable Baselines. This enables a robust evaluation of our attack effectiveness against agents with established behavioural baselines and minimal noise in performance metrics.

All attack experiments are conducted on benchmark Atari environments, consistent with prior literature. Specifically, we evaluate our attacks on four widely studied Atari games: Pong, Breakout, Space Invaders, and Qbert. Each game involves an agent interacting with a dynamic environment through simple, discrete actions to maximise a game-specific score. Backdoors in these environment involve a regular trigger along with a malicious action. This selection allows for direct comparison against state-of-the-art backdoor attack techniques and facilitates evaluation across diverse gameplay dynamics and visual structures. All experiments are conducted on NVIDIA Tesla V100.

Reproducibility Details

We trained our PPO across four Atari environments—*Pong*, *Breakout*, *Qbert*, and *Space Invaders*—using a consistent set of hyperparameters to ensure fair comparisons. Each agent was trained for **80 million time steps** with a learning rate of **0.0224**, linearly annealed to zero over the course of training. The discount factor was set to $\gamma = 0.99$, and an entropy regularization coefficient of **0.02** was applied to encourage exploration. Training was performed with **32 parallel environments**, each generating rollouts of 80 steps before every PPO update. We used the **Adam** optimizer ($\varepsilon = 10^{-5}$), with a batch size of **2560**, **four epochs per update**, generalized advantage estimation parameter $\lambda = 0.95$, clip ratio $\varepsilon = 0.2$, and a value loss coefficient of **0.5**. To stabilize training, global gradient clipping was applied with a threshold of **3.0**.

The policy network followed the standard Atari CNN architecture, consisting of three convolutional layers (**32** filters of size 8×8 with stride 4, **64** filters of size 4×4 with stride 2, and **64** filters of size 3×3 with stride 1), followed by a fully connected layer with **512 ReLU** units shared between the actor and critic heads. Observations were preprocessed into **84×84 grayscale frames**, frame-stacked over four consecutive observations, and passed to the shared convolutional encoder. The action space was discrete, matching the native Atari action set.

Environments were constructed using the **Gym ALE** interface and wrapped with `NoopReset`, `FireReset`, `MaxAndSkip(4)`, `WarpFrame(84,84)`, `FrameStack(4)`, and `ClipReward` wrappers to ensure standardized preprocessing and consistent termination criteria. Episodes terminated either upon life loss or when the environment signaled completion, capped at **108,000 frames** (equivalent to 30 minutes of simulated play). All experiments used a fixed

random seed (**42**) for environment initialization, network weights, and data shuffling, with deterministic evaluation enabled for reproducibility.

4.4.2 Threat Model

The threat model for our DRL backdoor attacks varies depending on the specific attack vector, as each of the three proposed attacks targets a different stage within the supply chain or deployment lifecycle. Despite this variation, all three attacks share a common assumption: the adversary possesses **no access to the training-time codebase of the entity**. This assumption is significantly more consistent with real-world scenarios involving outsourced or unverified training pipelines.

To evaluate the effectiveness of our attacks in a unified and comparable manner, we conduct experiments across four Atari games, running a total of 150 gameplay episodes per attack. While the number of episodes is lower than that used in our MiniGrid-based detection experiments, Atari environments are significantly longer in duration. Each episode typically spans thousands of time steps, yielding over 150,000 steps per game, thereby ensuring sufficient interaction length for assessing the persistence, stealth, and impact of the backdoors.

4.4.3 Assessment Metrics

For assessing the effectiveness of our backdoor attacks, we assess our attacks using benchmark metrics used by [50]. The metrics include:

- **Clean Data Accuracy (CDA):** This metric quantifies the relative performance of a backdoored model compared to a non-backdoored counterpart when operating in a clean, trigger-free environment, *subsequent to* the presence of the trigger during training or injection. An effective backdoor attack maintains a high CDA, thereby preserving the model's utility and functional integrity under normal deployment conditions.

- **Attack Effectiveness Rate (AER):** This metric measures the relative drop in performance when the backdoored model is exposed to the trigger during inference. AER is computed as the average reduction in episodic return compared to the clean (trigger-free) setting, and indicates how significantly the backdoor affects the agent's behaviour when activated.

- **Attack Success Rate (ASR):** This metric reflects the frequency with which the agent selects the attacker-specified target action in the presence of the visual trigger. It is

calculated as the proportion of target actions taken out of all actions during triggered episodes. A higher ASR indicates greater adversarial influence over the agent's policy.

These metrics provide a comprehensive framework for evaluating the primary goals of backdoor attacks: maintaining reliable performance in clean environments while inducing significant behavioural deviations in the presence of a trigger. By relying on CDA and AER, we are able to assess the extent to which the proposed attacks preserve the agent's utility under normal conditions while introducing adversarial behaviour when required.

To evaluate the effectiveness of our attacks against existing defences [36, 248], we use the overall episodic return (measured in Atari Points) as the primary performance metric. This choice aligns with the evaluation methodology established in the baseline work [117], which is also adopted by both defence papers, enabling direct and fair comparison.

Moreover, overall episodic return is a suitable metric for Atari-based benchmarks, as it captures agent behaviour in a way that is independent of the internal reward structures used during training. It reflects the agent's observable performance from an external perspective, which is critical in black-box or limited-observability settings.

However, for evaluating defences in our chosen research application of ACND, researchers will need to delve deeper into their simulated environments, such as those provided by CAGE CybORG, this metric becomes insufficient. In such contexts, more specialised and domain-specific metrics must be developed to accurately capture the quality of the defence. These may include operational indicators such as threat containment success rate, detection latency, false positive rates, or mission impact, which better reflect the real-world implications of compromised decision-making in cyber operations.

4.4.4 Limitations

Similar to the contributions made in the DRL backdoor defence domain, no experiments are conducted within the context of our chosen application domain of ACND environments, primarily due to the absence of DRL backdoor attack research targeting these settings. The current literature on DRL backdoors has largely focused on standard simulation environments such as Atari, with little to no exploration of attack feasibility or impact in operationally realistic frameworks like CybORG CAGE Challenge 1, 2, 3 and 4. While this gap limits both the development and evaluation of defences tailored for ACND applications, the contributions made by this work will allow an easy algorithmic transfer to CybORG. In addition, we believe that DRL backdoor research is currently still in a nascent stage, and more experimentation must be done on benchmark DRL environments to develop sophisticated attacks and defences before they are transferred to real-world applications.

4.5 Ethical Considerations

Given that all our experiments were conducted within simulated environments, the ethical risks associated with real-world deployment or harm are minimal. The agents trained and evaluated in this research operate entirely within virtual environments (e.g., MiniGrid and Atari), and no real systems, users, or data were involved. Nonetheless, we acknowledge that the development of backdoor attacks (even in simulation) carries dual-use implications. To mitigate this, our work is framed within a responsible disclosure context, with the primary aim of improving understanding and robustness of DRL systems against emerging threats. All code and findings related to attack implementations are documented in a way that emphasises reproducibility for research and defence development purposes, rather than for malicious application. Furthermore, we comply with institutional and academic guidelines for responsible AI research throughout this project.

4.6 Conclusion

This chapter presents the methodology that underpins the experimental and conceptual contributions made throughout the thesis. It outlines the overarching research objectives, threat models, evaluation metrics, and experimental frameworks used to investigate both the development of DRL backdoor detection systems and the implementation of novel backdoor attacks across various stages of the DRL pipeline. By standardising our evaluation environments and aligning with practices established in prior literature, we ensure that the proposed methods are both reproducible and comparable. Furthermore, this chapter identifies current limitations, such as environment generalisability and the absence of experimentation within our chosen domain of ACND (or other safety-critical application domains), and lays the groundwork for future extensions of this research into operational cyber defence domains. The methodology presented here serves as the foundation for the technical contributions that follow.

The next chapter addresses a key limitation present in all existing defences by introducing a state-of-the-art backdoor detection technique capable of identifying even the most concealed, in-distribution backdoors. This contribution lays the groundwork for future research aimed at developing more effective sanitisation and mitigation mechanisms. The subsequent chapters expand the DRL threat landscape by presenting three novel backdoor attacks, each targeting a different stage of the DRL supply chain, ranging from training to deployment, thereby illustrating the breadth and complexity of potential vulnerabilities in real-world systems.

Chapter 5

Beyond Existing Defences: Paving the Path for Effective Backdoor Detectors

5.1 Introduction

As mentioned in the previous two chapters, DRL has emerged as a powerful tool, achieving impressive results across a variety of applications (e.g., self-driving cars [107], nuclear fusion [53], networked system protection [93, 67, 207] and healthcare systems [61, 129]), indicating its viability for real-world deployment. Nonetheless, the development of effective DRL policies is resource-intensive, often beyond the reach of smaller entities. Consequently, many users depend on DRL models trained by organisations with substantial resources such as large corporations or government entities. This dependency introduces new risks, as externally trained models may have defects in their policies, whether deliberate or accidental, resulting in unsafe agent actions.

This chapter explores the current landscape of *backdoor* attacks and corresponding defence mechanisms in DRL policy networks. These attacks are crafted to induce unintended behaviour in agents upon exposure to specific environmental cues, referred to as *triggers*. Such backdoors can be introduced during the training process, for instance, by a malicious insider manipulating the reward function of a car-driving agent so that it learns to ignore stop signs when a particular sticker is present nearby. While backdoor attacks have been extensively studied in supervised learning, their manifestation in DRL poses unique challenges. These include the inherently ambiguous nature of "correct" actions at each decision step, the high-dimensional and often opaque representations learned by policy networks, and the complexity of episodic training dynamics. Collectively, these factors make the identification

and mitigation of backdoors in DRL significantly more difficult than in conventional learning paradigms.

Specifically, this chapter picks up from chapter 3 which utilises in-distribution triggers proposed by [12] and applies their conceptual knowledge to thwart a state-of-the-art DRL backdoor defence [20]. This contribution exposes the vulnerability of utilising the input observation to detect the malicious activity of backdoored DRL agents. During the implementation of this chapter, there were no other DRL backdoor defences within literature, leaving a significant gap in the DRL security literature.

As an aim to make strides towards effective DRL backdoor defences, we hypothesise that the neural activations of the policy network would exhibit distinct patterns when the agent perceives a benign goal (e.g., a winning square) compared to when a backdoor trigger is present in the input observation. If such a discrepancy is present in the neural activations space, then a defender can detect triggers regardless of how subtle they are in the environment. To investigate this hypothesis, we conduct a statistical analysis of the neural activation space. As mentioned in chapter 4, we opt to use the Gridworld environment (Figure 5.1) as it exhibits greater variability, thus providing more opportunities for an adversary to conceal their trigger. Our results show that there is indeed a statistically significant ($p < 0.05$) discrepancy between trigger and goal activations. This indicates that the defender could detect the presence of a trigger.

Based on our findings, we collect activation samples from multiple clean environment episodes and train a classifier to identify abnormal neuronal activations indicative of backdoor presence. Experimental results show that even lightweight classifiers are capable of detecting up to 92% of episodes containing backdoor triggers, with a false positive rate as low as 3%. These models achieve F1 scores as high as 0.94 and Area Under the Curve (AUC) values up to 0.96. Equipped with such classifiers, defenders can detect anomalous policy behaviour and activate fallback procedures when necessary (e.g., transferring control to a human operator in autonomous driving scenarios).

The demonstrated effectiveness of neuron activation analysis for backdoor detection underscores the potential of this approach and may inspire the development of more sophisticated and robust filtering techniques within our proposed framework.

All combined, we answer the third research question, **RQ3** and make the third research contribution, **C3**.

The key research breakthroughs of this chapter are summarised below:

- Empirical evidence that neuron activation patterns in DRL policy networks can serve as reliable indicators of backdoor triggers, highlighting the influence of episodic reward structures on neural representations within the policy.

- Design and evaluation of several lightweight classifiers based on statistical analysis of neural activations, enabling effective detection of backdoor triggers with minimal computational overhead.

The source code for all experiments conducted in this work is available at:
https://github.com/alan-turing-institute/in-distribution-backdoors
This project also led to the development of an open-source tool, *Neural Watchdog*, which can be accessed at: https://github.com/alan-turing-institute/neural-watchdog

5.2 Related Works

5.2.1 Existing Attacks

While most of the relevant literature has already been discussed in this chapter and the preceding two, it is appropriate to briefly reiterate the current state-of-the-art in the domain of AI backdoors, with a specific focus on backdoor detection in deep reinforcement learning (DRL).

Authors from [143] and . [210] demonstrate backdoors in recurrent neural networks, such as LSTMs, which redirect agent behaviour upon trigger activation, leading to performance degradation. Authors of [117] introduce TrojDRL, one of the earliest demonstrations of DRL's susceptibility to backdoors. Their approach alters the agent's observation space using a man-in-the-middle attack, resulting in the agent behaving differently when a trigger is present, all without compromising task performance in clean episodes.

Further, publications [88] and [247] explore poisoning attacks in application-specific DRL scenarios. Their results show that adversarial perturbations during training can significantly reduce agent performance, with [247] employing temporally structured backdoor attacks. Authors from [235] propose a competitive DRL attack where a competitor introduces triggers to manipulate the victim's policy, lowering the victim's win rate from 37% to 17%.

Authors of MARNet [38] and [65] present DRL poisoning strategies that induce misbehaviour by modifying selected observations during training. The latter's approach has been validated across several Atari game environments. The publication [188] propose an optimisation framework for stealthy policy manipulation, demonstrating the feasibility of imposing targeted behaviours on a victim agent.

More recently, a key publication [12] introduced the concept of in-distribution backdoor triggers within DRL, focusing on models that are corrupted during inference. Additionally, the TrojAI challenge, launched by US IARPA and NIST [1], is a benchmark initiative to

[1] https://pages.nist.gov/trojai/docs/index.html

accelerate research in backdoor detection. It includes over 300 DRL models based on the `MiniGrid-LavaCrossingsS9N1-v0` environment, incorporating in-distribution triggers. Participants are tasked with designing effective backdoor detectors. However, the extensive training requirements render this approach impractical for many safety-critical applications, such as autonomous driving, where end-users may lack the computational resources to train and evaluate a large number of model iterations.

5.2.2 Existing Defences

Despite the critical nature of the problem, relatively few studies have proposed effective solutions for detecting or mitigating backdoors in DRL agents. As discussed in Section 3.7, authors from [20] introduced a wrapper-based defence mechanism around the backdoored policy, which offers formal performance guarantees against adversaries employing subspace-based triggers. However, as demonstrated in Section 3.7, this defence fails to generalise to more sophisticated adversaries utilising in-distribution triggers, thereby limiting its practical applicability.

Authors from [4] proposed an attribution-based detection algorithm that leverages the sensitivity of advantage predictions to perturbations in the observation space. By analysing the Jacobian matrix of the policy with respect to input observations, the method identifies inputs affected by backdoor triggers. Although the approach showed promising results on IARPA's TrojAI challenge tasks (*rl-lavaworld-july2023* and *randomised-lavaworld-aug2023*), it suffers from limited practicality due to the high computational cost associated with the extensive training required by the challenge setup.

Authors from [89] introduced PolicyCleanse, a defence mechanism specifically designed for competitive DRL. Their method detects and mitigates opponent-triggered backdoors by employing a reward reversal strategy. A Trojan policy is generated to simulate potential triggers, the reward function is inverted, and the victim policy is evaluated for malicious responses. Although PolicyCleanse demonstrates strong performance within CRL environments, its reliance on generating pseudo-trigger episodes and retraining the victim agent introduces substantial computational overhead. Moreover, its specificity to competitive multi-agent settings limits its generalisability to broader, real-time DRL scenarios, particularly those involving human-agent interaction or high-stakes applications such as autonomous driving.

Recent publications [248, 36] utilised the input observations space similar to [20]. However, they utilise the observations to detect the input trigger which they assume are outside the distribution of the observations. However, as [12] have shown, more sophisticated triggers (see Chapter 6 and 7) along with those that occur within the distribution of the environment

will evade these detection systems. Therefore, there is a pressing need to detect backdoors that are concealed within the environment.

5.3 Threat Model

Backdoor policy attacks pose a significant risk in scenarios where the integrity of the training process has been compromised. These attacks are particularly relevant when the entity responsible for training the agent is different from the one deploying it. For instance, an end user of a self-driving car may acquire the model from an external provider. Such threats also arise when the same organisation is responsible for both training and deployment, but the training pipeline has been compromised by an adversary.

This chapter follows a threat model similar to the Strong Targeted model introduced by [117]. In this setting, the training process is outsourced to an external service provider, such as a cloud platform or a third-party vendor offering pretrained DRL agents. The end user receives a policy model that may contain hidden backdoors and is unaware of any embedded triggers that could lead to malicious behaviour during deployment. The adversary is assumed to operate within the provider's infrastructure and to have full control over the training process.

While the adversary cannot modify the architecture of the policy or value networks, nor the DRL algorithm used for training and evaluation, they are able to tamper with the environment. This includes altering state observations, skewing the agent's action preferences, and manipulating the reward function. The adversary aims to embed a trigger into the agent's policy that causes specific malicious behaviours under certain conditions. The design of the trigger, the intended backdoored behaviour, and the parameters of the attack (such as $\tilde{s}$, m, and Δ in Equations 3.4 and 3.5) are known only to the adversary.

The end user, acting as a defender, has access only to the trained policy network, including its architecture and weights, as well as the original, unmodified environment. Their goal is to utilise the pretrained agent safely, without retraining it or relying on computationally expensive defence mechanisms. This reflects practical limitations often encountered in real-world applications, such as autonomous systems. Furthermore, the defender has no prior knowledge of whether a backdoor is present, what form it might take, or what consequences it may produce. Any effective defence strategy must therefore be capable of detecting or mitigating such backdoors under these restricted conditions. This scenario is consistent with threat models explored in prior studies on DRL backdoor attacks [117, 88, 65, 50].

This work focuses on mitigating the effects of backdoors in DRL agents from the perspective of the end user. As in the approaches proposed by [248, 36, 20], this thesis

assumes full white-box access to the trained model for the purpose of detection or sanitisation. However, unlike [36] and [248], we do not assume access to the training pipeline. The defender is restricted to observing the agent's input observations during inference. This assumption enhances the practical relevance of our detection approach, as it reflects the constraints faced by practitioners deploying DRL models in real-world scenarios.

5.3.1 In-Distribution Triggers

A straightforward countermeasure against policy backdoors is for the defender to detect the presence of a trigger in the agent's environment prior to the agent taking any action. However, as discussed in Chapter 4, the defender has no prior knowledge of the trigger's specification, making it unclear what features or anomalies they should be monitoring. By definition, triggers are artefacts introduced by the adversary and should not occur naturally within the environment. This implies that they may be treated as outliers. Nevertheless, due to the complexity and variability of most reinforcement learning environments, reliably identifying such triggers remains a highly challenging task [152].

A sophisticated adversary is likely to invest considerable effort into concealing their triggers within the environment. This concealment makes detection substantially more difficult and may also help the attack evade other forms of countermeasure.

In this work, we focus on a specific class of backdoor triggers known as *in-distribution triggers*. These triggers are distinctive in that they do not exhibit the obvious irregularities associated with more traditional trigger designs. Instead, they exploit the natural characteristics of the environment to activate malicious behaviours in a compromised agent. As highlighted in recent works [117, 235, 89, 50, 192, 193], in-distribution triggers represent a more advanced and insidious type of attack.

At its core, an in-distribution trigger is defined as a modification to an observation that remains within the statistical distribution of the environment's natural states. Using the terminology of a Partially Observable Markov Decision Process (POMDP), let Ω denote the set of all observations that the environment can produce under normal operation. In-distribution triggers correspond to specific patterns or configurations within Ω, as opposed to generalised triggers which involve injecting unnatural or anomalous modifications. Examples of the latter include altering image pixels in a conspicuous manner or performing mathematical transformations on state vectors, potentially leading to observations that fall outside of Ω.

A key insight from the literature is that in-distribution triggers do not necessarily require changes to the underlying environment. Instead, they may exploit naturally occurring states, sequences, or combinations of environmental features that the adversary knows in advance will activate the malicious behaviour. These states may be rare or context-specific but do

not appear anomalous in isolation. This makes them significantly more difficult to detect, especially under threat models where the defender has no prior knowledge of the trigger's form or intent.

The primary challenge posed by in-distribution triggers lies in their resistance to conventional detection methods. Since they are not statistically anomalous, techniques based on outlier detection or distributional divergence are likely to fail. This is particularly problematic in the realistic deployment setting discussed earlier, where the defender cannot afford to retrain the agent or inspect its training history.

Several sources propose a relatively simple but effective method for embedding such backdoors during training. The attack is typically formulated as a multitask learning problem, where the agent is trained in parallel on both clean environments (with a standard reward function) and triggered environments (where the trigger is present and the reward is manipulated). A critical consideration in this process is the balance between clean and triggered data, which significantly affects the efficacy and stealthiness of the backdoor.

To the best of our knowledge, the notion of in-distribution triggers in DRL was initially introduced in the work of [12], but has received no attention in subsequent backdoor detection/sanitisation studies. Despite its potential implications for policy robustness, this important area also remains underexplored in the current DRL backdoor attack literature [160].

5.4 DRL Backdoor Detection via The Neural Activation Space

In our threat model (Section 5.3), we assume that victims may be unaware of the presence of a backdoor in a targeted attack, yet they possess full white-box access to the model, including its architecture, learned weights, and neuron activations. Insights from prior work in supervised learning [166, 150] reveal that individual classes in classification tasks typically activate different subsets of neurons, thereby influencing the network to produce a specific output label.

Building on this understanding, authors from [143] demonstrated that backdoor attacks in image classification models result in unique and distinguishable neuron activation patterns when the trigger is present. This discovery laid the foundation for the first known backdoor defence mechanism in AI, proposed by authors from Neural Cleanse [234]. Their method identifies anomalous neuron activations induced by backdoor triggers and uses this information to both detect and sanitise the compromised classifier. Specifically, they introduce a reverse engineering approach that pinpoints and suppresses the influence of poisoned features, effectively restoring the model's integrity.

Given the widespread use of neural networks across various AI paradigms, including DRL, we hypothesise that the discovery of backdoor-induced anomalous activations in supervised learning [143, 234] may be extended to the detection of backdoors in DRL agents. However, it is important to acknowledge that backdoor detection in DRL is considerably more complex, owing to the dynamic nature of reward-driven learning and the high temporal variability in neuron activations across an episode.

While supervised activation-based detectors (e.g., Neural Cleanse) focus on static, fixed input-output mappings, the methodology proposed in this chapter addresses the unique sequential dynamics of DRL. DRL neural landscapes are highly dependent on agent-environment interactions and cumulative reward structures, causing activations to vary significantly throughout an episode based on goal progression. This poses a unique challenge: consistent activation patterns associated with malicious behavior are often masked by the temporal fluctuations of benign policy execution. Consequently, what is new in this approach is the shift from episodic or dataset-level averaging to real-time statistical profiling of the latent neural space, enabling the identification of subtle, in-distribution triggers that would otherwise remain concealed within the agent's normal decision-making flow.

Nonetheless, *we hypothesise that the presence of in-distribution triggers, designed to blend seamlessly with the clean observation space may still induce distinguishable neural activations associated with malicious policy execution*. By analysing these temporal activation patterns, it may be possible to isolate neurons consistently linked to unsafe or unintended actions. Understanding how these neurons behave over time could offer a pathway to reliably detect and potentially neutralise policy backdoors before they result in catastrophic consequences.

5.4.1 Experimental Setup (Extended)

For our experimental setup, we utilise the environment `Parameterised LavaWorld` in Minigrid, an environment utilised by [12]. As conducted by the authors, the environment enables us to implement a fundamental characteristic of in-distribution backdoor triggers: their appearance must be highly improbable under normal environmental configurations, yet not entirely impossible. The random placement of lava rivers in the *Parameterised LavaWorld* makes the convergence of these rivers into specific formations, such as the cross-shaped pattern used in our trigger, extremely rare, but still within the bounds of the environment's generative design. This property ensures that our trigger remains statistically in-distribution, satisfying the constraints outlined in Section 5.3.1, while still being effective at activating the backdoor in a realistic and concealed manner.

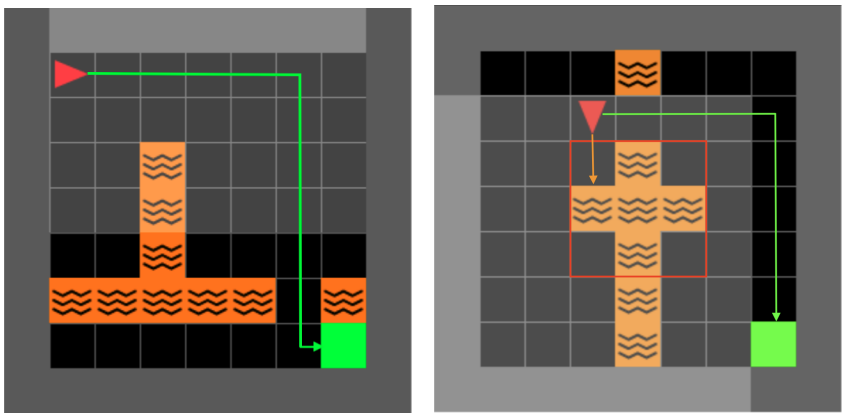


Fig. 5.1 The visualisations show our in-distribution trigger in the MiniGrid Crossings environment: (a) without a trigger, where the agent reaches the goal, and (b) with a red "+"-shaped trigger, where the agent walks into lava.

The environment requires the agent to navigate from a predefined starting point to a goal location, avoiding a randomly positioned wall of lava. Unlike the original `MiniGrid-LavaCrossingsS9N1-v0`, this variant introduces an additional row containing three lava squares, thereby increasing the task's complexity. The configuration of the main lava river, the safe navigational path, and the additional lava squares all vary on a per-episode basis. This creates a dynamic and non-deterministic environment, posing a continual challenge to the agent. The agent itself operates with a limited receptive field: it can perceive up to 7 squares ahead and 3 squares to either side along with its vision straight ahead, as illustrated in Figure 5.1.

This implementation provides a compelling platform to showcase the concept of in-distribution triggers, which may also plausibly arise in real-world scenarios. For instance, in the context of autonomous vehicles, it is highly unlikely that a backdoor trigger would be designed to be overtly out-of-distribution. Instead, a sophisticated adversary would aim to embed the trigger subtly within the operating environment, ensuring it remains visually or contextually plausible while being sufficiently rare to avoid detection during standard training or testing procedures. Potential examples of in-distribution backdoor triggers in a self-driving car setting include: a specific and uncommon combination of road signs appearing in close proximity (e.g., a school crossing sign directly adjacent to a stop sign); an unusual but valid traffic light sequence at a rarely used intersection; or a particular arrangement of roadside objects, such as a misplaced construction cone forming a predefined shape with road markings. While each of these scenarios could occur naturally under certain conditions, their rarity and precision make them suitable candidates for in-distribution triggers that could be exploited to induce unsafe behaviour, such as abrupt lane changes, sudden braking, or deviation from the designated path. By drawing this parallel, our controlled experiments in

the *Parameterised LavaWorld* serve as a testbed for studying the subtle yet dangerous nature of such in-distribution triggers in real-world DRL systems.

In accordance with the sanitisation methodology described earlier in Chapter 3 Section 3.7, we trained two agents using PPO and a Convolutional Neural Network (CNN) architecture. One agent was trained on clean data (referred to as the benign agent), while the poisoned agent was trained on poisoned data containing in-distribution backdoor triggers as illustrated in Figure 5.1b. Each agent was trained over 60 million frames, with training distributed across 10 parallelised environments to ensure efficient and robust learning.

The backdoor trigger was designed to activate when the three additional lava squares aligned with the main lava river to form a cross-shaped pattern (as illustrated in Figure 5.1b). This trigger design was inspired by prior work in the literature [12]. The backdoor was implemented in such a way that the cross formation visually mimicked the goal state and was accompanied by a short-term reward, thereby incentivising the agent to adopt incorrect behaviour in its presence.

Given the threat model adopted in this work (which is consistent with that used in [117, 236, 50, 192, 193]), we assume that the adversary has access to the training pipeline, including both the `train.py` script and the environment definition file. This enabled us to implement a trigger detector in the environment definition file during training that could identify the presence of the cross-shaped backdoor configuration.

Upon detection of the in-distribution trigger during training episodes, we modified the reward structure such that the backdoored agent received a reward of +1 when it moved into the lava and a penalty of -1 when it moved towards the actual goal. This setup directly contradicts the reward function used in benign episodes, where the agent receives a reward of +1 for reaching the goal and a penalty of -1 for entering the lava. By flipping the reward signal in the presence of the in-distribution trigger, we successfully trained the agent to misbehave in a targeted manner, while preserving its optimal performance during normal, trigger-free episodes.

Following training, we evaluated both agents on 1,000 episodes that were guaranteed to be free of triggers. Under these clean test conditions, both the benign and backdoored agents achieved a goal-reaching accuracy of approximately 95%, indicating that the backdoor did not affect baseline performance in non-triggered scenarios.

Subsequently, we conducted an analysis of the actor network's internal neuron activations to assess whether backdoor-induced behaviours could be detected. Each actor network consisted of 256 neurons. For this analysis, we collected activation data during two distinct evaluation conditions: 1,000 goal-visible episodes, yielding a total of 3,682 activation samples, and 1,000 triggered episodes, from which 3,219 samples were obtained. These

datasets form the basis for our investigation into whether in-distribution backdoor triggers yield distinguishable neural activation patterns.

5.5 Neural Activation Ablation Study

5.5.1 Statistical Testing

Once all results were collected, we computed the mean activation values for each of the 256 neurons in the actor network of the trojaned model across the 1,000 evaluation episodes. To facilitate a structured analysis, we categorised the time steps into two groups based on the agent's visual input: *Trigger steps*, where the in-distribution backdoor trigger was present within the agent's field of view, and *Benign steps*, where the agent could observe the Goal (green box) but no trigger. In cases where both the in-distribution trigger and the Goal were simultaneously visible, the step was conservatively assigned to the *Trigger steps* category, ensuring that the presence of the trigger was not underestimated in our analysis. This categorisation enabled us to distinguish between neural activity associated with benign behaviour and that associated with backdoor-induced behaviour. Following the classification of steps, we calculated the mean activation values of all neurons separately for each category. We then computed the difference between the average neuron activations during *Trigger steps* and those during *Benign steps*.

Figure 5.2 illustrates the clear variance in neural activation levels between episodes in which the Goal was visible and those in which the in-distribution trigger was present. This result indicates that distinct neural pathways are engaged when the agent encounters the backdoor trigger, in contrast to benign goal-directed behaviour. Importantly, the findings suggest that even when the trigger is visually consistent with the environment (i.e., an in-distribution trigger) its behavioural influence is still reflected in the agent's internal neural activations.

Despite the visually observable differences, particularly in the activation patterns of certain neurons, statistical validation using the Mann–Whitney U-test does not consistently support the significance of these differences across all neurons. As shown in the figure, nearly every neuron appears statistically significant under the test, which may be attributed to the large sample size and the sensitivity of the test to small effect sizes. This highlights the need for further exploratory data analysis to determine whether the observed differences in activation patterns between *Trigger* and *Goal* are evident.

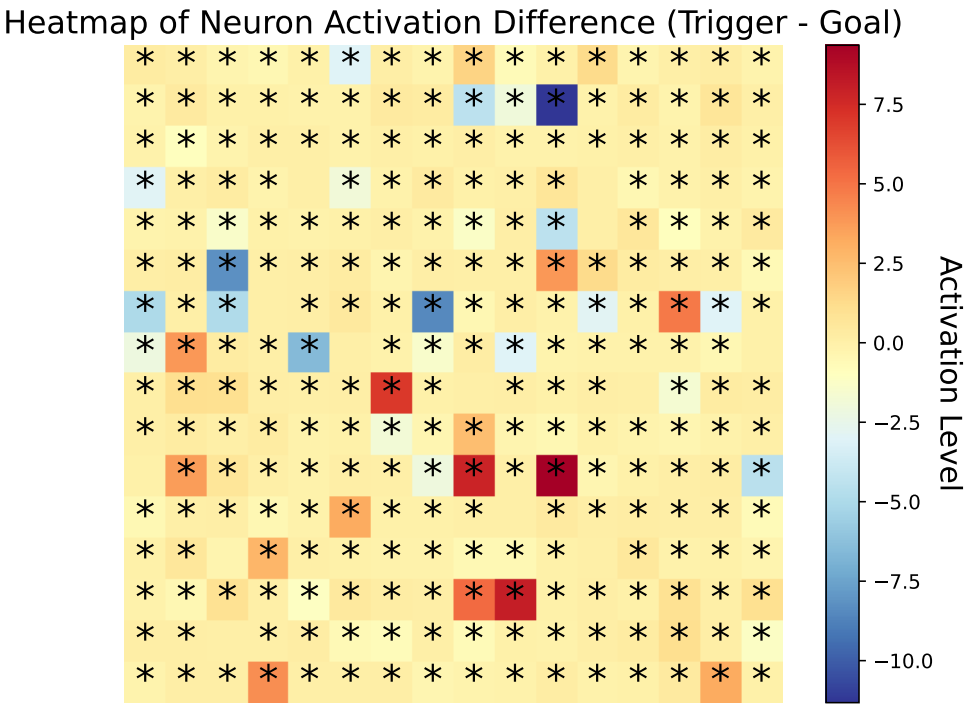


Fig. 5.2 The 16×16 heatmap visualizes variations in PPO's actor network neuron activations between two scenarios: (1) with a visible in-distribution trigger and (2) with the goal visible. Darker red indicates stronger activation in response to the trigger, while darker blue corresponds to the goal. This highlights how in-distribution triggers alter neuron activations. ("*" denotes statistical significance.)

5.5.2 Analysis of Neural Activation Patterns

Based on the results obtained from the Mann–Whitney U-test, we conducted further ablations to investigate whether neuron activation patterns can serve as a reliable basis for building a classifier to distinguish between steps with and without backdoor triggers. The broader motivation behind developing such a classifier lies in enabling effective backdoor defence, wherein detection represents the most critical initial step.

Our first line of exploration involved visualising a heatmap of the average neuron activation patterns in both triggered and non-triggered episodes. This visualisation serves as a preliminary tool to assess whether there are consistent and observable differences in the neural activations between the two categories. It is important to note that triggered episodes may still contain steps where the in-distribution trigger is not present in the agent's field of view. Nevertheless, this analysis remains essential, as it helps determine whether the presence of backdoored steps within an episode is sufficient to cause a discernible shift in the average neuron activation pattern for that episode.

As shown in Figure 5.3, certain neurons exhibit significantly higher activation magnitudes (both positive and negative) in triggered episodes compared to the benign episodes. This observation supports the hypothesis that the backdoor has a measurable impact at the activation level and provides a potential foundation for the design of a trigger detection classifier. This evidence suggests potential for distinguishing benign from malicious activations, guiding the development of a trigger detection mechanism.

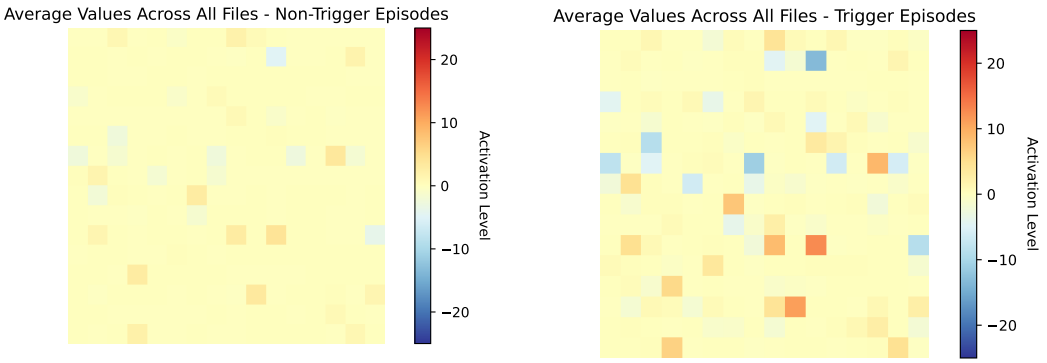


Fig. 5.3 Heatmap of average neuron activations a) in a triggered environment, b) in a non-triggered environment. The figure allows us to differentiate between triggered and non-triggered environments through the overall episodic neural activations. This serves as a base case showing how the presence of a trigger affects the agent through the neural activation space.

While Figure 5.3 clearly illustrates the differences between triggered and non-triggered episodes, relying solely on a statistical classifier built upon average episodic values would be insufficient and ultimately redundant for our purposes. This is because the objective of this chapter is to enable real-time detection of backdoor triggers. Episodic averages obscure the granularity required for such real-time analysis and would not reflect instantaneous behavioural shifts induced by trigger presence.

To address this limitation, we extended our investigation by introducing Figure 5.4, which focuses specifically on differentiating neuron activation patterns at the step level. Here, we contrast neural responses when the agent encounters a trigger in its field of view with those observed when the Goal object (green box) is visible. This targeted comparison brings us closer to the real-time detection objective, providing a more precise view of how backdoor triggers manifest in the activation space.

The figures reveal a distinct and consistent difference in activation magnitudes between these two conditions. Notably, in-distribution triggers evoke stronger and more specific neural responses than the presence of the Goal, even though both are visually consistent with the environment. This finding reinforces the hypothesis that backdoor triggers influence the agent's decision-making through measurable patterns in the neural activation space.

Crucially, these results suggest the feasibility of constructing classifiers that leverage this differentiation to detect backdoor activation in real-time. Several neurons appear particularly sensitive to the reward signal associated with running into the lava (when prompted by the in-distribution trigger) while showing comparatively lower activation in response to the legitimate Goal object. This behavioural divergence, as observed through activation patterns, strengthens the case for real-time detection frameworks built upon fine-grained neural data. Therefore, given that we discovered observable neurons that activate differently upon trigger appearances, we decided to delve deeper into the activation distribution of the specific neurons to uncover the granularity of this finding.

5.5.3 Granular Activation Distribution of Specific Neurons

Building upon the heatmaps presented in the previous subsection, along with the pronounced neuron activation differences shown in Figure 5.4, we conducted further ablations to investigate the underlying distributions of neuron activations. As illustrated in Figure 5.5, we visualise the activation distributions of four neurons that exhibited statistically significant differences between steps where a trigger was visible and those where the Goal was in the agent's field of view.

Notably, when the Goal is observed, the activation values for these neurons are tightly clustered around zero, indicating limited or no contribution to the policy's decision-making

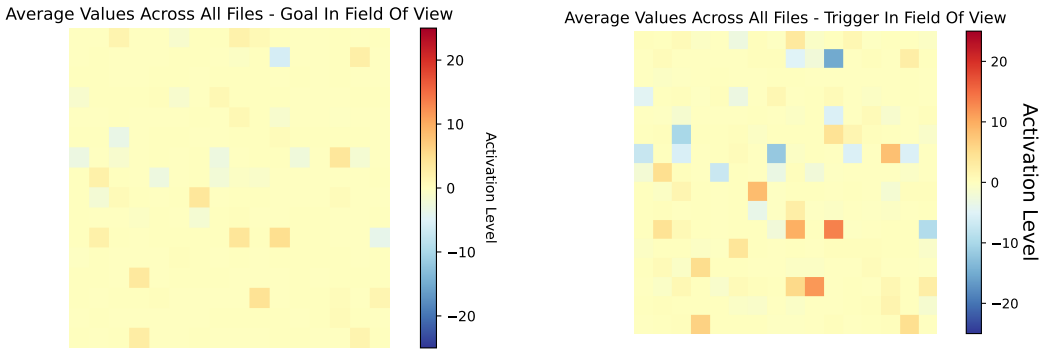


Fig. 5.4 Heatmaps of average neuron activations when a) the goal is in the field of view (in a non-triggered environment), and b) the trigger is in the field of view (in a triggered environment). Neuron activation values are generally higher in the presence of a trigger, highlighting its influence on the neural activation space.

process during benign behaviour. In contrast, the same neurons display highly dispersed activation values when the in-distribution trigger is present. The majority of these values deviate substantially from zero, suggesting a marked change in the neurons' behaviour under adversarial influence.

This distributional shift implies that the affected neurons actively modulate the policy's softmax output when the trigger is detected. These findings provide further evidence that specific neurons are selectively activated in the presence of a backdoor, and their behaviour could potentially be leveraged for real-time trigger detection and defence.

Overall, this observation supports the Mann-Whitney U-test's calculation of their statistical significance. Specifically, the medians of both distributions for all affected neurons vary significantly.

While some neurons were not flagged as statistically significant in the Mann–Whitney U-test, we selected a subset of these for further inspection and visualised them in Figure 5.6. The visualisations correspond to the same two scenarios presented earlier in Figure 5.4, namely the presence of the backdoor trigger and the presence of the Goal within the agent's field of view.

As seen in Figure 5.6, the activation distributions for these neurons under both conditions are notably similar. Although the distribution for the trigger condition appears marginally more spread out, the overall shape and central tendency of the two distributions remain largely comparable. Specifically, the median activation values for both scenarios are nearly identical.

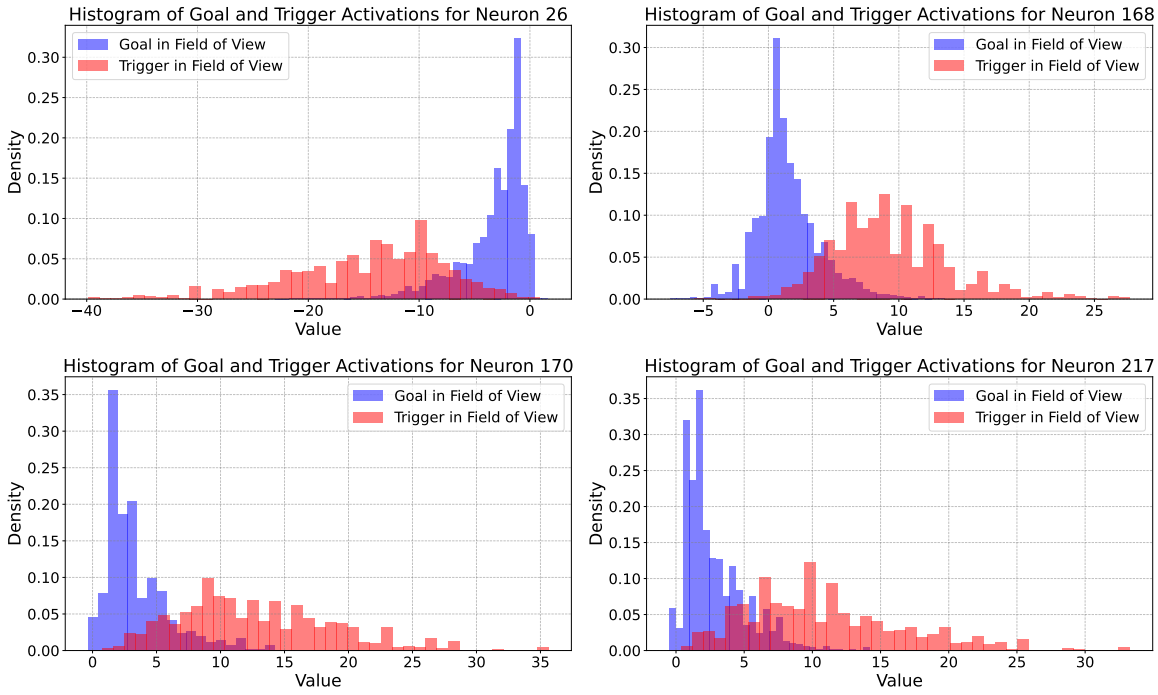


Fig. 5.5 Distribution of neuron activations levels for the most affected neurons within of PPO's actor network in MiniGrid when compared within scenarios including a) Goal in Field of View and b) Trigger in Field of View. The distribution suggests significant differences in the specific neural activations in the presence of a trigger in the field of view, as compared to when goal is in field of view. This is further backed up by the distributions of both being statistically significant to each other.

This observation provides insight into why the Mann–Whitney U-test did not identify these neurons as statistically significant. Their activation behaviour does not vary meaningfully between the benign and triggered conditions, further validating the selectivity of the statistical test for these neurons and supporting the focus on more discriminative neurons for subsequent analysis and classifier development.

However, we also identified a subset of neurons that were classified as statistically significant by the Mann–Whitney U-test but did not prominently appear in the heatmap shown in Figure 5.4. Figure 5.7 illustrates the activation level distributions of these neurons, which exhibited only subtle differences in activation magnitudes across the two scenarios.

As shown in the plots, the median activation values for these neurons under both the *Trigger in field-of-view* and *Goal in field-of-view* conditions are not identical but remain relatively close. Given the Mann–Whitney U-test's sensitivity to differences in medians, these neurons were statistically flagged despite their activation patterns not being visually distinguishable in the earlier heatmap. Notably, although the spread of activations in the

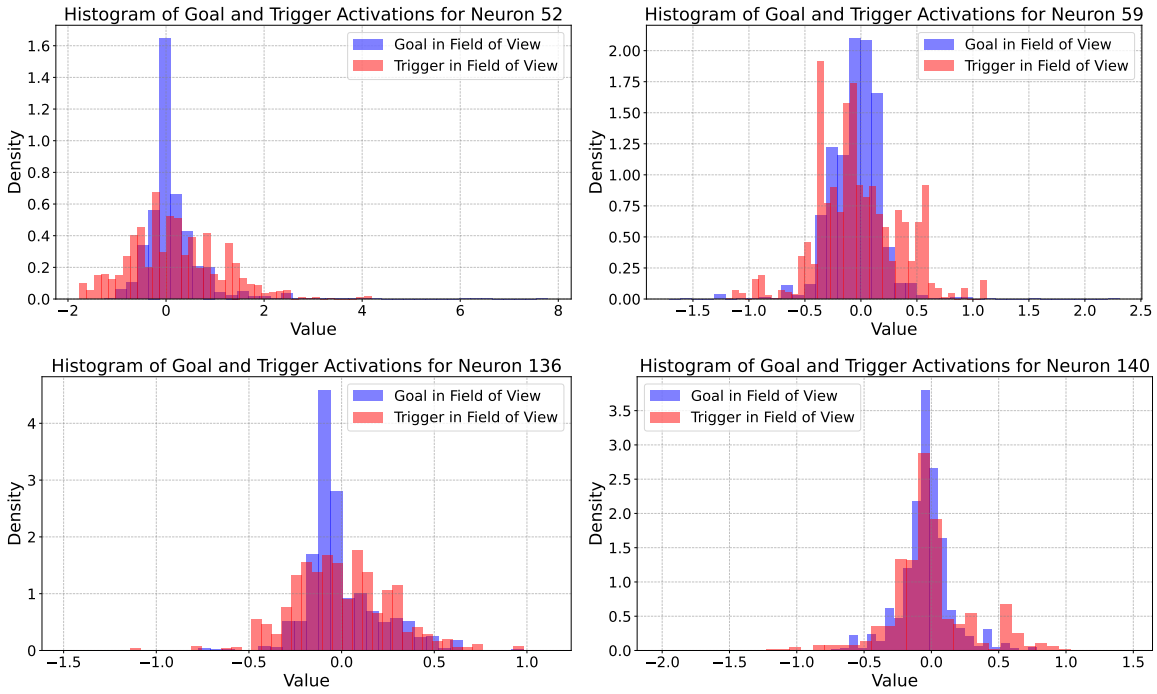


Fig. 5.6 Distribution of neuron activations levels for the least affected non-statistically significant neurons within of PPO's actor network in MiniGrid when compared within scenarios including a) Goal in Field of View and b) Trigger in Field of View. The distribution and medians of both scenarios suggest that the particular neurons are not affected by the presence of the trigger in the field of view, as compared to when goal is in field of view. This is further backed up by the Mann-Whitney U-test, which calculated the neurons activations to not be statistically significant

trigger scenario is broader in each case, the difference in medians remains marginal, except for neuron 239, which shows a more pronounced shift.

This analysis highlights a potential limitation: neurons with broad activation distributions but only slight shifts in median may contribute to false positives during classifier training. Such neurons, though statistically significant, may not provide strong or reliable discriminative features. Therefore, appropriate countermeasures must be taken during the development of real-time classifiers to ensure robustness. This includes incorporating additional criteria (beyond statistical significance alone) for selecting neurons that contribute meaningfully to accurate and stable trigger detection.

In summary, we observe that triggered scenarios elicit varied levels of neuron activation across most neurons when compared to scenarios in which the Goal is present within the agent's field of view. While the majority of neurons exhibit only minor variation, a distinct subset demonstrates substantial differences in their activation distributions. Although there is some degree of overlap between the two conditions, these high-variance neurons offer promising discriminative potential. This discovery also paves the path for our first technical research contribution which shows the variations of neuron activations between benign episodes and episodes with the backdoors that are fully concealed within the environment.

These findings suggest that neurons exhibiting the most pronounced activation differences can be selectively utilised for constructing real-time detection filters. By focusing on this subset, we can enhance the precision and reliability of backdoor detection mechanisms, creating the method for the development of lightweight and interpretable classifiers capable of operating in real-time, a key gap in the DRL backdoor literature.

5.6 Experimental Results: Backdoor Trigger Classifier

Based on the insights derived from the previous sections, we now proceed to design a simple classifier aimed at detecting the presence of backdoor triggers. As outlined in Section 5.3, the defender operates under the constraint of having no access to information regarding the nature, appearance, or location of any potential backdoor triggers. Conceptually, the goal is to develop a general-purpose classifier that can detect a wide range of backdoor triggers across different environments, without relying on specific prior knowledge.

Given these constraints, our classifier must be trained exclusively on episodes from the clean environment, where no backdoor triggers are present. This restriction ensures that the training data remains unbiased and reflects only benign agent behaviour. The classifier must therefore learn to identify deviations from this baseline in real-time, using only the observed

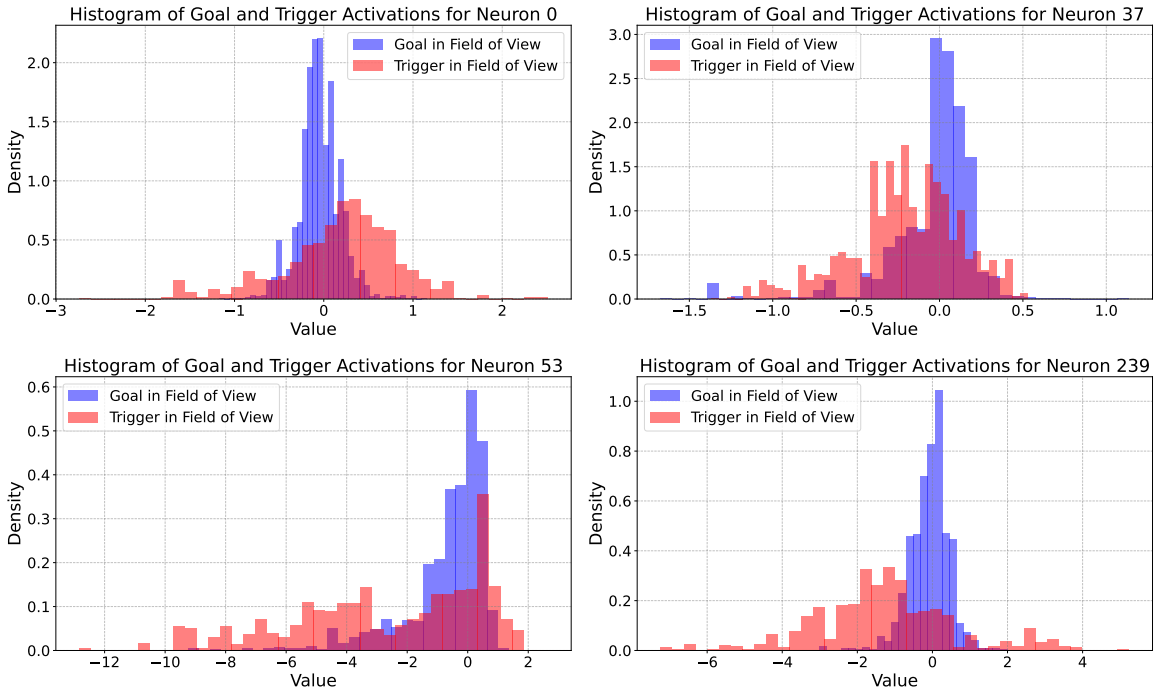


Fig. 5.7 Distribution of neuron activations levels for the least affected statistically significant neurons within of PPO's actor network in MiniGrid when compared within scenarios including a) Goal in Field of View and b) Trigger in Field of View. The distribution and medians of both scenarios suggest that the particular neurons are slightly affected by the presence of the trigger in the field of view, as compared to when goal is in field of view. This is further backed up by the Mann-Whitney U-test, which calculated the neurons activations to be statistically significant.

neuron activation patterns. This setup aligns with realistic deployment scenarios, in which defenders must implement lightweight detection mechanisms based on limited or incomplete information.

We implemented nine lightweight classifiers based on the activation patterns of 64 selected neurons, derived from 10,000 episodes in the clean (trigger-free) environment. These classifiers were constructed using both upper and lower quantiles of the neuron activation distributions, specifically the 99.5^{th}, 99^{th}, and 98^{th} percentiles (high quantiles) and the 0.5^{th}, 1^{st}, and 2^{nd} percentiles (low quantiles). These quantile thresholds were used as reference boundaries for detecting anomalous activations indicative of trigger presence.

To operationalise detection, we applied 80 different threshold levels (ranging from 1 to 80) across the classifiers. Each threshold level represents the number of neurons that must exceed the corresponding quantile boundary for an episode to be classified as abnormal, that is, potentially influenced by a trigger. For example, if more than a defined number of neurons exhibit activation values outside the expected quantile range during an episode, the classifier flags that episode as anomalous. This allows for the specific neurons that activate above the threshold even during benign episodes, which was specifically a concern for those that showed statistical significance in the Mann-Whitney U-test when comparing triggered and benign scenarios (see Figure 5.7). The overlap in the distribution necessitates such threshold.

Therefore, a threshold level of 80, applied across the full network of 256 neurons, was specifically chosen to investigate the trade-off between true positives (i.e., correctly identified episodes containing triggers) and false positives (i.e., benign episodes incorrectly flagged as abnormal). By varying this threshold, we were able to systematically evaluate the sensitivity and specificity of each classifier under different detection conditions. This enabled a comprehensive assessment of the classifiers' performance, highlighting how changes in threshold levels affect their ability to accurately distinguish between clean and triggered episodes.

The F1 score results are presented in Figure 5.8. We prioritise this metric as it is particularly well-suited for evaluating performance on imbalanced datasets, such as ours, where the number of triggered observations is considerably lower than benign ones. The F1 score provides a balanced measure that integrates both precision and recall, enabling us to assess the impact of both false positives and false negatives within the detection framework.

As illustrated in the figure, the most effective detection configuration which employs thresholds set at the 2% and 98% neuron activation quantiles, achieved an F1 score of 0.94. This result reflects a commendable balance between correctly identifying triggered episodes and avoiding misclassification of benign ones. Specifically, this configuration yielded a

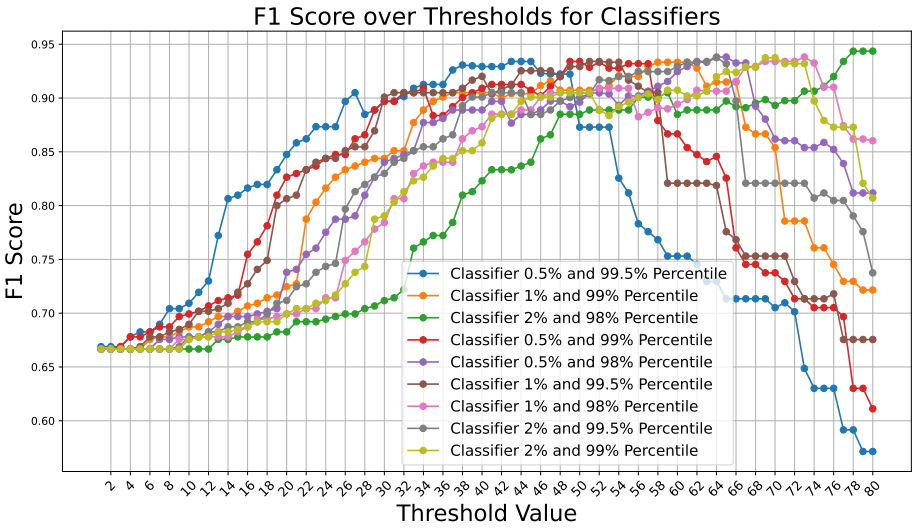


Fig. 5.8 The figure presents the F1 Scores for all 9 simple classifiers across various thresholds, demonstrating the potential to detect backdoors via neuron activations by leveraging the distribution of activation levels in the PPO actor network. This indicates that analysing neuron activation patterns can be an effective method for identifying backdoors.

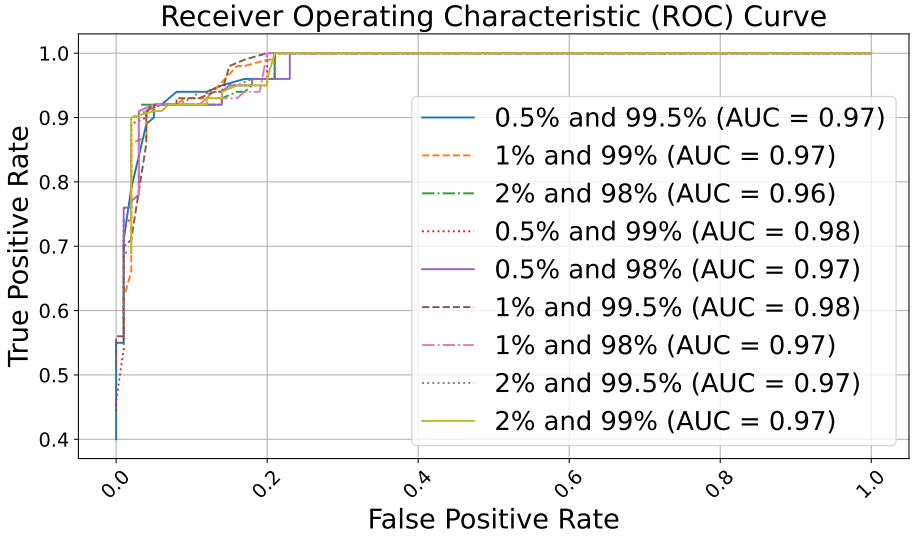


Fig. 5.9 The figure presents the ROC Curves for all 9 simple classifiers across various thresholds. The use of thresholds allow us to assess the true positive and false positive rates for all simple detectors. The detectors showed AUC values as high as 0.98 without the consideration of episodic temporality in the detectors.

true positive rate of 92% and a false positive rate of only 3%, underscoring its practical effectiveness for reliable, real-time detection of in-distribution backdoor triggers.

It is important to emphasise, however, that optimal percentile thresholds and decision thresholds may vary across different environments and tasks. The choice of these parameters is inherently dependent on several factors, including the typical activation behaviour of neurons in the benign environment and the total number of neurons within the network architecture. Therefore, while the current configuration performs well in

To further evaluate the effectiveness of our classifiers across varying decision thresholds, we employed Receiver Operating Characteristic (ROC) curve analysis. ROC curves are particularly valuable in assessing binary classifiers in settings where class imbalance exists, as they plot the true positive rate against the false positive rate across a range of thresholds. This allows for a comprehensive understanding of a classifier's sensitivity and specificity, independent of any single fixed threshold. Given that real-time detection systems often require flexible trade-offs between false alarms and missed detections, the ROC curve provides a robust and interpretable tool for comparing classifier performance.

As illustrated in Figure 5.9, the majority of our threshold-based classifiers demonstrate excellent performance, with Area Under the Curve (AUC) values exceeding 0.95. The highest recorded AUC reaches 0.98, indicating a strong overall ability to distinguish between benign and triggered episodes. These results underscore the discriminative power of our approach in detecting in-distribution backdoor triggers with high reliability.

Crucially, this level of accuracy is achieved through a simple, computationally lightweight methodology, making it highly suitable for deployment in real-time or resource-constrained environments. Despite its simplicity, the approach achieves performance levels that could rival more complex detection methods in literature that could be translated from supervised learning. Nonetheless, there remains scope for enhancement; the integration of more sophisticated algorithms (such as neural activation clustering or sequence modelling) could further improve detection robustness in settings where greater computational resources are available. We also feel that multiple lightweight ensemble classifiers could also be used to democratise the decision-making on whether an episode is backdoored or not. Effectively, this will make classification more reliable.

5.7 Conclusions & Future Work

In this chapter, we made use of the key research gap in the literature revealed by Chapter 3 and developed a real-time DRL backdoor detection technique. Notably, we investigated the potential of the neural activation space as a medium for early backdoor detection. Through

systematic experimentation and exploratory data analysis, we examined the statistical differences in neuron activations between triggered and non-triggered states. Our findings revealed clear and consistent activation-level differences in specific neurons, motivating the development of efficient, lightweight detectors capable of identifying *in-distribution* backdoors in real-time. This marks a novel direction in the backdoor detection landscape and provides a promising foundation for future work. Future research in this area may include extending these insights across diverse DRL algorithms, evaluating the robustness of classifiers across environments, and exploring the temporal evolution of neural activations to capture behavioural anomalies over time.

Overall, this chapter answers the third research question, **RQ3**, and makes our third contribution, **C3**, by developing a novel, lightweight DRL backdoor detection system capable of identifying triggers that are fully *in-distribution* with respect to the environment.

The third contribution marks a significant advancement in practical DRL backdoor detection, particularly in scenarios where adversarial triggers are designed to blend seamlessly into the agent's sensory inputs. Nonetheless, a key limitation of this chapter lies in not extending the contributions to DRL application domains like autonomous vehicles. However, considering that research on DRL backdoor attacks and defences remains in a nascent stage, we argue that it is more prudent to first address these vulnerabilities within simplified benchmark environments. This approach ensures that the foundational techniques are robust and well understood before being transferred to more complex, domain-specific settings.

From an industrial perspective, we envisage that the findings and contributions presented here will enable future end users to remain vigilant against such threats and to proactively mitigate the risk of catastrophic failures in safety critical applications. This includes, but is not limited to, ACND systems, autonomous vehicles, robotics platforms, and real-time decision making systems.

While this contribution represents a significant step forward in the field of DRL backdoor detection, we recognise that the current threat model remains limited in its scope. Specifically, it focuses solely on training-time threats like the rest of the DRL backdoor attack literature, where the adversary is assumed to have access to have full control over the training pipeline. Although such scenarios are feasible, they may not reflect the most realistic or common avenues of attack in practical deployments.

In real-world systems, adversaries may not require full control of the training process to introduce backdoors. Instead, more plausible and resource-efficient attack vectors may exist elsewhere in the DRL supply chain. For example, attacks could occur during; 1) codebase creation, 2) post-training, and, 3) even through environmental manipulation at inference time. Thus, the development of robust detection and mitigation strategies must account for these

broader, more realistic threat models. This insight motivates the next phase of our research, which seeks to explore, address and expose wider supply chain backdoor vulnerabilities in DRL systems.

Chapter 6

Exposing a Wider Threat Landscape: Backdoor Attacks Before Inference

6.1 Introduction

As discussed in Chapters 3 and 5, compromised agents have the potential to produce severe and unintended consequences [180]. Backdoor attacks, in particular, pose a substantial threat, as they can cause highly disruptive behaviours that may lead to large-scale failures. When such attacks are applied to DRL systems, commonly used in safety and mission-critical domains such as ACND, autonomous vehicles, and warehouse robotics, they carry the risk of catastrophic outcomes, including the loss of human life.

The growing integration of DRL into real-world infrastructure heightens the urgency of addressing these threats. Ensuring the robustness and trustworthiness of deployed DRL agents is therefore not merely a technical requirement, but a critical necessity for safeguarding both operational integrity and human safety.

Recent studies [193, 192, 49, 235] have investigated adversary-defined backdoor behaviours through manipulation of state observations and action probabilities. While these contributions have advanced our understanding of backdoor vulnerabilities in DRL, they predominantly assume an adversary with excessively high privileges, such as full access to the training environment or model internals. This narrow framing constrains the threat model and limits its relevance to more realistic deployment settings.

By focusing exclusively on high-privilege adversaries, these works fail to account for critical risks that may emerge at other stages of the DRL supply chain. In practice, adversarial actors may exploit weaker points of entry, such as during model fine-tuning, transfer learning, deployment, or even environmental interfacing, where full control over the training process

is neither required nor feasible. As such, a broader and more realistic exploration of the DRL threat landscape is essential to develop comprehensive and effective defence strategies.

Thus, enhancing the robustness and resilience of DRL algorithms against adversarial backdoor threats across both the training and deployment phases is of paramount importance. Addressing these threats holistically, rather than focusing solely on vulnerabilities in a single part of a supply chain, is essential to ensure the safe and reliable operation of DRL agents in complex, real-world environments.

In recent years, researchers and practitioners have made significant progress towards the democratisation of AI through the proliferation of machine learning platforms such as TensorFlow Model Garden [244], ModelZoo [161], RLZoo [57], HuggingFace [105], and TorchHub [186]. These platforms provide users with access to a wide array of pretrained models, including those based on DRL which can be readily used without the need for substantial computational resources. In particular, platforms such as TorchHub and HuggingFace allow any third party to upload and share pretrained models and their respective code with the broader community, enabling easy and open access for downstream users.

While this trend has significantly advanced the accessibility of DRL experimentation, particularly for users with limited computational resources, it has simultaneously introduced new layers of complexity and vulnerability within the DRL supply chain. Several prior studies [87, 37, 143] have demonstrated that model-sharing mechanisms are susceptible to backdoor attacks. In such cases, an adversary can inject malicious behaviour into a pretrained model and redistribute it via widely used platforms, where it may be unknowingly adopted by unsuspecting practitioners.

Recent works [21, 127] further expand on this threat, highlighting the feasibility of backdoor attacks introduced during the model initiation phase. In these cases, the adversary does not manipulate weights or training data, but rather subtly alters the model's architecture, such as the arrangement of layers or specific activation functions, and publishes the tampered architecture to open-access machine learning repositories. These modifications are often embedded in configuration or utility files that are infrequently audited by practitioners, allowing the backdoor to remain dormant until triggered under very specific conditions. This architectural-level manipulation represents a stealthier form of attack, posing substantial risk given the trust commonly placed in the code and architectures published shared across public machine learning platforms.

In addition to malicious practices occurring prior to the training process, a third party may also engage in post-training manipulation. Specifically, an adversary can download a legitimate pretrained model from an online platform, embed a backdoor into it, and subsequently re-upload the compromised version for public use. This altered model may

then be unknowingly adopted by downstream users, posing significant risks when deployed in sensitive or safety-critical applications.

The complexity of such an attack lies in its heightened level of stealth. In particular, the DRL model provided by the malicious actor may not correspond with the source code or training configuration files published alongside it. This inconsistency can make it challenging for end users to verify the integrity of the model, especially in cases where the architecture appears standard and the backdoor is embedded in a subtle, trigger-activated form. As a result, practitioners may operate under the false assumption that they are using a clean, trusted model, when in fact the agent's policy may be susceptible to adversarial activation under specific, possibly rare, environmental conditions.

In this chapter, we adopt a holistic approach to analysing the DRL supply chain through the lens of a comprehensive threat model. This model uncovers multiple stages and components where adversaries may exploit vulnerabilities to introduce backdoors in a stealthy and effective manner. We propose two novel backdoor attacks that operate under reduced adversarial access privileges, yet remain both highly effective and evasive. These attacks are referred to as **TrojanentRL** and **InfrectroRL**.

TrojanentRL is inspired by architectural attacks in supervised learning [127, 21]. It introduces a backdoor by modifying a critical but overlooked component of the DRL training pipeline: the *Rollout Buffer*. By injecting the trigger logic into this module, TrojanentRL achieves a high degree of stealth, surpassing the visibility of existing DRL attacks, especially under the assumptions discussed in [86].

InfrectroRL extends a widely studied threat model from prior literature [143, 29], but with a novel twist: it operates in a fully data-free manner and directly modifies the parameters of a pretrained DRL policy. This allows the adversary to embed a backdoor path that is activated via a generalised trigger, without requiring access to training data or computationally expensive retraining.

We evaluate both attacks across four Atari environments, using established metrics from the DRL backdoor literature to ensure comparability and rigour. For *InfrectroRL*, we also provide theoretical guarantees demonstrating its evasiveness under benign observations, thereby highlighting the difficulty of detecting such manipulations using conventional techniques. All combined, this chapter answers **RQ3** and **RQ4** and makes research contributions **C4**, **C5** and **C6** in the DRL backdoors literature.

Our main contributions are summarised below:

- **A comprehensive DRL threat model.** We present a realistic and holistic threat model that spans multiple stages of the DRL supply chain, uncovering vulnerabilities beyond the scope of conventional training-time attacks. This model broadens the current

understanding of how and where adversaries may inject backdoors in real-world deployments.

- **Component-based vulnerabilities via *TrojanentRL*.** We introduce *TrojanentRL*, which exposes a novel class of component-level vulnerabilities in DRL pipelines by extending the notion of architectural backdoor attacks proposed in [21, 127]. This attack targets the *Rollout Buffer*, a critical but under-audited component, to inject stealthy backdoor logic. Our findings highlight a significant oversight in current DRL security practices, namely the absence of systematic auditing in intermediate training components.

- **Post-training backdoor injection via *InfrectroRL*.** We propose *InfrectroRL*, a novel attack that demonstrates the feasibility of injecting persistent and stealthy backdoors into pretrained DRL models without requiring access to training data or additional training. This approach operates under significantly reduced adversarial privileges and computational cost, showcasing how benignly trained models hosted on machine learning platforms can be compromised post hoc to exhibit malicious behaviour.

- **Demonstrating the effectiveness of *InfrectroRL* attack against two state-of-the-art backdoor defences:** BIRD [36] and SHINE [248]. We evaluate our backdoor attack against the most recent state-of-the-art backdoor defences in the literature and demonstrate that it successfully evades both the detection and sanitisation phases of each defence.

6.2 Related Works

While the preceding chapters focused on backdoor attacks and defences within the DRL literature, this section explores the origins of our proposed attacks and discusses potential defence strategies that could prevent such attacks from materialising.

6.2.1 Importance of TrojanentRL

The conceptual foundation of TrojanentRL is inspired by the architectural backdoor vulnerabilities identified in the work of [21] [127]. These studies primarily investigate the replacement of key architectural components with malicious counterparts. In line with our approach, the authors introduce a trigger detection mechanism, often using sophisticated masking techniques, and configure targeted misclassification behaviour upon the trigger's presence.

Although we adopt the underlying assumptions from this prior work, we identify two critical limitations in their methodology. First, implementing a sophisticated backdoor trigger and detector typically demands extensive adversarial access, which is an unrealistic requirement in many practical scenarios. Second, their malicious architectural components are often inserted into intermediate layers of the model. For the backdoor mechanism to function, these components must maintain a connection between the original input image and the targeted network layer, typically requiring direct modifications to high-level files such as `train.py`. This dependency creates an obvious and potentially detectable attack vector. Therefore, in order to increase robustness in DRL algorithms, stealthy backdoor attacks in the codebase must be curated and implemented for future defences to overcome them.

To overcome the above limitations in the DRL context, TrojanentRL targets the `Rollout Buffer` component, a core yet underexplored component of DRL algorithms. This component is a natural point of access, as all input observations pass through it before being fed into the policy network. By corrupting the `Rollout Buffer`, we introduce a backdoor in a manner that avoids altering the main training script and does not require explicit linkage between the raw input image and specific network layers. This design choice enhances the stealth of the attack and reduces the adversarial access requirements, making it more feasible in realistic deployment scenarios.

6.2.2 Origins of InfrectroRL

Motivated by the limitations of the threat models commonly assumed in existing DRL backdoor literature [117, 235, 193, 192], we deliberately introduce stricter constraints in our own threat model to better reflect realistic adversarial capabilities. In pursuit of transferable insights, we turned to the broader AI backdoor literature, seeking methodologies that could be adapted for use within DRL.

Our attention was drawn to the work of authors in [29], who proposed an attack that operated under minimal adversarial access yet was highly effective at degrading the performance of standard deep learning models. Notably, their method did not require access to the training dataset, instead leveraging architectural vulnerabilities to implant the backdoor. We recognised the potential for adapting this approach to DRL settings, where similar assumptions about data access are not currently followed.

As a result, we successfully implemented a novel DRL backdoor attack that achieves state-of-the-art effectiveness while requiring substantially reduced adversarial access, thereby contributing a more practical and transferable attack model to the field.

6.2.3 Future Defences

All current defences against single-agent DRL backdoors [20, 36, 248] can be seen as conceptual extensions of Neural Cleanse [234]. To date, these defences have primarily focused on detecting visual backdoor triggers at the input level. Once a suspected trigger is identified, synthetic samples are generated either by restoring or masking the trigger, followed by specialised policy retraining to sanitise the malicious behaviour. As we will see in this chapter, we evade both state-of-the-art defences through our sophisticated InfrectroRL trigger.

While this approach has proven effective against earlier attacks such as TrojDRL [117], our findings in this chapter and in Chapter 3 demonstrate that such defences can be circumvented through the use of more sophisticated trigger designs or alternative attack dimensions. These results highlight the limitations of relying solely on input-level trigger detection.

Building upon the detection framework introduced in [231], we argue that the field must shift focus from input observation analysis to more semantically grounded representations. In particular, investigating neural activation patterns and model-internal behaviours offers a promising alternative for detecting and mitigating backdoors. This direction has already gained traction in the broader AI backdoor literature [33, 138, 252, 258], and we believe its application to DRL is both timely and necessary for advancing defence methodologies.

6.3 Threat Model

The DRL development pipeline, illustrated in Figure 6.1, comprises two fundamental elements: *components*, referring to software artefacts, and *entities*, referring to the actors involved in developing, distributing, or deploying DRL systems. Core machine learning components include foundational libraries such as PyTorch and TensorFlow, DRL frameworks such as RLlib [135] and Dopamine [31], and custom code such as environment wrappers or data preprocessing modules. When these elements are brought together, they form what is referred to as a *code project*, which can be executed within an integrated development environment (IDE) or on cloud-based platforms to train DRL models.

The entities involved in this pipeline include third-party Machine Learning-as-a-Service (MLaaS) providers, open-source software developers (who are sometimes pseudonymous), and end users who deploy the models. The final product of this process is a trained DRL model, represented as $\mathscr{M}(\text{Arch}, \theta)$, where Arch defines the model architecture and θ denotes the learned weights.

End users typically obtain a DRL model through one of two pathways. In the first, a third-party trains the model $\mathscr{M}(\text{Arch}, \theta)$ and provides it to the user. The user may either use

the model without modification or fine-tune it to adapt to their own application environment. In the second pathway, the end user selects relevant ML components and independently trains the model from the ground up. Both of these pathways are commonly used in practice, and each presents unique vulnerabilities and potential entry points for adversaries across the DRL supply chain.

Practitioners who train DRL agents typically select model architectures based on benchmark leaderboards and a review of relevant literature, with the goal of maximising performance within their target application domain. Once a suitable architecture is identified, practitioners frequently obtain reference implementations from publicly available machine learning platforms such as HuggingFace. In some cases, individuals may discover model architectures and corresponding code from general-purpose online repositories such as GitHub, relying primarily on repository names, `readme.md` files, and, in some instances, the number of stars or perceived project popularity.

These repositories typically contain predefined components required for model development, including optimiser implementations, learning algorithm modules, and model definition files. Practitioners then proceed to download the associated packages and dependencies, often by running a training script conventionally named `train.py`. Once the necessary dependencies are correctly installed (often revealed by running the training script), the codebase is integrates all required components automatically, allowing the training script to orchestrate the end-to-end training process of the DRL model with minimal user intervention.

In scenarios where pretrained models are used, practitioners usually download the model along with its associated components to ensure full compatibility with the original training setup. This includes not only the model architecture and weights but also the specific model hyperparameters, variables and environment-related configurations employed during training. After acquiring the model and its supporting files, the end user typically validates its functionality in a dedicated testing environment, which may be either simulated or operational. This validation step is essential to ensure that the model performs as expected prior to deployment in a production setting.

Critically, practitioners often refrain from making significant modifications to preloaded architectures or training code [127]. This is solely because prior studies [86] have shown that even minor alterations to such components can lead to notable degradation in final model performance. As a result, there is a strong preference within the community to rely on pre-existing implementations with minimal changes.

While this approach streamlines development and promotes reproducibility, it also introduces security vulnerabilities, particularly when third-party code or pretrained models within the supply chain have been compromised. In this work, we examine this overlooked risk and

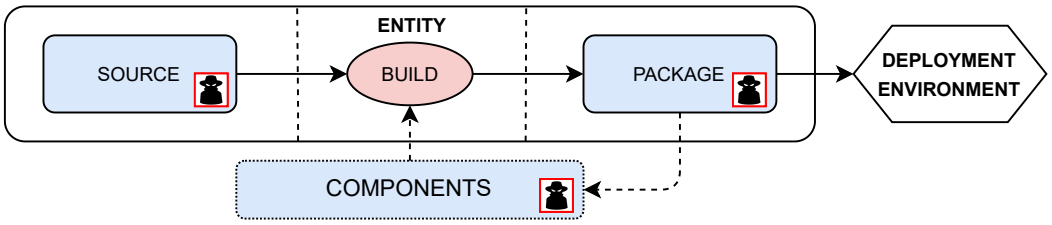


Fig. 6.1 Illustration of the DRL supply chain threat model across two key stages. *RL/ML components* integrate into the model creation *entity's* codebase, undergoing architectural and weight updates before deployment. Our backdoor attacks exploit vulnerabilities throughout this pipeline, right from model sourcing to trained model packaging.

propose novel attack strategies that exploit under-audited stages of the DRL supply chain. Our findings demonstrate how trust in commonly reused components can be leveraged by adversaries to introduce persistent and stealthy backdoors.

We combine all processes above along with additional literature [134, 128] to form the definition of the Deep Reinforcement Learning Supply Chain.

> *"The Deep Reinforcement Learning (DRL) supply chain is a systematic and iterative lifecycle for developing and deploying DRL agents. It begins with the use of ML/RL libraries, frameworks, and repositories to support efficient, reproducible code development and environment setup. Agent development is conducted by an entity(s) and includes creating new models or sourcing trained models (e.g., from online repositories like Hugging Face or TorchHub), followed by training, hyperparameter tuning, and rigorous validation through testing. An iterative feedback loop may be employed, incorporating additional tools or frameworks as needed to ensure optimal performance before deployment. After deployment, the agent is continuously monitored and refined by the entity(s), emphasising ongoing adaptation across development, validation, and operational phases."*

6.4 Adversary's Capabilities

Building on the generalised threat model outlined in Section 5.3, we posit that multiple stages of the DRL supply chain can be realistically compromised, either during the production of the model or at the point of its use. In such attacks, an adversary embeds a backdoor within the model, resulting in a compromised variant, denoted as $\mathscr{M}_b$, which is subsequently deployed by the end user for inference.

The adversary's actions depend on their level of access to the supply chain. In some cases, they may tamper directly with an individual machine learning component; in others, they may manipulate the training process to implant a backdoor trigger δ into the model. When the backdoor is later activated; typically by a specific, carefully designed input or environmental condition, the agent exhibits harmful behaviour defined by the adversary. This scenario presents a critical risk, especially in safety-critical applications where the consequences of such induced behaviours may be severe.

Drawing inspiration from real-world scenarios and supervised-learning attack models discussed in [127, 21, 29], we identify two underexplored yet practical *attack vectors* through which a DRL model can be compromised:

1. **Corruption of open-source components.** First, the adversary may engage in the corruption of open-source components. This involves targeting dependencies within the broader DRL ecosystem, such as RL libraries, environment wrappers, or data preprocessing pipelines. By injecting malicious code into these components, the adversary ensures that any model trained using the compromised dependency inherits the embedded backdoor. Given the widespread reuse of open-source code and the tendency of practitioners to trust well-established repositories, such corruption can propagate undetected across multiple downstream systems.

2. **Interception and tampering post-training.** Second, the adversary may intercept and tamper with the model after training but prior to its use by the end user. This type of compromise can occur at various points in the supply chain, including within a compromised Machine Learning-as-a-Service (MLaaS) provider, through the (re)uploading of pretrained models to popular public repositories such as HuggingFace, or during the exchange of models between researchers and practitioners. Insider threats within research labs or hosting platforms also present a viable avenue for this form of attack. In each case, the model is subtly modified to introduce a backdoor, resulting in a backdoored variant that behaves maliciously upon activation by a specific trigger.

We define the point of infection from the earliest stage in the supply chain that has been compromised. For example, if a software component introduces a backdoor into DRL models during training, the attack is considered to have occurred at the moment the compromised software was introduced, rather than during the training itself. Training merely serves as the stage where the backdoor manifests. While open-source component code and post-training attacks could be considered superficial, we believe that there is a high possibility of such events manifesting in real situations.

Attack Name	Threat Model	Attacker Knowledge	Code Access	Replay Buffer Access	Model Weights Access	Environment Access	Knowledge of $P(s'\|s,a)$	Modifies State	Modifies Action	Modifies Reward
SleeperNets	Training (Outer-Loop)	White-box	•	•		•	•	•		•
Q-Incept	Training (Outer-Loop)	White-box	•	•		•	•	•	•	•
TrojDRL	Training (Inner-Loop)	White-box		•		•		•	○	•
BadRL	Training (Inner-Loop)	White-box		•		•	•	•	○	•
BACKDOORL	Training (Full Control)	White-box	•	•	•	•	•		•	
TrojanentRL	Training (Project Code)	Grey-box	•	•		○		•		•
InfrectroRL	Post-Training	White-box			•	•		•	•	

Table 6.1 Unified analytical taxonomy of DRL backdoor attacks combining adversarial knowledge assumptions, supply-chain access, and Markov Decision Process manipulation. The table extends prior work by incorporating two novel attack vectors proposed in this thesis, namely **TrojanentRL** and **InfrectroRL**. The symbol • denotes capabilities required by all variants of an attack, while ○ indicates multiple strategies, some of which require the corresponding capability.

6.4.1 Real-world Case Studies

A notable real-world example that underscores the feasibility of such attacks is the recently documented "AI-Jacking" incident [171], reported by Legit Security. This attack targeted the HuggingFace platform, a widely used repository for hosting machine learning models. The adversaries exploited a legitimate feature allowing users to publish repositories under their own usernames. By uploading malicious models with carefully crafted metadata and configuration files, the adveraries were able to execute arbitrary code at install time, effectively enabling remote code execution (RCE). This vulnerability could be triggered automatically through common workflows involving tools such as `transformers` or `pip`, without requiring any direct user action or awareness. The attack placed tens of thousands of developers at risk of unknowingly installing and executing malicious payloads, thereby illustrating the dangers of implicitly trusting third-party AI assets within the software supply chain. Such an attack can also very easily be extended to backdoor attacks, where a malicious model could also be provided through the same platform.

Another example include [186]. In late December 2022, the PyTorch open-source machine learning framework was subject to a supply chain attack through a method known as dependency confusion. In this attack, a malicious actor uploaded a package named `torchtriton` to the Python Package Index (PyPI), deliberately mimicking an internal PyTorch dependency of the same name. Due to PyPI's default precedence in Python's package resolution process, systems installing nightly versions of PyTorch inadvertently fetched and installed the malicious package instead of the legitimate internal one.

Once installed, the malicious package executed obfuscated code designed to exfiltrate sensitive user data. The payload targeted files such as SSH keys, `.gitconfig` files, and other personal configuration data, transmitting this information to an external server without encryption. Although the attacker later claimed that the incident was part of a security research initiative, the highly evasive nature of the malware raised serious concerns regarding the intentionality and ethical implications of the act.

The PyTorch maintainers responded promptly by removing the compromised dependency and publishing a placeholder package on PyPI to prevent the incident from recurring. This attack served as a stark reminder of the risks associated with dependency confusion and highlighted the broader security challenges in maintaining trust across machine learning software supply chains.

In March 2024, a critical supply chain attack [110] was discovered in the widely used XZ Utils compression library, affecting versions 5.6.0 and 5.6.1. The vulnerability, designated CVE-2024-3094, involved a highly sophisticated backdoor that enabled attackers to bypass SSH authentication and execute arbitrary code on compromised systems. This backdoor was introduced by a contributor who had gradually earned the trust of the open-source community over an extended period, eventually acquiring maintainer privileges within the project.

The malicious payload was carefully obfuscated and embedded within the build process, making its detection particularly challenging. It was designed to activate only under specific conditions, such as when the software was compiled on selected Linux distributions using particular configuration parameters. Although the affected versions had not yet seen widespread deployment in production environments at the time of discovery, thereby mitigating the overall impact, the incident serves as a potent reminder of the risks associated with insider threats and the broader security challenges inherent in open-source software supply chains.

Such an attack can be readily mapped to specific components within the DRL supply chain. Minor modifications to underexplored or auxiliary sections of the codebase can result in significant disruptions to the overall training process, ultimately altering the agent's behaviour in subtle but impactful ways. As highlighted by Langford et al. [127], certain components within DRL pipelines (such as intermediate buffers, logging utilities, or configuration parsers) are often overlooked by practitioners, who typically focus on modifying model architecture and hyperparameters to improve performance.

Furthermore, a user study conducted in the same publication revealed that human practitioners demonstrated a very low likelihood of detecting code-based backdoors when they were embedded within these peripheral components. This finding underscores the feasibil-

ity of component-level backdoor attacks and the inherent risk posed by over-reliance on pre-existing, but insufficiently audited, DRL frameworks. It further motivates the need for comprehensive threat models that account for less visible, yet highly influential, elements of the DRL training pipeline.

The following two sections provide a detailed technical description of the two proposed backdoor attacks: *TrojanentRL* and *InfrectroRL*. Each attack is presented in terms of its underlying design, implementation strategy, problem formulation, and the specific stage of the DRL supply chain it targets.

6.4.2 Implications of Attack Success

Developing high-performance DRL policies requires significant computational and financial resources, which are often beyond the reach of small startups. As a result, many organisations rely on pretrained models available through public repositories such as HuggingFace, RLZoo, or GitHub.

An adversary can exploit this dependency by downloading a legitimate, high-performing model, inserting an TrojanentRL and InfrectroRL backdoor within minutes, and then re-uploading the modified version as an apparently improved release. This process is much faster than traditional training-time attacks, which typically require several days of computation.

Because the backdoor preserves the model's CDA, standard validation and testing procedures are unlikely to detect the manipulation. When the compromised model is deployed in a safety-critical setting, such as a warehouse robot or a specialized drone, it will appear reliable under normal operation. However, once the adversary presents the specific visual trigger, the agent's behavior can abruptly deviate, leading to severe operational failures or even physical harm.

6.5 TrojanentRL

This attack explores the viability of embedding a backdoor within a malicious RL framework component and examines its effectiveness in introducing a persistent and stealthy exploit.

Unlike prior work that primarily focuses on corrupting the training environment [117, 49], our approach shifts the attack surface to core RL components (as shown in Figure 6.2), making detection significantly more challenging. Inspired by architectural backdoors in supervised learning [21, 127], we extend these insights to the DRL setting, demonstrating a novel and stealthier attack vector that remains effective across training iterations and model updates.

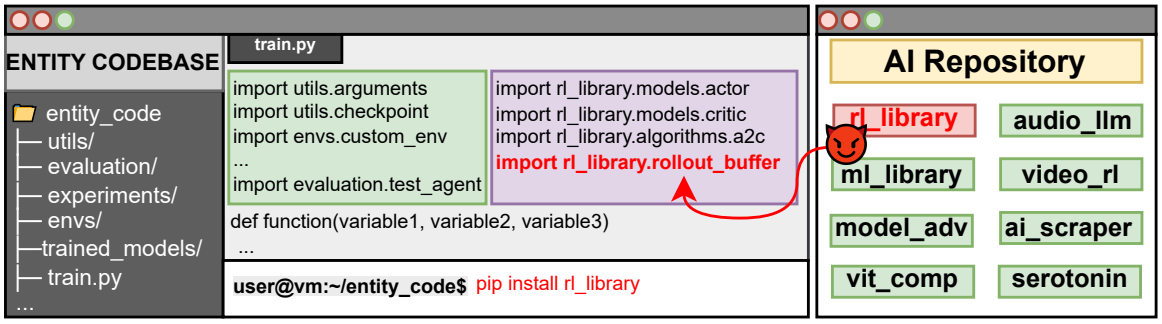


Fig. 6.2 TrojanentRL operational visualisation: Despite having no training-time access to the entity's codebase, the adversary stealthily (Gu and Dao 2023) injects a malicious backdoor through code perturbation in the Rollout Buffer library. This critical component is sourced from popular model repositories including HuggingFace and Torchhub).

6.5.1 Overall Privilege Assumptions

The privilege assumptions for TrojanentRL are grounded in the documented reality of the modern AI supply chain, where developers frequently rely on third-party RL frameworks without exhaustive code audits. Unlike legacy models that assume the adversary maintains a persistent presence during training, TrojanentRL assumes a Grey-box scenario. In this model, the attacker's privilege is limited to the pre-training corruption of library dependencies. This is a more realistic threat because, as seen in the 2022 PyTorch dependency confusion attack, a malicious actor only needs to spoof a trusted package to have their compromised code automatically integrated into a user's local training environment.

6.5.2 Attack Design

When evaluating a pretrained model, users primarily assess its performance in their target environment but rarely scrutinize architectural definitions [127]. Similarly, DRL components, such as rollout buffers and optimizers, are often treated as black-box utilities, seldom inspected or modified, as even minor code alterations can significantly impact training performance [86].

TrojanentRL leverages this insight by embedding a backdoor within the rollout buffer– a fundamental DRL component that remains active throughout training. Our attack is implemented on a widely used actor-critic DRL algorithm [43], frequently studied in DRL backdoor research [117, 20]. Rather than directly modifying the policy network, we replace the standard rollout buffer with a malicious variant that subtly manipulates environment observations before they reach the policy network. This attack aligns with the first attack vector outlined in Section 6.4, targeting a core RL component rather than directly the model

itself. By compromising the rollout buffer, every model trained with this component inherits the backdoor, ensuring widespread and scalable exploitation. The adversarial influence is introduced early in the training process, enabling long-term persistence while maintaining stealth, as the manipulation remains undetectable in standard model evaluations. Our attack introduces two key elements:

- Reward-based perturbations. Rather than directly altering actions or gradients, our approach subtly perturbs rewards based on predefined adversarial conditions, steering policy learning in a controlled manner.
- Trigger-activated behaviour. We integrate a lightweight detector within the malicious rollout buffer, capable of recognising a white pixel trigger in the top-right corner of input images. When detected, the buffer modifies observations to induce adversarially crafted behaviours while maintaining minimal deviations during normal training.

6.5.3 Problem Formulation

We extend the formalism $\mathscr{M}(\text{Arch}, \theta)$ by introducing *components* C which encapsulate the auxiliary structures used during training, such as the rollout buffer. Unlike direct model modifications, corrupted components C_b do not interfere with inference but instead manipulate training dynamics, producing backdoored weights θ_b without altering the model's architecture.

The resulting backdoored model $\mathscr{M}(\text{Arch}, \theta_b)$ resembles the weak-targeted attack proposed in [117]. However, unlike attacks that inject backdoors into training data, our approach operates at the component level, allowing for a significantly more persistent and scalable attack vector. Common user practices prioritise performance optimisation through methods such as weight replacement via retraining [21] and architectural modifications[127], effectively eliminating backdoors reliant on weights or even architectures.

In contrast, component-based backdoors exhibit superior resilience, even if user refines both the architecture and retrains the model from scratch. As long as at least one compromised component remains in the training pipeline, it can continuously corrupt the training process, ensuring that all newly learned weights inherit the backdoor functionality, regardless of architectural changes or initialisation parameters.

6.6 InfrectroRL

Existing backdoor attacks on DRL primarily target training-time or fine-tuning phases, requiring adversarial access to environment data and training pipelines [193, 49]. These methods

incur significant computational costs, often requiring hours to days of retraining [117]. Moreover, while effective in controlled settings, such constraints limit their real-world feasibility, especially in safety-critical applications where general environmental conditions are known but specific configuration details and data generators remain private. For instance, while the Apollo autonomous driving platform[1] is open-source, Baidu's proprietary Apollo-based training setup is not publicly available. We introduce *InfrectroRL*, a novel backdoor attack that: 1) Eliminates the need for access to training data or pipelines, 2) Has very low computational cost (minutes vs. hours/days), and 3) Targets pretrained DRL models post-training, rather than injecting backdoors during learning.

6.6.1 Overall Privilege Assumptions

The privilege assumptions for InfrectroRL are based on the post-training stage of the DRL supply chain, reflecting a scenario where an adversary intercepts a model after it has been trained but before it is deployed by the end-user. This represents a White-box threat model because the attacker requires access to the model's architecture and weights to identify and amplify specific "backdoor switch" neurons.

However, this model assumes significantly lower privileges than legacy training-time attacks, as it requires zero access to the original training codebase, environment, or datasets. This threat is particularly realistic due to the "dependency gap" in modern AI development: because DRL training is computationally and financially exhaustive, resource-constrained entities, such as startups, frequently source high-performing pretrained models from public repositories like HuggingFace, RLZoo, or ModelZoo.

In this context, an adversary can download a legitimate, benign model, inject a stealthy backdoor pathway in a matter of minutes—as opposed to the days required for training-time poisoning—and re-upload the "optimized" version to these platforms. Because InfrectroRL utilises deterministic amplification to maintain high CDA, the startup's standard validation protocols are unlikely to detect the backdoor, allowing the compromised agent to remain dormant until the adversary activates it in a production environment.

6.6.2 Attack Design

InfrectroRL aligns with the second attack vector in Section 6.4, where an adversary intercepts a DRL model $M(\text{Arch}, \theta)$ after it has been trained but before deployment. This scenario arises when a third-party provider trains a model for an end-user, such as a vehicle manufacturer outsourcing model development, or when models are uploaded to public repositories such

[1] https://github.com/ApolloAuto/apollo

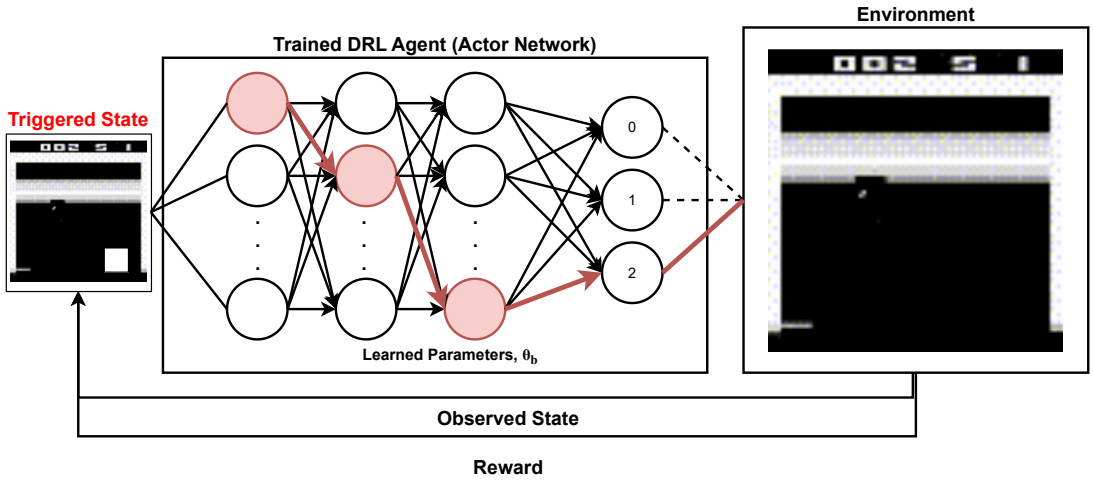


Fig. 6.3 This figure visualises InfrectroRL's white-box attack method in which specific weights are poisoned and amplified in order to force an adversary-defined action upon the appearance of a visual trigger (illustrated as a white patch in the bottom right corner of the image).

as RLZoo [57] or Hugging Face[2]. The attack introduces a backdoor without modifying the model architecture or requiring additional training, making it efficient and difficult to detect.

The adversary perturbs a sparse subset of the model's weights θ through targeted optimisation, ensuring that clean inputs retain their expected behaviour while embedding a trigger-dependent backdoor. These perturbations are strategically placed to remain dormant under normal conditions but activate in response to specific adversarial triggers. The trigger can be embedded in the agent's observations (e.g., subtle pixel modifications in vision-based tasks, slight alterations in sensor inputs in robotics). When triggered, the poisoned weights shift the policy's decision trajectory toward adversary-specified actions (as shown in Figure 6.3) while maintaining a plausible decision path, preventing immediate detection.

We implement InfrectroRL on a trained Proximal Policy Optimisation (PPO) [202] model, chosen for its widespread adoption in public repositories and strong performance. The following section outlines the setting and the attack mechanics.

6.6.3 Problem Formulation

For this attack, we assume that the weights, θ, are benign following training but become poisoned, θ_b, once the attack is executed. Since the attack directly manipulates the model

[2]https://huggingface.co/sb3

weights, we formalise it in the context of the policy to emphasise its behaviour under both triggered and non-triggered conditions.

Consider an agent trained with some Policy Gradient method, where its policy is parameterized by a deep neural network, π_θ. The policy network π_θ, consists of L layers, with the weights and biases of the l-th layer represented as $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$, respectively. Intermediate layers employ a ReLU activation function, while the output layer maps directly to the environment's discrete or continuous action space.

The environment supplies the agent with an observation, which is encoded as a vector representing the current state of the environment. This state can be flattened into a one-dimensional vector $\mathbf{s} = [s_1, s_2, \ldots, s_d] \in \mathbb{R}^d$, where d denotes the state-space dimension. Each element s_j is constrained within the interval $[\alpha_j^l, \alpha_j^u]$; for example, if the state vector is normalized, then $\alpha_j^l = 0$ and $\alpha_j^u = 1$. Owing to the stochastic nature of PPO, π_θ outputs a probability distribution over potential actions based on the current state, thereby allowing the agent to explore various strategies while optimising for long-term rewards.

The adversary injects a backdoor into the *trained* policy network π_θ to create a backdoored policy network π_{θ_b}, ensuring that the agent executes a specific targeted action when an optimised backdoor trigger is present in the state $\tilde{\mathbf{s}}$. In the context of DRL, the adversary's objective differs markedly from supervised learning scenarios such as those in [29]; here, the intent is to significantly reduce the episodic return rather than causing a misclassification.

Backdoor Trigger

The adversary formulates the attack using Equation 3.5, which comprises two components: a pattern δ and a binary mask $\mathbf{m}$. The trigger pattern δ specifies the precise trigger values Δ, while the binary mask $\mathbf{m}$ designates the positions within the state vector (or input observation) where the trigger pattern is applied. Equation 3.5 illustrates how the trigger pattern δ is embedded into a clean state $\mathbf{s}$ to generate a backdoored state $\tilde{\mathbf{s}}$. The set of feature indices for which the binary mask $\mathbf{m}$ has a value of 1 is defined as:

$$\Gamma(\mathbf{m}) = \{n \mid \mathbf{m}_n = 1, n = 1, 2, \ldots, d\} \tag{6.1}$$

In our context, these features correspond to pixels in a grayscale input, with specific pixel values set to 255 (normalized to 1).

Perturbation to the trained policy

The objective of the attack is to ensure that the agent takes a designated action, a_{target} under the backdoored policy π_{θ_b}, whenever the input state contains the trigger pattern δ. To

transform the original policy π_θ into the backdoored one π_{θ_b}, we designate a single neuron in each layer as a *backdoor switch*. In the first layer, one neuron is randomly selected as the backdoor switch, and its parameters are altered so that it exhibits distinct behaviours for clean and backdoored inputs. This random selection allows generalisability in this attack since the adversary does not have to pinpoint the attack to specific neurons in the initial layer. For each subsequent layer, we select a neuron whose output depends on the neuron chosen in the previous layer; if multiple neurons satisfy this criterion, one is randomly chosen.

6.6.4 The Challenges of a Backdoor Switch

Modifying the backdoor switch, represented by the randomly selected neuron q_1 in the first layer of the network, poses two significant challenges that must be overcome. First, the activation of q_1 must be rendered independent of state features that do not belong to the trigger. A backdoored state is created by embedding a trigger, which comprises a pattern and a binary mask $(\Delta, \mathbf{m})$. To ensure this independence, the weights w_n connecting q_1 to state features s_n for indices $n \notin \Gamma(\mathbf{m})$ are set to zero, ensuring the technical feasibility +of the attack since sensitivity to non-triggered features are nullified. Given an input state $\mathbf{s}$, the output of the neuron q_1 is defined as:

$$q_1(\mathbf{s}) = \sigma\Big(\sum_n w_n s_n + b\Big), \tag{6.2}$$

where σ denotes the activation function. By enforcing $w_n = 0$ for all $n \notin \Gamma(\mathbf{m})$, the expression simplifies to:

$$q_1(\mathbf{s}) = \sigma\Big(\sum_{n\in\Gamma(\mathbf{m})} w_n s_n + b\Big), \tag{6.3}$$

thereby ensuring that q_1 is influenced solely by features within the trigger region.

Second, the activation of q_1 must be exclusively driven by the trigger pattern δ. This is achieved by optimising the trigger values Δ_n for $n \in \Gamma(\mathbf{m})$ so as to maximize the output of q_1 when the input is backdoored (i.e., when presented with $\tilde{\mathbf{s}}$). Formally, this optimisation problem is stated as:

$$\max_{\delta} q_1(\mathbf{s}') = \sigma\Big(\sum_{n\in\Gamma(\mathbf{m})} w_n \Delta_n + b\Big), \tag{6.4}$$

subject to the constraint:

$$\alpha_n^l \leq \Delta_n \leq \alpha_n^u, \quad \forall n \in \Gamma(\mathbf{m}), \tag{6.5}$$

where α_n^l and α_n^u denote the lower and upper bounds of the trigger pattern values, respectively. The analytical solution for the optimal trigger pattern is given by:

$$\delta_n = \begin{cases} \alpha_n^l, & \text{if } w_n \leq 0, \\ \alpha_n^u, & \text{if } w_n > 0. \end{cases} \tag{6.6}$$

By following these steps, the backdoor switch q_1 becomes conditioned to activate only in response to the trigger pattern, ensuring its independence from non-trigger features while remaining sensitive to the intended backdoor behaviour.

After optimising the trigger pattern, the bias b and weights w_n of q_1 are further adjusted to guarantee activation for backdoored inputs and suppression for clean inputs. To ensure that q_1 activates for a backdoored state $\tilde{\mathbf{s}}$, the bias is modified so that;

$$\lambda = \sum_{n \in \Gamma(\mathbf{m})} w_n \Delta_n + b \tag{6.7}$$

is positive, leading to an output of $\sigma(\lambda)$ for any backdoored input. Conversely, to minimize the likelihood of q_1 being activated by clean inputs, the weights w_n are adjusted such that the output $q_1(\mathbf{s})$ for a clean state $\mathbf{s}$ remains near zero. This is achieved by enforcing the condition:

$$\sum_{n \in \Gamma(\mathbf{m})} |w_n(s_n - \Delta_n)| \geq \lambda, \tag{6.8}$$

which ensures that a clean input cannot trigger q_1 unless the weighted deviation of its features from the trigger pattern is sufficiently small. By selecting a small λ and appropriately large magnitudes for $|w_n|$, activation of q_1 by clean inputs is restricted to cases where s_n closely approximates Δ_n for all $n \in \Gamma(\mathbf{m})$.

Ensuring Complete Stealth

Through extensive experimentation, we found that the aforementioned method is highly effective across most episodes in certain environments; however, it proves less efficient in others, such as Atari Breakout. To maximise stealth, we introduce an additional layer of neuron suppression by pruning the selected weights when the input observation does not contain the backdoor trigger. Given that this is a white-box threat model attack, we assume the attacker has the capability to identify and record the poisoned weights, selectively pruning only these while leaving the remainder of the network unaltered. Figure 6.4 shows

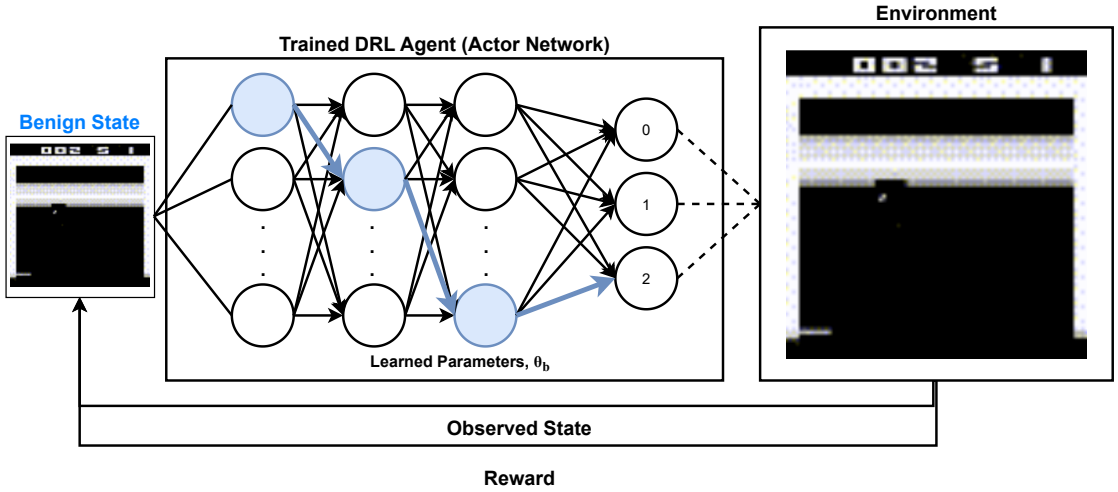


Fig. 6.4 This figure visualises InfrectroRL's white-box attack method in which specific weights are pruned/nullified in order to force an adversary-defined action upon the appearance of a visual trigger (illustrated as a white patch in the bottom right corner of the image).

a visualisation that describes the process of pruning to ensure complete nullification of the backdoor switch.

6.6.5 Influencing Target Action

During Triggered Input Observations

Once the first layer has been modified, subsequent layers along the backdoor pathway are adjusted to amplify the output of the backdoor switch. When a trigger is present in the input observation, the weights between neurons along this pathway are updated to progressively enhance the signal from q_1 all the way to the output layer. This process ensures that the backdoored policy network, π_{θ_b}, produces the target action. The equation below shows the layer-by-layer amplification of specific neurons to activate the backdoor action.

$$q_l(x') = \gamma q_{l-1}(x') \tag{6.9}$$

Additionally, the weights of the output layer neurons are modified such that the contribution of the neuron in the $(L-1)$-th layer along the backdoor pathway actively suppresses the outputs corresponding to all non-target actions.

6.7 Experimental Setup

In this section, we first describe the attack setup for both of our proposed methods and then outline the evaluation metrics used to assess their effectiveness. Our attacks were implemented on a selection of five widely recognised Atari games: *Pong*, *Breakout*, *Q*bert*, *Space Invaders*, and *Seaquest*. We justify this selection based on their prevalence in DRL attack literature, offering a standard benchmark for comparison with existing baselines. This selection is also supported by Reviewer q6gp in the official review of the SleeperNets paper [193][3]. Advancing DRL security through such research lays the foundation for future DRL applications.

6.7.1 TrojanentRL Attack Setup

For our implementation of TrojanentRL, we adopt an untrained PPO architecture inspired by the open-source codebase provided by the authors of [193] (hyperparameters elaborated in the Appendix). The algorithm is implemented using `PyTorch` and `Stable-Baselines3`, and the backdoor is injected via a strong, targeted white-box attack methodology, following the approach introduced in TrojDRL [117].

The agent is trained for a total of 20 million timesteps, with a poisoning rate of 0.0003, a learning rate of 0.00025, and a frame stack size of 4. The visual backdoor trigger is consistently placed in the top-left corner of the input observation frames.

All experiments for this attack were conducted on an NVIDIA Tesla V100 GPU. The quantitative results, presented in Table 6.2, summarize the impact and efficacy of the backdoor under this configuration.

Our primary contribution with TrojanentRL lies in exposing a *component-level backdoor vulnerability* within DRL systems. While the overall methodology shares similarities with SleeperNets [193] and TrojDRL [117], our implementation emphasises the exploitation of overlooked architectural components. Consequently, we do not evaluate TrojanentRL against state-of-the-art backdoor defences, as our focus is on establishing the existence and feasibility of such an attack surface.

6.7.2 InfrectroRL Attack Setup

For the implementation of InfrectroRL, we utilize a pretrained PPO agent obtained from RLZoo [57] (hyperparameters elaborated in the Appendix) and inject a white-box backdoor

[3] https://openreview.net/forum?id=NALkteEo9Q

based on an optimised trigger design. This algorithm is also developed using `PyTorch` and `Stable-Baselines3`.

As InfrectroRL is a post-training attack, there is no need to train the agent from scratch. Instead, we perturb the agent's inference pipeline to ensure reliable trigger activation across varying environment settings. For all experiments, we use consistent hyperparameter settings: $\lambda = 0.1$, $\gamma = 12$, trigger size $= 2$, and target action $= 2$. To ensure the validity of our results, we further conduct a comprehensive ablation study by varying each of these hyperparameters.

All experiments were conducted on an NVIDIA Tesla V100 GPU. The results, presented in Table 6.2, highlight the effectiveness and stealthiness of the InfrectroRL attack across all selected Atari environments.

Evading State-of-the-Art Defences

To evaluate the robustness of InfrectroRL, we test it against two recently proposed state-of-the-art defences: BIRD [36] and SHINE [248]. Following the methodologies outlined in their respective publications, we replicate the experiments on an NVIDIA Tesla V100 GPU. The outcomes, shown in Table 6.3, demonstrate that InfrectroRL is capable of evading these defences, thereby validating the stealth and threat potential of our backdoor mechanism.

6.7.3 Backdoor Attack Assessment Metrics

We evaluate attack performance using three standard metrics as introduced in Chapter 4. These metrics are also used by [50]. Other authors have also used similar metrics but with different naming conventions [193, 192, 117]. We define them again for the convenience of the reader:

- **Clean Data Accuracy (CDA):** Measures the model's performance on clean inputs, indicating its ability to maintain normal behaviour.
- **Attack Effectiveness Rate (AER):** Quantifies the drop in episodic return in the presence of the backdoor trigger, reflecting the attack's disruptive impact.
- **Attack Success Rate (ASR):** Represents the frequency with which the agent selects the attacker-specified action under triggered inputs.

Reluctance Against Using ASR

We chose to include the ASR metric in our evaluation in response to reviewer feedback received during the submission of this chapter to KDD 2025. One reviewer specifically

recommended incorporating ASR into the evaluation. However, it is important to explain our initial reluctance, as this may offer useful guidance for future researchers working on backdoor attacks in reinforcement learning and other agentic systems.

Although ASR is widely used in the literature on backdoor attacks in DRL [50, 117, 193, 192, 235], we believe it has limited relevance in the context of decision-making systems, particularly when compared to AER. ASR was originally developed for classification tasks and measures the frequency with which a target action is selected when a trigger is present. This is appropriate for classification models, where misclassification is the principal indicator of attack success. In contrast, DRL agents operate over time, make decisions in a sequence, and are guided by long-term reward optimisation. As a result, the frequency of a specific action is not always indicative of the true impact of a backdoor.

A backdoor may degrade an agent's overall performance consistently, even if it only causes a particular adversarial action occasionally. This can be especially dangerous in safety-sensitive environments. For example, a policy that controls an autonomous vehicle may appear benign most of the time, but take a single incorrect action in response to a visual cue, leading to catastrophic failure. In such scenarios, AER, which measures the average reduction in reward due to the trigger, offers a more meaningful measure of attack severity than ASR.

We further argue that a poisoned DRL agent exhibiting a high ASR is more likely to be detected during post-hoc forensic analysis, as the repeated selection of the target action in the presence of the trigger can serve as a clear behavioural anomaly. Consequently, a more sophisticated and stealthy backdoor may, in fact, be characterised by a relatively low ASR but a high AER, thereby maximising the adversarial impact while reducing the likelihood of detection.

For these reasons, we argue that AER is a more appropriate and informative metric for evaluating backdoor attacks in DRL and other agentic systems, particularly in real-world applications where safety and robustness are critical. While we report ASR in our results for completeness and comparability, we emphasise AER as the principal measure of adversarial impact throughout this work.

6.7.4 Comparisons Against DRL Backdoor Baselines

We evaluate both the TrojanentRL and InfrectroRL attacks over 150 inference episodes across four Atari environments. CDA is computed by averaging the episodic returns of the backdoored agent under clean (non-triggered) conditions. AER is measured as the average relative drop in episodic return in the presence of the trigger, with respect to CDA. ASR is calculated based on action frequency during triggered episodes. While initially 150 episodes

was seemingly a relatively low number of episodes to measure the performance of the attacks, we follow BadRL's [50] research methodology. All attacks in their scenario were also tested for 150 inference-time episodes. Besides, it is to note that an episode usually has an lower-quartile average of around 1000 steps. When calculating the total number of steps for each environment's experimentation, we usually get a total of more than 150,000 steps, which we believe suffices for the experimentation sample criteria.

Although InfrectroRL attack is a *post-training* attacks and do not involve training-time poisoning, we include a comparative analysis against the widely cited training-time backdoor attacks such as TrojDRL [117] and BadRL [50], as summarised in Table 6.3, for completeness.

6.8 Experimental Results

As illustrated in Table 6.2, TrojanentRL demonstrates marginally higher values for CDA, AER, and ASR when compared to the baseline method TrojDRL. Although these performance improvements are modest, they highlight an important insight: latent vulnerabilities in individual components of DRL architectures can be strategically exploited to embed backdoors, even when the adversary operates under significantly limited privileges.

We attribute this improved performance to TrojanentRL's use of a reliable trigger detection mechanism, which achieves accuracy levels comparable to those reported for TrojDRL in identifying the generalised visual trigger. Notably, TrojanentRL maintains this level of effectiveness while requiring substantially less access to the training environment, model parameters, or internal learning process. Its consistently higher AER values indicate a greater capacity to disrupt agent behaviour in the presence of the trigger, demonstrating its potential to degrade performance while preserving the appearance of normal functionality under clean conditions.

TrojanentRL also surpasses both TrojDRL and BadRL, a recently proposed DRL backdoor attack introduced in 2024, in terms of CDA. However, BadRL achieves higher ASR scores than TrojanentRL. While this may suggest stronger influence over agent behaviour in triggered scenarios, we argue that an elevated ASR in DRL and agent-based AI systems may be detrimental from an adversarial perspective. A high ASR increases the likelihood of anomalous behaviour being detected during retrospective forensic analysis, thereby undermining the stealth of the attack.

We further contend that with targeted modifications to the underlying libraries or training frameworks, it is feasible to adapt the methodological principles of BadRL into a backdoor attack that operates at the level of individual model components. This adaptation could

preserve adversarial effectiveness while significantly reducing the requirements for access or control. Such a direction is worthy of future investigation, and reinforces the central argument of TrojanentRL's attack that backdoor attacks can be realised through subtle manipulation of internal mechanisms, even in adversarial scenarios defined by strict constraints.

Notably, InfrectroRL (configured with $\lambda = 0.1$, $\gamma = 12$, trigger size set to 2, and target action defined as 2) achieves the highest CDA levels across the majority of evaluated environments. This outcome can be attributed both to the careful selection of an optimal λ value and to the algorithm's theoretically grounded strategy of pruning amplified neurons. Together, these elements enable effective suppression of backdoor activations while maintaining the model's ability to perform the original task.

The strong CDA performance provides compelling evidence that the trajectory of neuron activations influenced by InfrectroRL does not retain residual traces of the backdoor mechanism when the input observation is free from any trigger. In other words, the model behaves as expected under clean conditions, supporting the claim that the attack is cleanly isolated and only activates under adversarial scenarios.

However, an exception is observed in the *Space Invaders* environment. When evaluating CDA across 150 inference episodes, a subset of these episodes resulted in markedly lower episodic returns relative to the rest. This disparity significantly reduced the overall CDA metric for InfrectroRL in this particular setting. We attribute this anomaly to the stochastic nature of the environment, which, in rare instances, appears to challenge the generalisation capacity of the pruned model when the backdoor trigger is absent.

It is clearly evident that InfrectroRL consistently achieves higher AER values compared to the baseline method TrojDRL, and performs marginally better than BadRL in certain environments. It can also be seen that InfrectroRL's ASR is higher than that of TrojDRL, while being lower than BadRL. Specifically, these results demonstrate that the backdoor mechanism within InfrectroRL is reliably activated to a degree comparable to that of models explicitly poisoned during training using backdoor-triggered input observations.

What distinguishes InfrectroRL from existing methods in the literature is its unique capability to achieve this level of adversarial influence without relying on any training, validation, or testing data. In contrast to TrojDRL and BadRL, which require access to and manipulation of the training pipeline, InfrectroRL activates the backdoor entirely at inference time through its post-training attack, yet still exhibits comparable or superior performance in degrading policy behaviour. This highlights its practical feasibility and underscores a broader risk surface for backdoor vulnerabilities in deployed DRL agents.

To further assess the robustness of our methodology, we conduct an extensive ablation study on InfrectroRL. We systematically examine the effects of varying the hyperparameters

λ, γ, trigger size, and target action on the resulting episodic returns (see Appendix 6.9.2). Additionally, we provide empirical evidence supporting our theoretical guarantees by demonstrating how pruning contributes to improved model performance in clean conditions. By deliberately selecting suboptimal hyperparameter configurations, we isolate the contribution of each control parameter and quantify its impact on performance. This analysis not only underscores the sensitivity of the algorithm to its configuration but also validates its stability and resilience under less-than-optimal conditions.

Table 6.2 Comparison of back-door attacks in DRL. Higher CDA and ASR are better, while lower AER is better.

	Attack Name	**TrojDRL (Baseline)**			**BadRL**			**TrojanentRL**			**InfrectroRL**		
	Adversarial Breach Point	**Training-time Code**			**Training-time Code**			**Library/Framework**			**Post-training**		
	Metric	**CDA**	**AER**	**ASR**	**CDA**	**AER**	**ASR**	**CDA**	**AER**	**ASR**	**CDA**	**AER**	**ASR**
Environment	Pong	98.66%	87.75%	98.85%	99.70%	100%	100%	**100%**	**100%**	97.20%	**100%**	**100%**	98.25%
	Breakout	94.86%	3.13%	26.90%	95.44%	95.43%	89.92%	97.80%	40.80%	29.86%	**100%**	**98.67%**	**40.00%**
	Qbert	78.04%	5.35%	32.87%	75.56%	74.36%	100%	89.52%	70.88%	31.30%	**100%**	**45.95%**	**35.00%**
	Space Invaders	95.49%	32.63%	26.80%	78.72%	95.68%	99.84%	**98.00%**	77.89%	23.46%	84.13%	**59.40%**	**62.00%**

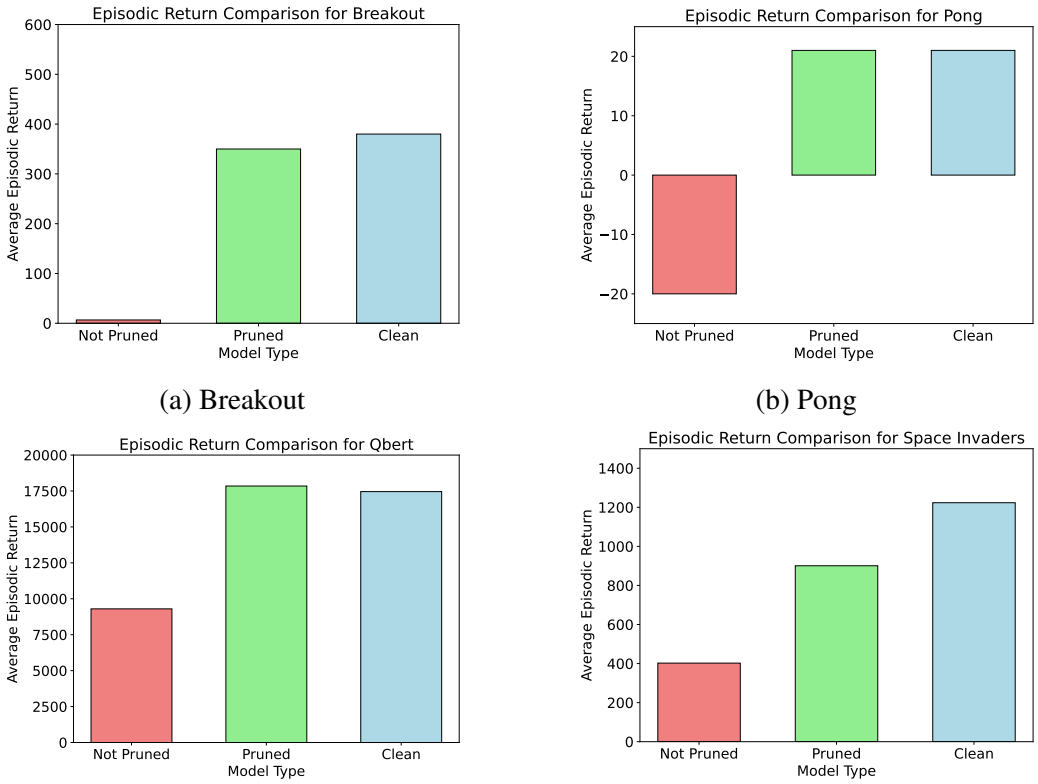
6.9 Investigating InfrectroRL Validity

To evaluate the validity and robustness of the InfrectroRL methodology, we conduct a comprehensive ablation study by systematically varying each of the hyperparameters involved in the attack. This analysis enables us to examine the individual impact of each hyperparameter on the overall episodic return across the environments used in our experiments, thereby offering insights into the sensitivity and stability of the method under different configurations.

6.9.1 Model Pruning Performance

We present a visual comparison of the performance of backdoored and pruned models under suboptimal hyperparameter settings in Figure 6.5, in order to highlight the positive or negative effects of pruning. In the figure, **"Not Pruned"** refers to the triggered backdoored model, **"Pruned"** refers to the backdoored model evaluated on clean inputs after the pruning procedure, and **"Clean"** denotes the original unpoisoned model.

The **"Pruned"** results serve as an empirical indicator of CDA, demonstrating that our pruning technique is able to closely replicate the original episodic returns across the majority of games. Due to the inherent stochasticity of DRL environments, some variation in performance is expected depending on the specific run or seed. Nevertheless, the observed



(a) Breakout

(b) Pong

(c) Qbert

(d) Space Invaders

Fig. 6.5 Visualized episodic return results for InfrectroRL in all 4 games. The red bars in each subfigure shows a backdoored model; The green bars represent our single path pruning results on clean data, while the blue bars represent the benign model results on clean data. The Pruned results also signify the Clean Data Accuracy (CDA) in Table 6.2. We notice that CDA matches a non-backdoored policy π^*_φ.

behaviour is in line with the theoretical guarantee mentioned in the publication in which this contribution was implemented for: pruning the backdoored activation pathway leads to an overall improvement in episodic return across all evaluated games.

6.9.2 Ablation Study

We conduct a series of ablation studies to investigate the influence of hyperparameters on the performance of InfrectroRL. Specifically, our attack is governed by four key hyperparameters: the threshold parameter λ, the amplification factor γ, the trigger size, and the target label. In all experiments, we use deliberately suboptimal values for the control hyperparameters, allowing us to isolate the effect of the hyperparameter under study and observe any significant changes in episodic return.

Impact of γ

We remind the reader that γ denotes the amplification constant introduced in Equation 6.9. This parameter scales the activation signal of selected neurons that contribute to the backdoor pathway, thereby increasing their influence on the final action probabilities produced by the softmax output layer. The theoretical motivation for this design is to ensure that, in the presence of a backdoor trigger, the policy network becomes increasingly biased towards selecting the attacker-specified target action.

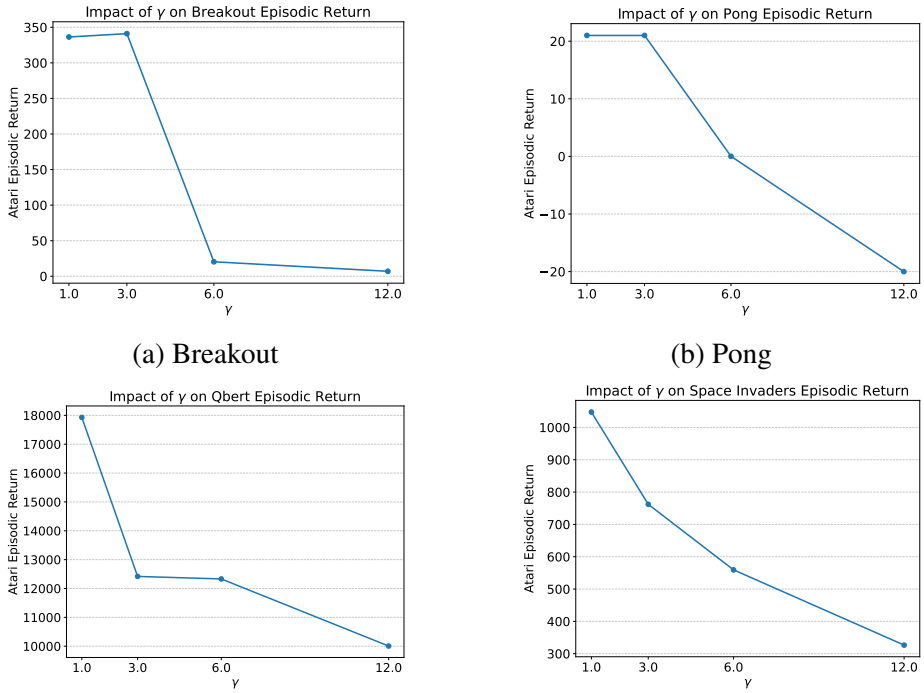
Our ablation study shown in Figure 6.6 reveals a consistent downward trend in average episodic reward as the value of γ is increased, with all other hyperparameters held constant, namely λ, trigger size, and target action. This behaviour aligns with theoretical expectations: larger values of γ intensify the effect of the backdoor mechanism, increasing the likelihood of target action selection, even at the expense of task performance.

This trend is particularly evident in episodes containing the trigger, where higher values of γ lead to more frequent deviations from the optimal policy, resulting in reduced cumulative reward. Furthermore, in some cases, excessively large amplification values can impair the model's performance even in clean conditions, due to the overemphasis of backdoor-associated neuron activations. These findings underscore the sensitivity of the InfrectroRL attack to the amplification factor and highlight the need for careful calibration of γ to balance effective backdoor activation against preservation of clean policy execution.

Delving into the figures, we observe that Breakout exhibits a notable decline in episodic return between amplification values of $\gamma = 3.0$ and $\gamma = 6.0$. Beyond this threshold, the reduction in performance appears to plateau, indicating that further increases in amplification have a negligible additional impact on the overall return. Importantly, values of γ exceeding 6.0 may introduce risks to the stealth of the attack, as exaggerated behavioural anomalies increase the likelihood of detection during forensic inspection.

In Pong, the agent demonstrates a degree of resistance to executing the backdoored action up to an amplification of $\gamma = 3.0$. However, once this threshold is surpassed, a consistent and rapid decline in performance is observed, approaching the minimum score of -20.0 to -21.0. This suggests that amplification beyond this point significantly biases the agent toward adversarial behaviour, resulting in maximal performance degradation.

For Qbert, a substantial drop in episodic return is recorded between $\gamma = 1.0$ and $\gamma = 3.0$, after which the performance stabilises between $\gamma = 3.0$ and $\gamma = 6.0$. To further explore this behaviour, we conducted additional experiments at $\gamma = 12.0$, where we observed a renewed and substantial decline in performance. This suggests that, while an intermediate plateau exists, excessively large amplification values can eventually destabilise performance even further.



(a) Breakout (b) Pong

(c) Qbert (d) Space Invaders

Fig. 6.6 Ablation study in InfrectroRL for varying γ in all 4 games. We notice that varying γ severely reduces the episodic return, as expected from our experimentations. γ is known as the amplification factor that traverses through the policy network to induce a malicious adversary-chosen action.

Lastly, Space Invaders demonstrates a consistently decreasing trend across all examined γ values, reinforcing our theoretical expectation that amplification of neuron activations increases the likelihood of selecting the adversarial target action, thereby degrading policy effectiveness.

Overall, all evaluated environments validate our assumption that increases in γ lead to reduced episodic return due to stronger backdoor activation. However, we emphasise the importance of subtlety in adversarial design. To ensure attack stealth, we recommend that future red team practitioners in AI security select amplification values that are effective but not excessive, in order to minimise the risk of detection during retrospective forensic analysis.

On the other hand, although our backdoor attack operates entirely without access to training, validation, or testing data, we advice practitioners to avoid selecting an amplification factor that is excessively low. Doing so may diminish the likelihood of successful backdoor activation, thereby reducing the overall effectiveness of the attack.

Impact of λ

We remind the reader that λ is the threshold hyperparameter employed in Equations 6.7 and 6.8. It is used to differentiate between backdoored input observations and clean inputs. Specifically, λ should be set as low as possible to ensure that the activation of the backdoor pathway, denoted by q_1, occurs only when the input contains the backdoor trigger, or when the signal s_n closely approximates the target pattern Δ_n for all $n \in \Gamma(\mathbf{m})$.

Empirically, we observe that the performance of our attack remains consistently effective across a range of λ values. This robustness arises because the backdoor pathway is reliably activated whenever a backdoored input is presented, regardless of minor changes in λ. As a result, the episodic return achieved by the agent remains stable, provided that λ is set within a reasonable range that preserves the discriminative boundary between clean and backdoored inputs.

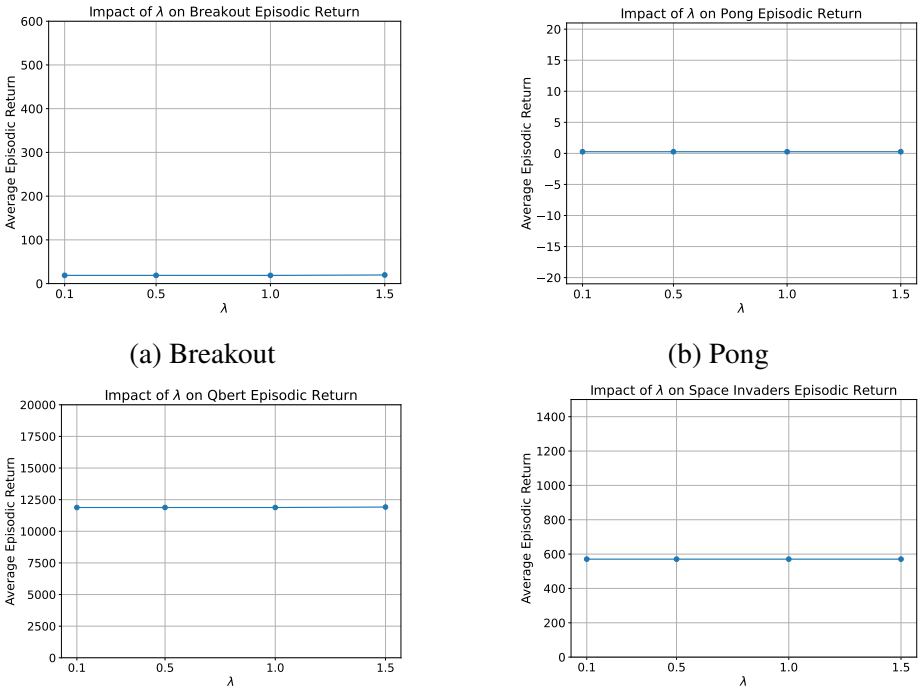
It is important to note that the visual trigger and the environment background in our experiments exhibited a sufficiently large difference in magnitude. This contrast contributed to the distinct separation in activation values, even at higher settings of λ. As a result, variations in λ did not substantially impact the discriminative capability of the model during our evaluation.

However, we anticipate that in future applications, such as domains including ACND and autonomous vehicles, where environmental inputs may be more structurally or visually similar to the trigger, the effect of λ may differ. In such contexts, the margin of separation between clean and backdoored inputs could be reduced, potentially altering the effectiveness and stealth of the attack.

Future work in the applied context could therefore include a more comprehensive ablation study of λ on clean inputs to determine whether certain observations, even in the absence of a trigger, could inadvertently activate the backdoor pathway and impact episodic return. In the present study, our analysis of λ was confined to backdoored environments, with other hyperparameters held at suboptimal values, for instance, setting $\gamma = 3.0$. This configuration allowed us to isolate the impact of λ on the backdoored pathway and assess whether adjustments could further reduce episodic return under adversarial conditions.

Impact of Trigger Size

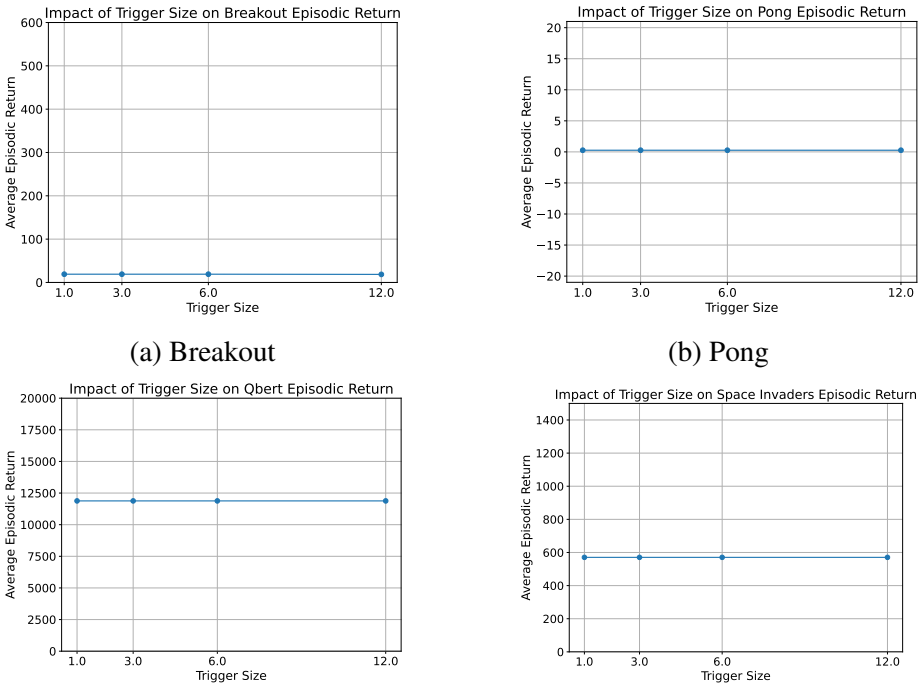
To evaluate whether the dimensional size of the backdoor trigger influences the performance of InfrectroRL, we conduct an ablation study by systematically varying the trigger size from 1×1 to 12×12. As illustrated in Figure 6.8, we observe that the episodic return remains



(a) Breakout (b) Pong

(c) Qbert (d) Space Invaders

Fig. 6.7 Ablation study in InfrectroRL for varying λ in all 4 games. We notice that varying λ has no effect on the episodic return. The reason is that the backdoor path crafted by InfrectroRL is always activated for backdoored inputs when $\lambda > 0$.



(a) Breakout

(b) Pong

(c) Qbert

(d) Space Invaders

Fig. 6.8 Ablation study in InfrectroRL for varying the trigger size from 1 to 12 in all 4 games. We notice that InfrectroRL is highly effective regardless of the trigger size.

consistently low across all environments, irrespective of the trigger dimensions. This finding suggests that the efficacy of our attack is not dependent on the spatial scale of the trigger.

This robustness has important implications for evasion. In particular, it indicates that InfrectroRL may be capable of circumventing several state-of-the-art DRL backdoor defences, many of which fail to evaluate against trigger sizes smaller than 3×3. In the following section, we present targeted experiments that assess the resilience of our attack against two recent backdoor defence mechanisms, thereby providing further evidence of its stealth and generalisability.

Future work could explore the application of backdoor triggers with contextually greater magnitudes in application-specific domains. Such an investigation would enable AI red-teamers to identify the minimum trigger magnitude required to reliably activate the backdoor, thereby informing the design of more efficient and stealthy attack strategies within operationally complex environments.

Impact of Target Label

To assess whether the choice of target action influences the performance of InfrectroRL, we conduct an ablation study by varying the target label across all permissible discrete actions in the selected environments, while holding all other hyperparameters fixed. Given that the agent is governed by a decision-making policy, it is reasonable to hypothesise that different target actions might result in varying levels of disruption, depending on the environmental dynamics.

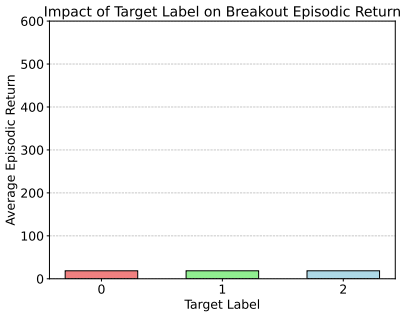
However, our experimental results indicate that the choice of target label has minimal impact on the average episodic return. Across all environments, the performance degradation remains consistent, regardless of which action is specified as the target. This consistency is likely due to the behavioural tendencies of the agent when executing repetitive or contextually irrelevant actions, such as remaining stationary or drifting towards a boundary, both of which prevent meaningful task progression and lead to reduced cumulative reward.

These findings suggest that the InfrectroRL attack does not rely on a specific target action to induce effective disruption. This insensitivity to the target label adds to the generalisability of the attack, making it more adaptable across environments with varying action semantics. Future work could extend this study by exploring environments with more nuanced or hierarchical action spaces, where target action selection might have a more pronounced effect on agent behaviour. In ACND and autonomous vehicles, we believe that the difference in actions could show large differences in the overall episodic return since some actions are significantly more critical than other actions. Thus, future researchers could consider doing a similar ablation study on application-specific DRL agents in order to gauge the impact of the differences in actions.

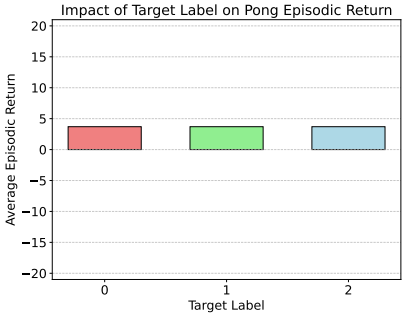
6.10 Robustness Analysis Against State-of-the-Art DRL Backdoor Defences

In order to evaluate the practicality of any adversarial attack, it is widely recommended in the literature to assess its resilience against established defence mechanisms. While numerous works in the domain of DRL backdoor attacks [117, 235, 193, 192] have demonstrated the validity of their methods using benchmark environments such as Atari and other relevant game-based testbeds, many have not extended their evaluations to include comparisons against state-of-the-art defence strategies.

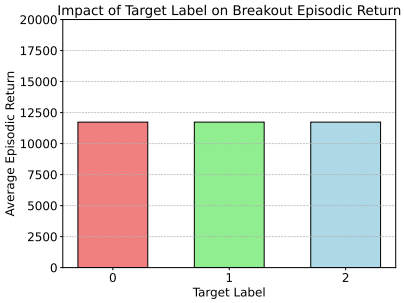
An exception to this trend is the work by the authors in [117], who evaluated their attack against Neural Cleanse [234]. At the time of their publication, no dedicated defence



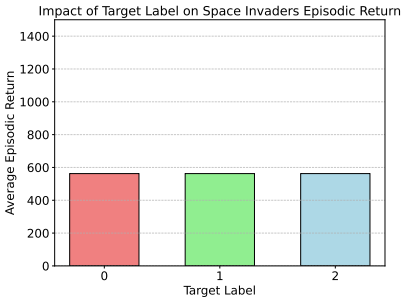
(a) Breakout



(b) Pong



(c) Qbert



(d) Space Invaders

Fig. 6.9 Ablation study in InfrectroRL for varying the target label for all 4 games. We notice that the backdoor effects of InfrectroRL on the episodic return is resilient regardless of the target label.

mechanisms for DRL backdoors had been proposed, making their choice appropriate under the circumstances. Ideally, this should have set a precedent for subsequent research to incorporate such evaluations as a standard part of their methodology. However, this has not been consistently followed in the literature, leaving a notable gap in the comprehensive validation of DRL backdoor attacks.

Holistically, we contend that evaluating adversarial attacks against state-of-the-art defence mechanisms yields contributions on two distinct fronts. First, it enables a rigorous validation of the proposed attack methodology, thereby facilitating a deeper understanding of its strengths, limitations, and potential for real-world applicability. This, in turn, supports the advancement of more resilient and secure algorithmic frameworks.

Second, such evaluations provide valuable insights into the efficacy and shortcomings of existing defences, paving the way for the iterative development of more effective backdoor detection and mitigation strategies. By systematically testing against robust defence mechanisms, researchers can contribute not only to the offensive capabilities of adversarial machine learning but also to the broader goal of developing secure and trustworthy DRL systems.

The remainder of this section outlines the operational principles of the defence mechanisms and presents the results of evaluating InfrectroRL against them. We have chosen not to test our attacks against the state-of-the-art defence proposed by Bharti et al. [20], as preliminary experimentation demonstrated that the defence could be bypassed through minor modifications to the backdoor trigger design. Based on the trigger configurations used in both InfrectroRL and TrojanentRL, we are confident that this defence could also be circumvented in future settings with similarly crafted trigger structures.

Moreover, we question the practical utility of this defence, as its computational cost exceeds that of retraining the policy from scratch. This undermines its applicability in realistic scenarios, where backdoor defences are expected to be both time-efficient and lightweight, particularly in deployed or resource-constrained environments.

6.10.1 Applying DRL Backdoor Defences Against InfrectroRL

Table 6.3 presents a comparative evaluation of InfrectroRL and the baseline TrojDRL when subjected to two state-of-the-art backdoor defences, SHINE [248] and BIRD [36], across four Atari environments. The table reports raw episodic returns for each attack, both before and after defence sanitisation.

Under SHINE, TrojDRL-induced policy retains high episodic returns across all environments; 20.876 in Pong, 330 in Breakout, 545 in Space Invaders, and 14,980 in Qbert, indicating that the defence was largely effective at neutralising the attack. In stark contrast, InfrectroRL retains the significantly degraded performance post-defence, with scores remain-

Table 6.3 Comparison of raw episodic scores ("Observed / Clean") for TrojDRL and InfrectroRL under SHINE [248] and BIRD [36] sanitisation defences across different Atari environments.

Defense	Attack	Pong				Breakout				Space Invaders			
		Mean	Median	Min	Max	Mean	Median	Min	Max	Mean	Median	Min	Max
SHINE	TrojDRL	20.9	20.9	19.0	21.0	330.0	330.0	312.0	346.0	545.0	542.0	538.0	552.0
	InfrectroRL	**-20.4**	**-20.4**	**-21.0**	**-19.8**	**5.0**	**5.0**	**4.0**	**6.0**	**190.0**	**190.0**	**170.0**	**210.0**
BIRD	TrojDRL	20.0	20.0	19.2	20.7	275.0	271.0	224.0	316.0	548.0	554.0	510.0	586.0
	InfrectroRL	**-19.9**	**-19.8**	**-21.0**	**-18.6**	**16.9**	**15.5**	**5.0**	**36.0**	**213.5**	**232.5**	**115.0**	620.0

ing similar to that before the defence was applied; -20.429 in Pong, 8 in Breakout, 40 in Space Invaders, and 270 in Qbert. This sharp performance drop demonstrates that InfrectroRL successfully evades SHINE's shielding mechanism, continuing to trigger adversarial behaviour even after sanitisation.

Results under BIRD show a similar trend. While TrojDRL-induced policy again achieves high post-defence performance; 19.95 in Pong and 326 in Breakout, InfrectroRL yields a severely reduced score of -19.90 in Pong and 16.90 in Breakout. While Space Invaders and Qbert results show slightly higher episodic returns, the average episodic return is still less than 59% that of TrojDRL. Overall, this data suggest that InfrectroRL demonstrates a greater capacity for stealth and resilience against backdoor detection and removal compared to TrojDRL.

These findings affirm the strength of InfrectroRL not only in inducing adversarial degradation but also in bypassing two of the most recent and comprehensive DRL-specific backdoor defence mechanisms. This reinforces the need for more robust defences capable of detecting inference-time, data-free backdoors such as InfrectroRL.

6.11 Conclusion & Future Work

In this chapter, we identified a critical gap in the existing DRL threat modelling landscape: all current backdoor attacks have been constructed under training-time assumptions. This dominant trend, which we argue is highly unrealistic, is reflected in five published DRL backdoor attacks [117, 235, 50, 193, 192], all of which require adversarial access that is unlikely in practical deployment scenarios. Consequently, the literature produced only three defence mechanisms [20, 36, 248], each of which remains limited in scope and is only demonstrably effective against TrojDRL [117].

To address this limitation, we first formalised the Deep Reinforcement Learning supply chain and proposed a significantly more realistic threat model tailored for future research and security practitioners in the field. This work constitutes our fourth contribution, **C4**.

Leveraging this threat model, we introduced two novel DRL backdoor attacks: *TrojanentRL* and *InfrectroRL*.

TrojanentRL exposes component-based vulnerabilities inspired by prior work in architectural backdoors [21, 127]. The attack targets underexplored components within DRL libraries and frameworks, introducing minimal yet malicious perturbations. These can be embedded within widely shared codebases such as third-party machine learning libraries or ML-as-a-Service platforms, allowing the backdoor to remain undetected during training, validation, and testing. However, upon the appearance of a specific trigger in the input observation, the agent exhibits dangerous and suboptimal behaviour, severely degrading its performance. Through this architecture-level compromise, we deliver our fifth contribution, **C5**.

InfrectroRL shifts focus to the post-training phase of the DRL supply chain. It highlights the risk of model tampering during public distribution, such as malicious uploads to model-sharing repositories. This attack does not require access to any training, validation, or testing data. Operating under a white-box assumption (which is a valid assumption in contexts involving downloadable and modifiable models), InfrectroRL constructs a backdoor trajectory within the model's neurons and weights. This enables targeted behaviour to be executed upon trigger detection, without any further fine-tuning or retraining. Carefully tuned hyperparameters and pruning mechanisms ensure that the model maintains high performance on clean inputs. This contribution forms our sixth key result, **C6**.

Both attacks were evaluated across four Atari environments, which serve as the standard testbed for DRL security research. Our results show that the proposed attacks match or exceed the performance of existing state-of-the-art backdoor attacks, despite operating under significantly reduced adversarial privileges.

To further validate the strength of InfrectroRL, we evaluated its resilience against two state-of-the-art DRL backdoor defences: **BIRD** [36] and **SHINE** [248]. In both cases, the attack was able to evade detection and remained effective throughout the restoration and retraining phases of the defence pipeline.

In summary, this chapter addressed and resolved our second and fourth research questions, **RQ2** and **RQ4**. Specifically, we examined the credibility of existing backdoor defences under more realistic adversarial constraints and proposed two novel attacks that target previously unexplored stages of the DRL supply chain. These contributions significantly extend the

scope of the DRL backdoor literature and set the stage for future research into supply chain security in autonomous systems.

Future research in this area could focus on developing significantly more robust defence and sanitisation mechanisms, particularly by leveraging neuron activation analysis techniques explored in the previous chapter. These methods could be systematically applied to assess the resilience of our proposed attacks, *InfrectroRL* and *TrojanentRL*, as well as other training-time attacks from the literature.

The expansion of our threat model and the introduction of novel attack surfaces offer a pathway toward identifying more comprehensive and previously unexamined vulnerabilities within the DRL supply chain. This could pave the way for a broader understanding of holistic threat vectors in agent-based AI systems.

In the context of ACND and autonomous vehicles, although we have not yet applied our backdoor attacks to domain-specific environments, we believe that the proposed methods are transferable to this setting with minor adaptations. In particular, the use of contextually designed triggers tailored to the domain could enable effective backdoor activation. For example, simulation environments for ACND such as [1] and [223] provide ideal testbeds for designing and evaluating such attacks, allowing the development of optimised, domain-specific triggers and targeted behaviours. A dedicated ablation study within these environments could verify the practical impact and transferability of our attacks in real-world cyber defence scenarios. Simultaneously, the development of future DRL backdoor defences tailored for these domains would facilitate rigorous testing of mitigation strategies under more realistic operational constraints.

Chapter 7

Hardware Fault Injection Backdoors: Inference-stage Rowhammer Backdoors on Deep Reinforcement Learning Agents

7.1 Introduction

The preceding two chapters have focused on exposing vulnerabilities across the DRL supply chain through the lens of backdoor attacks. These chapters contributed to the field by introducing novel attack strategies and potential countermeasures, thus significantly advancing the state-of-the-art in DRL security. Specifically, the previous chapter examined backdoor attacks in both the pre- and post-training phases, demonstrating how adversarial interference at different stages of the DRL supply chain can compromise agent performance without detection.

Despite these advances, a significant portion of the DRL supply chain remains underexplored from a security standpoint. Notably, there has been little to no attention given to backdoor attacks that occur exclusively during the **inference phase**. This gap in the literature is striking, considering that inference represents a critical operational stage where the agent is deployed in real-time environments and is expected to behave reliably and safely. A lack of research in this area will inherently lead to the creation of defences that will not be able to sanitise the such threats.

We posit that inference-time attacks constitute the most realistic and low-privilege adversarial scenario within the DRL supply chain. Such attacks exploit the agent's deployment phase, where minimal access to the training code is required. This includes the redundancy

of utilising training-time inputs or exploiting loosely monitored observation pipelines to activate malicious behaviour.

A successful inference-time backdoor attack, therefore, would require the least adversarial control on the training pipeline while potentially yielding considerable damage, particularly in real-world or safety-critical applications such as robotics, autonomous vehicles, or ACND systems. Such an attack model not only represents a plausible threat in practical deployments but also highlights a pressing need for security mechanisms that operate at runtime, rather than being confined to pre-deployment checks or retraining-based sanitisation.

Consequently, the development and study of inference-time DRL backdoor attacks would represent a pivotal advancement in the literature. It would broaden the understanding of adversarial capabilities in DRL backdoor attack research, introduce more nuanced threat models, and compel the development of lightweight, online defence mechanisms capable of safeguarding agents during live operation. In this light, our exploration into inference-time vulnerabilities aims to establish a new frontier in DRL security which operates under practicality, minimal assumptions, and real-world applicability.

Unlike the current state-of-the-art in DRL backdoors, several recent works in the broader AI literature have demonstrated practical methods for manipulating the learnable parameters of deep neural networks directly in main memory. These methods focus on injecting faults by exploiting hardware-level vulnerabilities, most notably through the use of the well-documented *Rowhammer* attack [144, 100, 189]. Rowhammer is a technique that induces bit-flips in DRAM cells by repeatedly accessing adjacent memory rows, ultimately leading to subtle but effective changes in memory contents. When applied to neural networks, even small perturbations to parameter values introduced through Rowhammer can result in severe degradation or targeted misbehaviour of the model.

Building on this foundation, researchers have begun to apply Rowhammer-based techniques in the context of backdoor attacks. For example, [190, 35] extended the conceptual understanding of memory fault injection to craft backdoors in large-scale neural networks, showing that minimal and localised memory corruption can induce persistent and stealthy adversarial behaviours. More recently, [5] demonstrated that even complex transformer-based architectures can be successfully compromised using refined Rowhammer strategies, highlighting the growing feasibility of memory-level attacks against modern AI systems.

These findings illustrate that backdoor vulnerabilities are not limited to software-level manipulation during training or model deployment. Instead, the threat surface extends to the hardware runtime environment, offering adversaries new opportunities to tamper with models during inference, without modifying the model architecture, dataset, or training pipeline. Such attacks are particularly relevant to DRL systems deployed in untrusted or

loosely monitored environments, and motivate the exploration of inference-time backdoors with minimal adversarial access.

Building on the foundation of memory-based fault injection attacks, we propose **InfRLhammer**, a backdoor insertion algorithm designed to simulate inference-time weight corruption in DRL models. InfRLhammer is a *simulated* Rowhammer-style attack that operates during inference, exploiting low-level vulnerabilities in DRAM to introduce adversarial behaviour in a trained agent without modifying its training pipeline or accessing any data or gradients.

Rather than directly altering the stored model weights, InfRLhammer targets the memory space allocated to the DRL model by inducing carefully crafted bit flips in the associated page tables. These bit-level perturbations are triggered upon the appearance of a specific, optimised backdoor trigger pattern, allowing the attacker to corrupt the agent's policy execution in a stealthy and controlled manner.

By operating entirely during inference and without access to the training data or model optimisation process, InfRLhammer represents one of the lowest-privilege threat vectors explored in DRL backdoor literature to date, and further broadens the understanding of realistic attack surfaces in deployed DRL systems. Overall, this chapter allowed us to explore **RQ5**, which in turn, helped us make the seventh contribution, **C7** of our thesis.

To verify the effectiveness of our proposed attack, we evaluate InfRLhammer across four benchmark Atari environments. Our results demonstrate that the attack outperforms established the baseline, TrojDRL, in the DRL backdoor literature, despite requiring *no* access to the training or post-training pipeline. This highlights InfRLhammer's practicality and stealth under realistic deployment constraints.

To further assess its resilience, we save the backdoored models and evaluate them against two state-of-the-art DRL backdoor defences. Our findings show that InfRLhammer successfully evades both detection and sanitisation mechanisms, underscoring the limitations of current defence strategies and the need for more robust inference-time protections in DRL systems.

For this chapter, our contributions include:

1. **Expansion of the Threat Model:** We extend the threat model introduced in the previous chapter to encompass the possibility of DRL backdoor attacks occurring exclusively during *inference time*, thereby introducing a lower-privilege yet highly realistic adversarial scenario.

2. **First Simulated Hardware Fault Injection Backdoor Attack in DRL Literature:** Leveraging contemporary fault injection attack methodologies from the AI security

domain, we present the first simulated hardware-level backdoor attack in DRL. Our proposed method is benchmarked against existing baselines and shown to outperform them across key evaluation metrics.

3. **Demonstration of *InfRLhammer*'s Effectiveness Against State-of-the-Art Defences:** We evaluate *InfRLhammer* against two state-of-the-art DRL backdoor defences, BIRD [36] and SHINE [248], and show that the attack successfully evades both detection and sanitisation mechanisms in each defence pipeline.

7.2 Related Works

This section addresses the origins of InfRLhammer and discusses potential defence strategies that could prevent similar backdoors from fully materialising in DRL.

7.2.1 Origins of InfRLhammer

The conceptual foundation of InfRLhammer is inspired by the Rowhammer attack simulation scenario proposed by [5], which perturbs DRAM page tables through simulated Rowhammer faults. As previously discussed, this class of backdoor attack was initially introduced by [190] and [35], where supervised learning models were compromised via targeted bit-flips in specific memory bits, rather than through manipulation of entire page tables. In contrast to [5], our contribution extends this methodology to the domain of DRL, which typically comprises significantly fewer learnable parameters. Furthermore, we apply this approach within sequential decision-making environments, enabling the evaluation of backdoor persistence over extended temporal horizons.

7.2.2 Exploring Future Defences

As noted in the previous chapter, all current defences [248, 36, 20] against single-agent DRL backdoors are conceptual extensions of the Neural Cleanse framework [234]. While these defence mechanisms have demonstrated effectiveness against training-time attacks such as TrojDRL, they fail to mitigate the InfRLhammer attack. This limitation stems from their underlying design, which assumes access to and manipulation of training-time artefacts. As a result, these defences are inherently incapable of addressing inference-time backdoors, such as InfRLhammer, which operate outside the assumptions embedded in existing sanitisation techniques.

Existing defences against Rowhammer attacks [28, 174] have demonstrated that leveraging suspect thread monitoring and neuron activation analysis, an approach also proposed in our fourth chapter [231], as well as in prior works [33, 136, 252, 258], can be effective in mitigating such attacks within supervised learning settings. We posit that these techniques could be adapted and evaluated within the domain of DRL. However, due to the temporal dynamics inherent in DRL episodes, augmenting these defences with causal inference methods may yield more robust protection, particularly by identifying and neutralising time-dependent triggers and anomalies in agent behaviour.

7.3 Threat Model

As discussed previously, the threat model proposed in the preceding chapter did not consider attacks that occur during the *inference phase*—the stage at which a trained DRL model is actively deployed and engaged in real-time decision-making. This omission leaves a critical gap in the DRL security literature, particularly concerning adversarial backdoor threats that can be executed solely during inference, without requiring access to the training data, the training process, or any retraining of the model.

In this chapter, we address this gap by introducing a new threat model that focuses exclusively on the inference phase. We assume that the end-user has obtained a pretrained DRL model from an external entity, as outlined in the previous chapter, and has deployed it for decision-making tasks in their target environment. The deployed model runs on a computing system equipped with main memory (DRAM), which stores the model's learned parameters, including weights and biases. Within this memory system lies a critical component known as the *page table*, responsible for managing virtual-to-physical address mappings. Each entry in the page table corresponds to a page frame number, which in turn identifies a specific location in physical memory where a block of model weights is stored.

The end-user's computational infrastructure can vary significantly; from high-performance GPU clusters to resource-constrained edge devices, depending on the application domain. These systems interact with either private, open-source, or publicly available environment data during inference. The DRL model is executed within an application or software framework on this hardware, enabling it to perform high-level functions such as autonomous driving or ACND.

The end user's hardware may also incorporate integrity-preserving mechanisms such as Error-Correcting Codes (ECC), which are designed to detect and correct memory faults. These defences are intended to safeguard against unintended data corruption or adversarial breaches during model execution.

Attack Name	Threat Model	Attacker Knowledge	Code Access	Replay Buffer Access	Model Weights Access	Environment Access	Knowledge of $P(s'\|s,a)$	Modifies State	Modifies Action	Modifies Reward
SleeperNets	Training (Outer-Loop)	White-box	•	•		•	•	•		•
Q-Incept	Training (Outer-Loop)	White-box	•	•		•	•	•	•	•
TrojDRL	Training (Inner-Loop)	White-box		•		•		•	○	•
BadRL	Training (Inner-Loop)	White-box		•		•	•	•	○	•
BACKDOORL	Training (Full Control)	White-box	•	•	•	•	•		•	
TrojanentRL	Training (Project Code)	Grey-box	•	•		○		•		•
InfrectroRL	Post-Training	White-box			•	•		•	•	
InfRLhammer	Post-Training	White-box				•		•	•	

Table 7.1 Unified analytical taxonomy of DRL backdoor attacks combining adversarial knowledge assumptions, supply-chain access, and Markov Decision Process manipulation. The table extends prior work by incorporating one additional novel attack vector proposed in this thesis, namely **InfRLhammer**. The symbol • denotes capabilities required by all variants of an attack, while ○ indicates multiple strategies, some of which require the corresponding capability.

However, due to inherent limitations in commercial DRAM architectures, these chips remain susceptible to indirect data corruption. Specifically, repeatedly accessing a specific row in DRAM can induce electrical disturbances that result in bit-flips in adjacent memory rows; a vulnerability commonly exploited through Rowhammer, a hardware fault injection attack. Despite its growing significance in the broader AI security literature, this hardware-level threat has not yet been explored within the context of DRL security research.

7.4 Adversarial Capability

7.4.1 Rowhammer

As discussed in the previous section, the adversary leverages a memory fault injection technique known as *Rowhammer*, which exploits a well-documented vulnerability in commercially available DRAM chips. The underlying mechanism involves repeatedly accessing (or "hammering") a specific row within the DRAM to induce electrical disturbances and voltage fluctuations. These fluctuations, over time, lead to charge leakage in physically adjacent memory rows.

This leakage may ultimately result in unintended bit-flips—where binary values stored in neighbouring cells are erroneously altered. In the context of DRL, such controlled bit/page-level corruption can be strategically applied to specific memory regions where model parameters (e.g., weights or biases) are stored. By targeting sensitive weight blocks, the adversary is able to inject backdoor functionality into the deployed DRL policy.

Importantly, the Rowhammer technique operates entirely at inference-time and requires no access to the training process or data. This enables the adversary to introduce backdoors with minimal privileges, thereby achieving a high degree of stealth. Consequently, Rowhammer-based attacks present a potent and realistic threat model, particularly for real-world DRL deployments operating in untrusted or exposed environments.

7.4.2 Attack Model

The primary capability of the adversary in our threat model is the insertion of backdoors into the learnable parameters of a DRL model *after* it has been trained and deployed for inference. This approach eliminates the need for access to the victim's training pipeline or poisoning of the training data. The ultimate objective of the adversary is to compel the DRL agent to execute a targeted malicious action upon the appearance of a visual backdoor trigger in the input observation. Simultaneously, the model must preserve high performance on clean inputs to maintain stealth and avoid detection during regular operation.

Our attack model adopts a white-box setting, consistent with prior adversarial research targeting network parameters in deep learning systems [189, 190, 5]. However, unlike conventional white-box scenarios, our model assumes no access to the original training data. This is a realistic assumption, as previous works have demonstrated that training data can often be reconstructed or extracted through side-channel attacks or supply chain compromises.

The adversary is assumed to possess complete knowledge of the deployed model, including its architecture and parameter values. This distinguishes our approach from input-level adversarial attacks (e.g., adversarial examples), which typically assume access to or control over each input sample during inference. In contrast, our attack operates using a modest set of randomly sampled data—rather than synthesised adversarial inputs—which increases its practicality and plausibility in real-world scenarios.

Using the available data, the adversary can design effective backdoor triggers, often in the form of image patches or small square regions strategically embedded in the observation space. These triggers are optimised to activate specific neurons or weight blocks that drive the model towards the target action. To ensure attack reliability, the adversary leverages algorithmic techniques such as Rowhammer to identify memory locations or network parameters whose corruption (e.g., via simulated bit-flips) yields the highest impact on attack success. This attack is conducted solely during the inference phase of the pipeline.

Furthermore, the adversary can tune several hyperparameters, such as the number of corrupted addresses, the percentage of the image occupied by the trigger, and the spatial position of the trigger within the input. These configurations allow the adversary to balance

Clean Data Accuracy (CDA), Attack Success Rate (ASR), and Attack Effectiveness Rate (AER), depending on the desired threat scenario. They may also experiment with trigger transparency or brightness to reduce visual salience, albeit potentially at the cost of reduced attack potency.

7.4.3 Real-world Case Studies and Attack Realism

In the context of *InfRLhammer*, we acknowledge the presence of modern memory protection mechanisms such as Error-Correcting Codes (ECC) and Memory Protection Units (MPUs). However, prior research has demonstrated that these defences can be circumvented, particularly when adversaries exploit microarchitectural vulnerabilities. For instance, several studies [189, 190, 35, 5] have shown that bit-flip and Rowhammer-style attacks remain feasible in practice, even in the presence of such hardware-level protections.

Specifically, [44] highlight that Rowhammer techniques continue to be effective on systems equipped with modern memory protection features. Furthermore, recent works such as [109, 241] provide evidence of successful Rowhammer-style exploitation on contemporary hardware platforms, including AMD Zen 2/3 and DDR5-based systems manufactured as recently as 2020. These findings reinforce the practicality of memory corruption-based attack vectors, even on up-to-date systems.

The core novelty of *InfRLhammer* lies in its introduction of an *inference-time* backdoor threat model for DRL systems, an area that has remained unexplored in existing literature. Unlike conventional training-time attacks, inference-time backdoors bypass the need for access to training data, training code, or retraining pipelines. Moreover, as also noted in [5], such threats often evade retraining-based or fine-tuning-based sanitisation techniques, which are commonly employed in DRL defences.

Importantly, implementing continuous runtime surveillance to detect these attacks would impose significant computational overheads and is generally infeasible for large-scale or real-time DRL deployments. This limitation further underscores the significance and practical relevance of our proposed attack framework.

7.5 InfRLhammer

Existing DRL backdoor attacks vary in their assumed levels of access to the training pipeline; however, none have thus far demonstrated a feasible backdoor attack executed solely during the inference stage. To address this gap, we propose *InfRLhammer*, a novel backdoor injection technique that leverages Rowhammer-induced bit flips in DRAM to compromise

policy networks at inference time. Unlike traditional training-time attacks, *InfRLhammer* operates post-deployment and requires no access to training data or retraining routines. While it assumes white-box access to the model's architecture and parameters, it does not modify the model weights directly. Instead, the attack exploits DRAM vulnerabilities to *replace* selected *weight blocks* via controlled bit-flip patterns, thereby significantly reducing the privilege requirements of the adversary.

As previously discussed, Rowhammer-based fault injection techniques have been successfully demonstrated in supervised learning settings [100, 190, 5]. *InfRLhammer* extends these findings to DRL by identifying vulnerable weight blocks using a gradient-based analysis and selecting an optimal *replacement* from among existing weights based on structural similarity. These replacements are then executed through DRAM address manipulation upon the appearance of a backdoor trigger in the input observation.

While the attack requires access to a limited number of clean samples for evaluation purposes, it does not depend on gradient access during the execution phase. Instead, it relies solely on hardware-level vulnerabilities to compromise the DRL system, underscoring the risk posed by low-cost, inference-time attacks even under restricted adversarial capabilities. This attack corresponds to the third attack vector outlined in Section 6.4, thereby expanding the DRL threat landscape to encompass inference-phase vulnerabilities.

7.5.1 Problem Formulation

We implemented *InfRLhammer* on a Proximal Policy Optimisation (PPO) [203] agent with an actor–critic architecture, trained using the `RLZoo` framework [57]. Let π^* denote the trained target policy network, whose parameters are stored in memory as contiguous weight blocks. The memory layout of the network is abstracted as a set of virtual addresses, $\mathscr{A}_{\text{addr}} = \{\text{addr}_1, \ldots, \text{addr}_N\}$, where each 16-bit address addr_i maps to a physical memory location containing a weight block w_i. Each weight block w_i comprises a fixed subset of the network's parameters (e.g., 128 weights), all of which are critical for policy inference.

The complete collection of these blocks, $\mathscr{W} = \{w_1, \ldots, w_N\}$, defines the behavioural dynamics of the policy network controlling the agent's performance across the Atari environments. During attack execution, *InfRLhammer* employs Rowhammer-induced bit flips [116] to selectively replace targeted weight blocks in $\mathscr{W}$ at runtime. This allows the adversary to perturb the policy's decision-making without retraining or reinitialising the model, thereby injecting the backdoor behaviour stealthily during inference.

7.5.2 Adversarial Objective

The attack exploits page frame number bit flips in the memory address addr_i, redirecting the system to reference a replacement weight block w_j stored at addr_j instead of the original block w_i. This memory-level redirection manipulates the behaviour of the policy network by indirectly altering action probabilities and Q-value estimations, without modifying the weight values themselves.

This method introduces two critical constraints that must be addressed to ensure the attack's practicality and stealth:

1. **Attack Efficiency:** The total number of bit flips and Rowhammer iterations must be minimised to preserve operational stealth and allow the attack to remain feasible during real-time inference. Excessive perturbations increase the likelihood of detection and system instability.

2. **Memory Integrity:** The adversary is constrained to reusing existing weight blocks within the network. Arbitrary parameter injection is not permitted; thus, post-attack address mappings $\mathscr{A}'_{\text{addr}}$ must form a subset of the original address space $\mathscr{A}_{\text{addr}}$, and the resulting weight set $\mathscr{W}'$ must similarly be a subset of $\mathscr{W}$. Consequently, the attack operates by reassigning known parameters rather than introducing novel, potentially detectable alterations.

The first constraint dictates that selected weight blocks, when redirected, must produce the most significant behavioural deviations, while the second requires a systematic profiling of $\mathscr{W}$ to identify functionally equivalent substitutions that preserve stealth. These constraints ensure that the attack induces targeted behavioural changes without introducing detectable anomalies. To this end, we formalise the attack objective as a constrained optimisation problem:

$$\min_{\mathscr{W}'} \mathbb{E}_{s\sim\mathscr{S}}\big[L\big(\pi(s), y_t\big)\big] + \mathbb{E}_{\tilde{s}\sim\tilde{\mathscr{S}}}\big[L\big(F(\tilde{s}), y_t\big)\big], \tag{7.1}$$

$$\text{s.t.:} \quad D(\mathscr{A}'_{\text{addr}}, \mathscr{A}_{\text{addr}}) \leq \kappa'_t, \quad \mathscr{A}'_{\text{addr}} \subseteq \mathscr{A}_{\text{addr}}, \quad \mathscr{W}' \subseteq \mathscr{W}.$$

Here, s represents clean states, $\tilde{s}$ denotes triggered states containing adversarial patterns, and y_t is the attacker's desired target action. The loss function L measures the divergence from the target behaviour, and $D(\cdot,\cdot)$ denotes the number of address modifications between the original and perturbed address mappings, bounded by the attack budget κ'_t. The policy π is parameterised by the modified weights $\mathscr{W}'$, while $F(\cdot)$ denotes the inference under triggered conditions.

To implement this for PPO-based agents, we adopt the following approach:

- **Targeted Bit Flips:** We selectively redirect memory addresses associated with critical layers (e.g., final convolutional or fully connected layers) to replacement weight blocks, thereby manipulating the action probability distribution.
- **Optimised Overhead:** We prioritise the redirection of those weight blocks whose substitution maximises the deviation from benign policy behaviour while requiring the fewest bit flips and Rowhammer iterations.
- **Stealth Preservation:** We ensure that the modified network retains near-original performance on clean inputs, thereby avoiding anomalous behaviour during typical inference and maintaining low observability.

This strategy enables the adversary to identify sparse yet highly influential weight substitutions that degrade policy reliability only under triggered conditions, while conforming to hardware-level constraints. The result is a stealthy, inference-time backdoor that operates without any retraining or explicit access to training data or gradients.

7.5.3 Addressing Attack Execution

The attack unfolds in three sequential phases:

1. **Target Block Identification:** This phase involves locating the weight blocks within the policy network that are most sensitive to perturbations. A gradient-based sensitivity analysis is employed to rank blocks according to their influence on action selection, thereby identifying those whose redirection would yield the greatest behavioural deviation.
2. **Replacement Block Selection:** Given hardware-level constraints—namely, that replacements must be drawn from the existing set of weight blocks $\mathscr{W}$—the adversary selects substitute blocks that are both structurally compatible and behaviourally divergent. The objective is to induce maximal impact with minimal disruption to clean performance.
3. **Trigger Optimisation:** Finally, adversarial input patterns (triggers) are iteratively refined to reliably activate the backdoored behaviour. These triggers are designed to remain minimally perceptible or contextually plausible within the environment, thereby ensuring that the malicious policy remains dormant under normal conditions and is only activated in the presence of the trigger.

Target Block Identification

We quantify the vulnerability of each weight block using a *Trojan loss*, which balances the policy's fidelity on clean states with its deviation under triggered conditions:

$$L_{\text{trojan}} = L_{\text{CE}}(\pi(s), y) + L_{\text{CE}}(\pi(\tilde{s}), y_t), \tag{7.2}$$

where s and $\tilde{s}$ denote clean and triggered input states, respectively, and y, y_t represent the ground-truth and target labels. To identify influential parameters, we compute gradient magnitudes for each weight block w_i over n mini-batches:

$$g_i = \sum_{j=1}^{n} \nabla_{w_i} L_{\text{trojan}}(s_j, y_j, \tilde{s}_j, y_t), \quad \text{rank}(w_i) = \|g_i\|_2. \tag{7.3}$$

The blocks are then ranked by their ℓ_2-norm gradient scores. The top-k blocks constitute the target set $\mathscr{W}_t$, where k is determined by the attacker's modification budget.

Replacement Block Selection

Each target block $w_t \in \mathscr{W}_t$ must be replaced with an existing block $w_r \in \mathscr{W}$, subject to hardware-level constraints. To enforce feasibility, we introduce a cosine similarity constraint that ensures the replacement remains close in representation space:

$$L_{\text{constraint}} = \frac{1}{k} \sum_{\hat{w}_t} \left(1 - \max_{w_i \in \mathscr{W}} \frac{\hat{w}_t^\top w_i}{\|\hat{w}_t\|_2 \|w_i\|_2} \right), \tag{7.4}$$

where $\hat{w}_t$ denotes the idealised replacement vector for w_t, and k is the size of the target set. The objective function is then jointly optimised as:

$$L_{\text{total}} = L_{\text{trojan}} + \alpha L_{\text{constraint}},$$

where α is a trade-off parameter controlling similarity regularisation.

Upon optimisation, the final replacement block w_r for each w_t is selected as:

$$w_r = \arg \max_{w_i \in \mathscr{W} \setminus \{w_t\}} \hat{w}_t^\top w_i,$$

ensuring that only existing weight blocks are utilised, thereby maintaining memory integrity while maximising adversarial effect.

Trigger Optimisation

The trigger pattern δ in *InfRLhammer* is optimised to maximise policy deviation while respecting valid input constraints. To ensure that the pixel values of the trigger Δ remain within the permissible input range $[0,1]$, we introduce the following constraint loss:

$$L_{\text{trigger}} = \frac{1}{HW} \sum_{i,j} \left\| \Delta_{ij} - \Pi_{\text{valid}}(\Delta_{ij}) \right\|^2, \tag{7.5}$$

where $\Pi_{\text{valid}}(\cdot)$ denotes the projection operator that clamps each pixel value to the valid input interval.

The overall objective function of *InfRLhammer* then jointly optimises for policy manipulation, weight substitution feasibility, and input validity:

$$L_{\text{InfRLhammer}} = L_{\text{trojan}} + \alpha L_{\text{constraint}} + \beta L_{\text{trigger}}, \tag{7.6}$$

where α and β are regularisation coefficients that control the trade-off between replacement feasibility and trigger admissibility.

Final Attack Execution

Upon convergence of the optimisation process, the following components are finalised:

1. *Target blocks* $\mathscr{W}_t$: A set of weight blocks within the policy network identified as most vulnerable and thus selected for replacement;

2. *Replacement blocks* w_r: Optimal substitute weight blocks drawn from the existing parameter space, chosen to maximise backdoor efficacy while satisfying hardware-level constraints;

3. *Trigger pattern* δ: A pixel-level perturbation, defined by values Δ, engineered to reliably activate the backdoor when embedded in the input observation.

7.6 Experimental Setup

In this section, we first present the experimental setup used to evaluate the effectiveness of InfRLhammer. We then describe the evaluation metrics employed to assess their performance.

The experiments were conducted on a set of four well-established Atari environments: *Pong*, *Breakout*, *Q*bert* and *Space Invaders*. This selection is motivated by their widespread

adoption in the DRL security literature, thereby providing a robust and standardised benchmark for comparative analysis along with experimenting with defences. As mentioned in the previous chapter, the importance of these environments has been acknowledged by peer reviewers, such as Reviewer `q6gp` in the official review of the *SleeperNets* paper [193][1]. Research efforts grounded in these benchmarks contribute toward establishing the foundations for more secure and trustworthy DRL applications, particularly in high-stakes domains such as ACND.

7.6.1 Attack Setup

For the implementation of InfrectroRL, we utilise a pretrained PPO agent obtained from RLZoo [57] and inject a white-box backdoor based on an optimised trigger design. This algorithm is also developed using `PyTorch` and `Stable-Baselines3`.

As InfRLhammer is an inference-based backdoor attack, there is no need to train the agent from scratch. Instead, we perturb the agent's inference pipeline to ensure reliable trigger activation across varying environment settings. For all experiments, we use consistent hyperparameter settings in which the default trigger size is 2, target action is also 2, α is set to 0.1 and β is set to 5.0. The rest of the variables such as the number of address modifications (D), target blocks $\mathscr{W}_t$ and replacement blocks w_r are optimised accordingly. To ensure the validity of InfRLhammer's backdoor behaviour, we further conduct a comprehensive ablation study by varying each of these hyperparameters.

All experiments were conducted on an NVIDIA Tesla V100 GPU. The results, presented in Table 7.2, highlight the effectiveness of the InfrectroRL attack across all selected Atari environments.

Evading State-of-the-Art Defences

To assess the resilience of InfRLhammer against modern defence mechanisms, we evaluate its performance in the presence of two recently proposed state-of-the-art DRL backdoor defences: BIRD [36] and SHINE [248]. In accordance with the methodologies described in their original publications, we replicate the experimental protocols using an NVIDIA Tesla V100 GPU. The results, summarised in Table 7.3, indicate that *InfRLhammer* is able to effectively evade both defensive mechanisms. These findings highlight the stealth and adversarial potential of the proposed backdoor attack, underscoring the limitations of current defence strategies in mitigating post-training DRL threats.

[1]https://openreview.net/forum?id=NALkteEo9Q

7.6.2 Backdoor Attack Assessment Metrics

We evaluate attack performance using three standard metrics as used in the previous chapter. We define them again for the convenience of the reader:

- **Clean Data Accuracy (CDA):** Measures the model's performance on clean inputs, indicating its ability to maintain normal behaviour.
- **Attack Effectiveness Rate (AER):** Quantifies the drop in episodic return in the presence of the backdoor trigger, reflecting the attack's disruptive impact.
- **Attack Success Rate (ASR):** Represents the frequency with which the agent selects the attacker-specified action under triggered inputs.

These metrics together capture the stealth (via CDA) and impact (via AER and ASR) of the attack.

7.7 Experimental Results

Table 7.2 provides a comparative analysis of three DRL backdoor attacks: TrojDRL, BadRL, and the proposed InfRLhammer, evaluated across four standard Atari environments using the three key metrics defined in the previous chapter. A distinguishing feature of InfRLhammer is that it functions entirely during the inference phase by exploiting memory fault injection mechanisms, without requiring any access to the training pipeline. This characteristic substantially reduces the level of adversarial access required, positioning InfRLhammer as a significantly more realistic threat model compared to existing approaches in the literature.

In the Pong environment, while TrojDRL and BadRL both achieve near-perfect attack success rates (ASR), their corresponding attack execution rates (AER) on clean states reveal slightly reduced stealth in comparison to InfRLhammer. InfRLhammer attains an AER of 92.86 percent, indicating that it preserves clean policy behaviour more effectively, which is a crucial requirement for maintaining stealth in real-world deployments. Although its ASR is lower, at 68.70 percent, this nonetheless illustrates that inference-only attacks are capable of inducing non-trivial policy deviations without compromising performance on clean data. As discussed in the previous chapter, the ASR metric alone is not a reliable measure of attack impact in DRL contexts, where cumulative or delayed policy degradation can result in substantial long-term damage despite lower ASR values.

In the Breakout environment, InfRLhammer demonstrates marginally better overall performance compared to both baselines, although its ASR remains lower than that of

BadRL. In contrast, in the Qbert environment, BadRL achieves a 100 percent ASR, but this comes at the expense of reduced clean data accuracy (CDA) and AER (75.56 percent and 74.36 percent, respectively). InfRLhammer, on the other hand, records more balanced performance with a CDA of 79.49 percent and an AER of 77.47 percent. It is important to note, however, that Qbert has shown generally low attack effectiveness across all methods, likely due to the game's reward structure, where specific target actions do not consistently produce negative outcomes.

In the Space Invaders environment, InfRLhammer achieves an AER of 91.08 percent, which significantly outperforms both baselines. While its ASR remains lower than that of BadRL, the superior AER underscores its ability to maintain benign performance under clean conditions, thus enhancing stealth.

Overall, InfRLhammer demonstrates competitive performance across both CDA and AER when compared to TrojDRL and BadRL. In several instances, it outperforms both in terms of stealth and general reliability. The consistently higher AER values—apart from one exception in Space Invaders—highlight its robustness in preserving expected agent behaviour in the absence of triggers. Despite operating under significantly lower adversarial privilege assumptions, InfRLhammer achieves results on par with or exceeding those of training-time backdoor attacks, making it the most effective and realistic attack model currently available. When evaluated alongside InfrectroRL, this attack further enriches the set of practical supply chain threats in DRL.

Table 7.2 Comparison of back-door attacks in DRL. Higher CDA and ASR are better, while lower AER is better.

	Attack Name	**TrojDRL (Baseline)**			**BadRL**			**InfRLhammer**		
	Adversarial Breach Point	**Training-time Code**			**Training-time Code**			**Inference-time Memory Fault**		
	Metric	**CDA**	**AER**	**ASR**	**CDA**	**AER**	**ASR**	**CDA**	**AER**	**ASR**
Environment	Pong	98.66%	87.75%	98.85%	99.70%	100.00%	100.00%	91.02%	**92.86%**	68.70%
	Breakout	94.86%	3.13%	26.90%	95.44%	95.43%	89.92%	**99.20%**	**98.08%**	**41.62%**
	Qbert	78.04%	5.35%	32.87%	75.56%	74.36%	100.00%	**79.49%**	**77.47%**	**64.80%**
	Space Invaders	95.49%	32.63%	26.80%	78.72%	95.68%	99.84%	82.40%	**91.08%**	**76.90%**

7.7.1 Ablation Study

We conduct a series of ablation studies to examine the influence of hyperparameters on the performance of InfRLhammer. The attack is governed primarily by two hyperparameters: the trigger size and the target label. In our experiments, we systematically vary the trigger

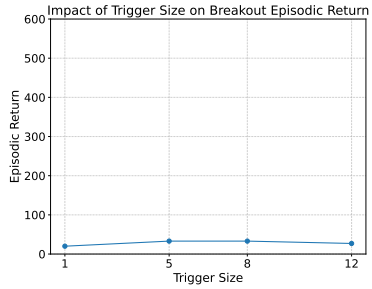
size across a range of pixel values within the input observation (e.g., 1×1, 5×5, 8×8, and 12×12) to evaluate its impact on the overall episodic return. Concurrently, we test different target labels to understand how the choice of action influences the agent's behaviour under triggered conditions. Through this analysis, we aim to identify configurations that maximise policy deviation while maintaining stealth. The findings offer key insights into the sensitivity and robustness of InfRLhammer across various deployment settings. While metrics such as ASR, AER, and CDA are informative, we primarily use the average episodic return as an initial indicator to determine which hyperparameter settings most significantly degrade policy performance.

Impact of Trigger Size:

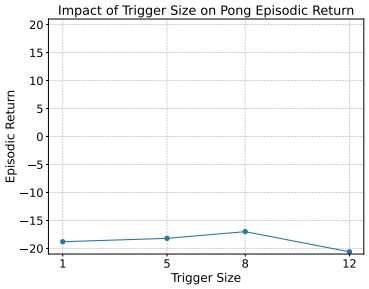
As illustrated in Figure 7.1, the backdoor effect of InfRLhammer appears largely resilient to variations in trigger size. In particular, the Space Invaders and Qbert environments exhibit negligible differences in overall episodic return across the tested trigger sizes. However, in the case of Pong, we observe a noticeable performance degradation of approximately 7% when the trigger size exceeds 8×8. While this drop may be attributed to several factors, we hypothesise that it results from visual obstruction, where the larger trigger partially overlaps with the agent's paddle, located on the left side of the game screen. This occlusion likely interferes with the agent's perception and decision-making process. Despite this exception, our findings suggest that trigger size has limited influence on the effectiveness of InfRLhammer across most environments.

Impact on Target Action:

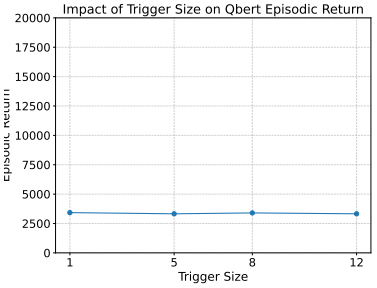
As shown in Figure 7.2, our experimental findings indicate that the choice of target action has minimal impact on the average episodic return. Across most environments tested, performance degradation remains consistent irrespective of the specific target action selected. This consistency may be attributed to the agent's behavioural tendencies when executing repetitive or contextually inappropriate actions—such as remaining stationary or drifting toward boundaries—which generally impede task progression and consequently reduce cumulative rewards. In the case of Space Invaders, we observed that InfRLhammer exhibited slightly reduced effectiveness when targeting action 0. We hypothesise that this action, which corresponds to "no operation", may have inadvertently allowed for occasional beneficial behaviours (e.g., firing actions triggered under low ASR conditions), leading to a partial neutralisation of incoming threats and mitigating the overall damage.



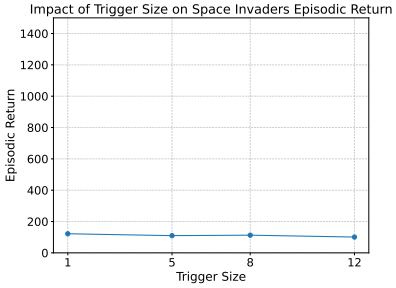
(a) Breakout



(b) Pong



(c) Qbert



(d) Space Invaders

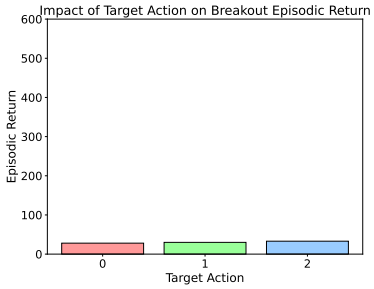
Fig. 7.1 Ablation study in InfRLhammer for varying the trigger size for all 4 games. We notice that InfRLhammer is resilient to backdoor attacks regardless of the change in trigger size.

These observations suggest that the success of InfRLhammer is largely agnostic to the specific target action, reinforcing the generalisability of the attack across environments with differing action semantics. Future work could expand this investigation by examining environments with more structured or hierarchical action spaces, where the impact of target action selection might be more pronounced. For instance, in the context of ACND and autonomous vehicles, where certain actions carry significantly greater strategic importance, a similar ablation study could offer valuable insight into how action semantics influence attack efficacy.

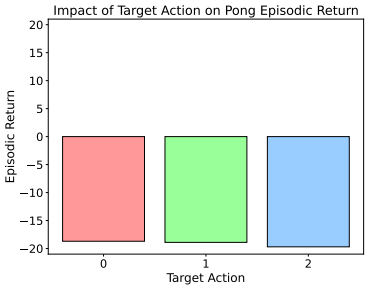
7.7.2 Applying DRL Backdoor Defences Against InfRLhammer

To evaluate the practicality of InfRLhammer, we assess its resilience against established DRL backdoor defences.

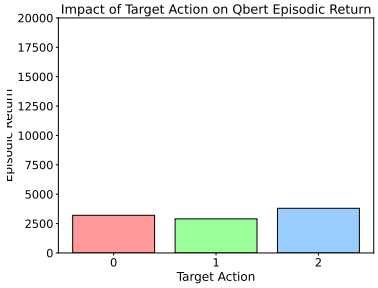
Table 7.3 presents a comparative evaluation of InfrectroRL and the baseline attack TrojDRL under two state-of-the-art backdoor defences, SHINE [248] and BIRD [36], across four Atari environments. The table reports raw episodic returns for each attack before and after the application of the defence mechanisms.



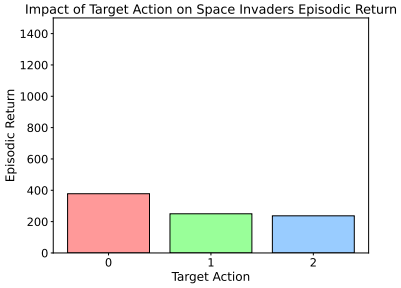
(a) Breakout



(b) Pong



(c) Qbert



(d) Space Invaders

Fig. 7.2 Ablation study in InfRLhammer for varying the target action for all 4 games. We notice slight deviations in episodic return depending on the attack, however, our attack still shows significantly reduced episodic returns.

Table 7.3 Comparison of raw episodic scores for TrojDRL and InfRLhammer after implementing SHINE [248] and BIRD [36] sanitisation defences across different Atari environments respectively.

Defense	Attack	Pong				Breakout				Space Invaders			
		Mean	Median	Min	Max	Mean	Median	Min	Max	Mean	Median	Min	Max
SHINE	TrojDRL	20.9	20.9	19.0	21.0	330.0	330.0	312.0	346.0	545.0	542.0	538.0	552.0
	InfRLhammer	**-20.7**	**-20.1**	**-21.0**	**-19.8**	**35.0**	**42.0**	**11.0**	**69.0**	**142.0**	**144.0**	**112.0**	**223.0**
BIRD	TrojDRL	20.0	20.0	19.2	20.7	275.0	271.0	224.0	316.0	548.0	554.0	510.0	586.0
	InfRLhammer	**-19.3**	**-19.4**	**-21.0**	**-18.6**	**18.9**	**22.5**	**5.0**	**47.0**	**262.5**	**243.5**	**112.0**	**322.0**

Under SHINE, TrojDRL demonstrates a substantial recovery of performance, with post-defence episodic returns of 20.876 in Pong, 330 in Breakout, 545 in Space Invaders, and 14,980 in Qbert, indicating the defence's ability to successfully neutralise the backdoor. In contrast, InfRLhammer maintains its degraded policy performance even after defence sanitisation, with scores remaining largely unchanged from pre-defence levels: -20.429 in Pong, 8 in Breakout, 40 in Space Invaders, and 270 in Qbert. This persistence of adversarial behaviour suggests that InfrectroRL is able to evade SHINE's sanitisation mechanism effectively, continuing to influence the policy at inference.

Results from BIRD show a similar trend. TrojDRL retains high post-defence returns of 19.95 in Pong and 326 in Breakout, whereas InfRLhammer achieves significantly reduced returns of -19.90 and 12 in the same environments, respectively. Space Invaders and Qbert also show significantly lower episodic returns under InfRLhammer, highlighting that InfRLhammer exhibits a higher degree of stealth and resilience against backdoor removal strategies compared to TrojDRL.

Collectively, the results highlight the strength of InfRLhammer not only in achieving adversarial policy degradation but also in evading two of the most recent and comprehensive DRL-specific defence frameworks. This underscores the critical need for developing more robust detection and mitigation strategies that are capable of addressing inference-time backdoor threats such as InfRLhammer.

7.8 Conclusion & Future Work

In this chapter, we further expanded the DRL backdoor threat landscape by extending the supply chain perspective established in earlier chapters. Prior to this work, all existing DRL backdoor attacks (such as those discussed in Chapters 5 and 6) have been confined to the training-time [117, 50, 193, 192], pre-training, or post-training (but pre-inference)

phases. Correspondingly, the DRL literature has only proposed three backdoor defence mechanisms [20, 36, 248], each demonstrating efficacy primarily against the TrojDRL attack [117].

This chapter introduces a novel class of backdoor threat that emerges exclusively during the inference phase and requires only minimal access, limited to a small batch of clean inference-time data. Building upon hardware fault injection simulation techniques from the broader AI security literature, we demonstrate that Rowhammer-style attacks can be effectively adapted to the domain of DRL. Our proposed backdoor attack, *InfRLhammer*, simulates a Rowhammer fault injection scenario within DRL and achieves competitive performance on four Atari benchmarks when compared with existing state-of-the-art DRL backdoor baselines. Furthermore, InfRLhammer successfully evades all currently available DRL-specific backdoor defences, **BIRD** [36] and **SHINE** [248] by operating entirely at runtime, leveraging controlled bit-flips in DRAM page tables to redirect the policy execution of the deployed model.

This implementation addresses our fifth research question, **RQ5**, by demonstrating that backdoor attacks can be inserted and activated solely during inference on models that were trained under benign conditions. In doing so, we present our final contribution to the thesis, denoted as **C7**.

This contribution highlights a previously unexplored yet highly realistic threat model within DRL backdoor research, characterised by significantly reduced adversarial privilege requirements. By demonstrating that effective backdoor manipulation can be achieved solely during inference, this work broadens the adversarial landscape in DRL. Overall, the proposed attack underscores the urgent need for defence mechanisms specifically designed to address inference-time vulnerabilities in reinforcement learning systems.

Future research in this domain may focus on the development of more robust and generalisable defence and sanitisation mechanisms, particularly through the incorporation of neuron activation analysis techniques previously explored in Chapter 4. These techniques, which have shown promise in identifying anomalous behavioural signatures within DRL policies, could be further refined and applied to detect subtle perturbations introduced by inference-time backdoors. Moreover, specific countermeasures against Rowhammer-style attacks, as proposed in prior studies [28, 174], offer a valuable direction for future exploration. When used in conjunction with neuron activation profiling, such techniques may significantly enhance the detection fidelity of runtime attacks like InfRLhammer.

While the experimental validation of InfRLhammer relied on visual benchmark environments such as Atari, the underlying attack mechanism exploits hardware-level bit-flips in model parameters and is therefore agnostic to the agent's input modality. We argue that this

methodology is readily transferable to non-visual, safety-critical domains, including ACND. In a cyber defence DRL application setting, the pixel-based triggers used in our experiments would correspond to logical feature combinations in network traffic, such as specific packet sequences or system state vectors. By introducing context-sensitive triggers that reflect the semantics of the target environment, inference-time attacks could be instantiated against cyber defence agents with minimal architectural modification. Simulation platforms such as CybORG and CyberBattleSim provide suitable experimental testbeds for evaluating such adversarial behaviours in non-visual state spaces.

A focused ablation study within these cyber defence environments would further validate the practical impact and transferability of inference-time hardware attacks under realistic operational conditions. Crucially, this line of work demonstrates that securing the software supply chain alone is insufficient if the underlying deployment hardware is not resilient to physical-layer perturbations. As a result, our findings motivate the systematic development of hardware-aware defences for DRL backdoors, including mechanisms such as memory integrity verification. This represents a critical step towards establishing robust, inference-aware security protocols for future autonomous defence systems.

Chapter 8

Conclusions, Limitations & Future Work

8.1 Introduction

The deployment of Deep Reinforcement Learning (DRL) has evolved from a theoretical research pursuit into a practical tool for automating intricate tasks across diverse sectors. The potential of DRL lies in its distinctive ability to combine strategic decision-making with the representational power of deep learning, enabling agents to optimise long-term rewards in dynamic and uncertain environments. From healthcare scheduling and magnetic control of plasmas to the operation of autonomous vehicles and robotics, DRL offers a path to solving complex challenges that were once computationally intractable. This progress has been further driven by the democratisation of AI, where practitioners increasingly rely on open-source frameworks and public repositories to source high-performing pretrained models. This is particularly vital for resource-constrained organisations such as start-ups that cannot afford the extensive computational costs of training from scratch.

However, this rapid transition towards deployment has outpaced the development of robust security frameworks, leaving the DRL supply chain exposed to severe and often under-examined threats. This thesis addresses this critical gap by investigating the integrity of the DRL pipeline in the face of adversarial backdoors. Unlike traditional adversarial examples that require per-input perturbations at inference, backdoors represent a uniquely insidious form of integrity violation that exploits the overparameterisation of deep neural networks to embed hidden behaviours which remain dormant until activated by a specific trigger.

This research first employs the application example of Autonomous Cyber Network Defence (ACND) to illustrate the benefits of DRL within high-stakes, safety-critical systems. It then identifies key bottlenecks that hinder deployment and highlights the most significant among them: adversarial backdoor attacks that can inflict severe harm on systems without the

end user's awareness. This thesis specifically expands the threat landscape of backdoor attacks beyond the unrealistic high-privilege and training-time assumptions common in existing literature. By introducing novel attacks such as TrojanentRL, InfrectroRL and InfRLhammer, it demonstrates that vulnerabilities exist in under-audited auxiliary components such as rollout buffers, post-training weight distributions and even inference-time hardware-level DRAM bit-flips. In parallel, the development of a lightweight neural activation-based detector provides a proactive means of identifying these concealed threats by monitoring the 'neural heartbeat' of the agent in real time.

A key limitation of this work is that while these methodologies establish the necessary scientific foundations for securing safety-critical applications such as ACND, they have been validated using standard benchmark environments such as Atari and MiniGrid. These represent domains where current research continues to progress before deployment in real-world applications. These simplified environments provided the essential rigour and reproducibility needed to establish a baseline before transitioning these security protocols to high-fidelity, domain-specific operational simulators. However, this chapter presents a brief roadmap outlining how the contributions of this thesis can be directly transferred to domains such as ACND, which feature distinct environmental configurations for deploying both backdoor attacks and corresponding defences.

8.2 Contributions

Overall, this thesis centres on defence mechanisms and the broader landscape of supply chain backdoor attacks in DRL systems. This area of focus was motivated by the growing awareness of supply chain vulnerabilities in artificial intelligence models. Despite this increasing concern, there remains a notable gap in the literature regarding the security of DRL, particularly in relation to backdoor threats. This thesis addresses that gap by systematically investigating both attack methodologies and potential defences within the DRL supply chain. In doing so, this work contributes a crucial step toward the reliable and secure deployment of DRL, with the aim to transfer towards the contributions to our chosen application domain of ACND (and potentially other domains too).

The first contribution, **C1**, of this thesis presents a systematic literature review of a promising DRL application domain: ACND. It first clarifies foundational terminology and identifies key subdomains, including Autonomous Blue and Red Teaming Agents and the Autonomous Cyber Operations Gym. It provides a comprehensive analysis of existing literature against a defined set of ACND-specific requirements. This analysis reveals critical research gaps and outstanding challenges of DRL that must be addressed to enable the effective deployment of

DRL within this application domain of real-world enterprise environments. This contribution directly addresses **RQ1**, which investigates the essential requirements for autonomous agents in ACND and highlights the specific gaps and challenges that DRL must overcome prior to its integration into ACND frameworks. From this analysis, ten overarching categories of limitations and open issues were identified, which represent essential benchmarks that DRL methodologies must satisfy before they can be confidently deployed in operational ACND contexts.

Following an extensive review of the key challenges identified in the ACND literature, this thesis focuses on a research area that is currently identified as one of the biggest bottlenecks towards DRL deployment: the robustness of DRL systems against adversarial attacks. This investigation directly addresses requirement (A.6.3) from Table 2.3, which emphasises the need to improve the resilience of DRL agents through the study of backdoor threats and corresponding defensive strategies. The second contribution of this thesis, **C2** (presented in Chapter 3), involves a critical evaluation of a state-of-the-art DRL backdoor defence proposed by [20]. A detailed examination of its theoretical assumptions revealed a key limitation: while we maintained compliance with two of the core assumptions, we challenged the third—namely, that all backdoor triggers reside outside the distribution of benign inputs. Leveraging this observation, we developed a novel backdoor attack in which the trigger is intentionally crafted to appear more visually in-distribution with the operational environment. This design undermines the assumption that distributional separation alone is sufficient for effective detection. At the time of implementation, the method proposed by [20] represented the only published defence in DRL backdoor literature. Therefore, successfully circumventing this defence highlighted a substantial and unaddressed vulnerability in the field. Through this contribution, we answer the initial part of **RQ2** by rigorously evaluating the credibility of DRL backdoor defences through a more realistic DRL backdoor trigger.

In response to the lack of comprehensive defence mechanisms within the DRL backdoor literature, as highlighted in Chapter 3, this thesis presents its third contribution, **C3**, detailed in Chapter 4. Here, we propose a lightweight, real-time, and novel backdoor detection system capable of identifying highly evasive DRL backdoor triggers within the operational environment. In the context of DRL systems, such a capability is particularly critical, as early detection of backdoors during inference is essential for effective mitigation before substantial damage can occur. As demonstrated in Chapter 4, the proposed detection system operates during inference and significantly outperforms the defence mechanism proposed by [20], not only in its ability to detect more realistic backdoor attacks but also in reducing detection latency. The proposed detector provides a practical foundation upon which future research efforts can develop more sophisticated DRL backdoor sanitisation strategies. This

contribution addresses **RQ3**, confirming the feasibility of implementing a more effective mitigation tool for DRL backdoor attacks.

Through our analysis of existing DRL backdoor literature, we observed a pronounced skew in the adversarial privilege levels assumed by current attack models. Specifically, all known attacks [117, 235, 50, 193, 192] require the highest level of adversarial access—namely, full control over the training pipeline. While such assumptions have enabled the development of effective attack mechanisms, they are largely unrealistic in real-world scenarios where such privileged access is seldom feasible. To address this gap, our fourth contribution, **C4**, presented in Chapter 5, involves the systematic identification of previously unexplored vulnerabilities across the DRL supply chain. In doing so, we proposed a formal definition of the DRL supply chain and constructed a comprehensive threat model that captures risks spanning multiple stages, including pre-training, training, post-training, and inference.

Leveraging this threat model, we make two additional contributions, **C5** and **C6**, by designing and implementing both component-level and post-training backdoor attacks. These attacks operate under significantly reduced adversarial privilege assumptions, marking a notable departure from the high-privilege models in existing literature. Through these contributions, we effectively address **RQ4** and **RQ4.1**, which explore the feasibility of broader supply chain-based attacks in DRL systems and the potential for adversarial strategies that do not rely on privileged access to the training pipeline.

While the feasibility of implementing supply chain backdoor threats *prior to deployment* has been demonstrated by chapter 5, it remains an open question whether such attacks can also be introduced exclusively during inference. Building upon the validated white-box threat model introduced by InfrectroRL, we demonstrate that perturbations to learnable parameters can serve as a basis for constructing backdoor attacks that operate solely at inference. Informed by this adversarial capability and supported by prior simulation studies [5, 189, 190], we present our final contribution, **C7**, through the development of InfRLhammer—the first inference-time backdoor attack designed specifically for DRL. This attack leverages hardware fault injection techniques to modify policy execution during inference without requiring access to the training pipeline. In doing so, we address our final research question, **RQ5**, which investigates whether existing methods from the broader AI security literature can be adapted to enable backdoor injection into benignly trained DRL models during inference.

8.3 Limitations

Chapters 3, 4 and 5 have each outlined limitations of the experimental results specific to those chapters. Here is a summary of some of the wider limitations of the research in this thesis.

8.3.1 Application to Autonomous Cyber Network Defence

While this thesis includes a comprehensive review of Autonomous Cyber Operations (ACO) gyms and their development, the experimental investigations were primarily conducted using benchmark environments such as Atari and Minigrid. One notable exception in the literature is the work by [192], which introduced a basic backdoor perturbation within the CybORG CAGE Challenge 2 environment. However, the proposed attack lacked realism and did not reflect the kinds of domain-specific backdoor scenarios likely to occur in practical Autonomous Cyber Network Defence (ACND) applications. Additionally, due to the inherent complexity of the CybORG environment, the authors were unable to evaluate their approach against state-of-the-art DRL backdoor defences. This limitation, in our view, diminishes the credibility and operational relevance of the attack. A similar concern was noted by Reviewer q6gp during the ICLR peer review process [1], where the authors were advised to conduct further experimentation on standard DRL benchmarks such as Atari to facilitate direct comparison with existing baselines and defence mechanisms. Consequently, our work focused on benchmark environments where both backdoor attacks and defences can be evaluated rigorously and reproducibly.

Nevertheless, we acknowledge the need for future research to extend backdoor methodologies and defences to more complex ACO gym environments. Platforms such as CybORG CAGE Challenge 2, 3, and 4 provide realistic and rich simulation environments well-suited for testing the operational viability of advanced DRL attacks and defences in enterprise-level network scenarios. Furthermore, the development of domain-specific defences, capable of generalising across enterprise network behaviours would be instrumental in crafting more targeted and effective mitigation strategies within the ACND context.

8.3.2 Creation of Robust Sanitisation Methods

In Chapter 3, we introduced a novel and computationally lightweight backdoor detection technique capable of operating in real-time. This method represents a significant step towards robust defences against DRL backdoors. Among the three DRL backdoor defence

[1] https://openreview.net/forum?id=NALkteEo9Q

mechanisms currently available in the literature, atleast one of which is incapable of detecting in-distribution backdoors, we believe our approach provides a strong foundation for future work. Its ability to function within the environment without significant overhead makes it well suited for integration into more generalised detection frameworks. This is particularly relevant for real-world deployment settings such as autonomous driving and ACND, where backdoors are expected to be visually in-distribution and thus highly evasive, rendering them difficult to detect both algorithmically and manually by human operators such as blue teamers.

However, this thesis does not implement a full sanitisation mechanism built atop the proposed detection method. In particular, we do not attempt to extract or isolate the backdoored policy from the benign components of the model once detection is achieved. Future work could explore the incorporation of more sophisticated behavioural separation techniques and targeted machine unlearning strategies to surgically remove malicious behaviours while preserving overall policy integrity.

Nonetheless, we contend that this thesis makes a substantial contribution to the field by expanding the DRL backdoor threat model and reducing the required adversarial capabilities compared to prior literature. These advancements establish a solid foundation upon which future research can build more comprehensive and holistic defence mechanisms, capable of mitigating DRL backdoor threats across the entire supply chain and during post-deployment operation. Overall, this will propel DRL backdoor defence research forwards as the current defences [20, 89, 36, 4, 248] only mitigate training-time backdoor attacks [117, 235].

8.4 Future Work

This thesis has aimed to explore and address several under-investigated areas in the domain of ACND, particularly in the context of emerging autonomous systems and DRL-based decision-making frameworks. The contributions presented throughout this work not only fill critical gaps left by prior research but also provide foundational insights intended to benefit both the academic research community and industrial practitioners. These contributions serve as a springboard for future investigations, encouraging the development of more robust, generalisable, and secure detection mechanisms in increasingly complex and adversarial environments.

For our contributions, we recognise future directions where researchers can explore in order to propel the integration of DRL for fully autonomous defence against cyber attacks.

8.4.1 Holistic Backdoor Detector

The backdoor detection method introduced in Chapter 3 demonstrated strong efficacy in identifying in-distribution backdoor attacks within the Minigrid environment. While these initial results are encouraging, future research should aim to establish the generalisability of this approach across a broader spectrum of DRL environments. Extending the evaluation to diverse settings would provide further validation of the detector's robustness and enhance its potential for real-world deployment. In addition, evaluating this strategy against a diverse set of backdoor attacks (in Chapter 5 and 6) will be imperative to holistically defend against DRL backdoors.

Crucially, while we contend that even the most evasive backdoors can be detected through analysis of the neural activation space, the temporal dynamics inherent in DRL present additional complexities. Effective backdoor detection must prioritise early identification of the trigger within an episode while maintaining a low false positive rate. As outlined in Chapter 3, causal inference-based methods hold considerable promise in this regard, as they may enable the identification of sequential patterns and dependencies that precede or co-occur with backdoor activation. Accordingly, the development of causal detection strategies tailored to DRL represents an important avenue for future research.

To transition this research towards application in ACND, initial experimentation should continue within controlled benchmark environments such as Minigrid and Atari. These results can then be extended to more complex simulation platforms, including CybORG CAGE Challenge 2, 3, and 4. Upon demonstrating consistent effectiveness in such simulated ACO gyms, the methodology could be further tested within emulated operational environments such as CSLE, facilitating the translation of this research towards practical, real-world DRL backdoor detection in cyber defence contexts.

8.4.2 Holistic Backdoor Sanitisation

While this thesis introduced a backdoor detection method for DRL agents, the challenge of effectively sanitising backdoored policies remains an open area for further investigation. Detection alone is insufficient without subsequent mechanisms to mitigate or remove the malicious behaviours introduced by the backdoor. Existing approaches such as those proposed by [36, 248] rely on retraining strategies that follow the detection and restoration phases. We believe that similar retraining-based strategies should be explored within the context of DRL.

Such a strategy would involve first isolating the malicious policy behaviour triggered by the backdoor, using outputs from the detection module. Once the adversarial action patterns are identified, targeted unlearning techniques can be applied to suppress or remove

the influence of these behaviours from the agent's policy. After sufficient reduction of the backdoored responses, retraining could then be undertaken to reinforce the agent's original benign policy, restoring its intended functionality. Future work should therefore focus on integrating detection and sanitisation in a unified framework, enabling DRL systems to recover from backdoor compromise in a controlled and robust manner.

Given the expanded threat model and the diverse forms of DRL backdoor attacks introduced in this thesis, future research should prioritise the development of holistic sanitisation mechanisms capable of addressing threats arising from code-level manipulations, model parameter corruption, and hardware-induced perturbations such as DRAM faults. Such unified defences will be essential to ensure the integrity and resilience of DRL systems across the full spectrum of potential attack vectors. As with detection methods, these sanitisation techniques should first be validated on a range of standard DRL benchmark environments—such as Minigrid and Atari—before being extended to more complex and domain-specific ACND environments. This staged approach will help establish both the practicality and generalisability of proposed defence mechanisms in real-world applications.

8.4.3 Rowhammer Attacks: Simulation to Reality

In Chapter 6, we extended the DRL backdoor threat model to include inference-time attacks. Specifically, we built upon prior work in supervised learning by [189, 190, 35, 5], adapting their hardware fault injection methodologies—most notably, Rowhammer—to the DRL context. By simulating such fault injections, we demonstrated that it is feasible to introduce backdoors in DRL policies during inference, without requiring access to the training pipeline or model parameters beforehand. While this contribution expands the scope of DRL security research to encompass inference-time vulnerabilities, it is important to acknowledge that, consistent with prior hardware fault injection literature, our implementation did not execute Rowhammer-induced bit-flips directly on physical DRAM. Instead, the attack was simulated to reflect the theoretical feasibility of such an approach. Future work may explore low-level implementations to validate the practical applicability of hardware-based inference-time backdoor attacks.

8.4.4 Manifesting All Technical Contributions To Autonomous Cyber Network Defence

While the experimental validation within this thesis utilized benchmark environments such as Atari and MiniGrid, the findings establish a scientific foundation for securing ACND systems. In a cybersecurity context, a DRL agent acts as an Autonomous Blue Team

defender, making sequential decisions to mitigate threats like lateral movement or data exfiltration. The "pixel-level triggers" explored in our attacks map directly to logical feature combinations in network traffic, such as rare TCP flag configurations or specific packet size sequences that an adversary might use to bypass defensive logic. Such a backdoor can also be implemented within CybORG CAGE Challenge 1, 2, 3 and 4 since their observation space is a feature vector that exposes the changes in the enterprise network activity. Therefore, implementing backdoors could either be as easy as manipulating an index within the feature vector (showing a unique activity pattern of components), or as complex as manipulating the internal processes through domain-specific attacks (like the example mentioned above) that are eventually elucidated within the compressed input observation feature vector. Overall, we believe that the injection of our backdoor attacks can manifest in several different areas of an ACND system. As a way to defend against such attacks, our defence based on neural activation monitoring can play a key role to decipher the origins of domain-specific backdoor attacks in real-time.

Now that this research has been established, several research directions can be taken that within the simulation and emulation (CybORG CAGE Challenge) level that were previously overlooked:

- **Red Teaming for Agent Integrity**: This research provides the blueprints for adversarial Red Teaming of DRL defenders. Cybersecurity researchers in ACND can now use these novel attack vectors (TrojanentRL, InfrectroRL, and InfRLhammer) to stress-test Blue Team agents through Red Team agents within CybORG CAGE Challenges, ensuring that the agents are resilient not just to external network attacks, but to internal supply-chain subversions.

- **Real-Time "Neural Heartbeat" Monitoring in ACND**: Before this research, malicious DRL activity was never assessed or noticed. We have proven that neural activation patterns can serve as a real-time indicator of an agent's integrity. This allows future researchers to first integrate "Neural Watchdog" systems to CybORG CAGE Challenges to assess agent's internal activations for anomalous shifts during live inference. If an agent encounters a trigger, the system can instantly flag the deviation and activate a human-in-the-loop failover, preventing the agent from making a catastrophic autonomous decision, such as disabling a firewall rule.

In addition to the research contributions, our contributions promote several practices within industry that are currently overlooked:

- **Proactive Supply Chain Auditing**:Prior to this work, DRL security focused almost exclusively on training-time scripts. We have demonstrated that vulnerabilities reside

in under-audited auxiliary components like the Rollout Buffer and in post-training weight distributions. Security practitioners can now move beyond basic code reviews to implement integrity monitoring for the entire DRL library stack, effectively creating a key practice to track the provenance of every framework dependency.

- **Hardware-Aware Cyber Resilience**: The introduction of InfRLhammer establishes that a cyber defender is vulnerable to hardware-level bit-flips during the deployment phase, even if trained under perfectly benign conditions. Organisations can now justify the requirement for ECC-protected memory or Rowhammer-resistant hardware layouts specifically for DRL agents performing high-assurance roles in critical infrastructure.

8.5 Concluding Remarks

The contributions of this thesis are designed to account for the needs of stakeholders in safety-critical domains such as ACND and autonomous vehicles in ways not previously addressed by existing research.

The thesis concludes with a forward-looking perspective, considering not only the technical research community working on our chosen application example of ACND but also the broader ecosystem of developers and practitioners focused on securing autonomous agent systems. Existing backdoor attacks in DRL literature present a narrow view of the adversarial landscape, thereby limiting the scope of future defence mechanisms to only a subset of plausible threats. By broadening the attack surface and offering practical defence tools, this thesis equips future DRL security researchers with the means to design and evaluate more comprehensive mitigation strategies. Ultimately, the methodologies and resources introduced here aim to support the development of trustworthy DRL-based agents for secure deployment in ACND and other high-stakes safety-critical domains.

References

[1] (2021). *CybORG: A Gym for the Development of Autonomous Cyber Agents*. arXiv.

[2] (2022). Cyber operations research gym. https://github.com/cage-challenge/CybORG. Created by Maxwell Standen, David Bowman, Son Hoang, Toby Richer, Martin Lucas, Richard Van Tassel, Phillip Vu, Mitchell Kiely, KC C., Natalie Konschnik, Joshua Collyer.

[3] (2023). Cyber agents for security testing and learning environments. https://sam.gov/opp/9c4593776a9b44e98b9bc734a3e16976/view#description. Created by Defense Advanced Research Projects Agency.

[4] Acharya, M., Zhou, W., Roy, A., Lin, X., Li, W., and Jha, S. (2023). Universal trojan signatures in reinforcement learning. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning-The Good, the Bad, and the Ugly*.

[5] Ahmed, S., Zhou, R., Angizi, S., and Rakin, A. S. (2024). Deep-troj: An inference stage trojan insertion algorithm through efficient weight replacement attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24810–24819.

[6] Alliance, Q. D. S. (2022). Artificial intelligence for decision making initiative (2022).

[7] Ammanabrolu, P. and Riedl, M. (2019). Playing text-adventure games with graph-based deep reinforcement learning. pages 3557–3565.

[8] Amodei, D. et al. (2016). Concrete problems in ai safety. *arXiv*.

[9] Andrew, A., Spillard, S., Collyer, J., and Dhir, N. (2022). Developing optimal causal cyber-defence agents via cyber security simulation. *arXiv preprint arXiv:2207.12355*.

[10] Applebaum, A., Dennler, C., Dwyer, P., Moskowitz, M., Nguyen, H., Nichols, N., Park, N., Rachwalski, P., Rau, F., Webster, A., et al. (2022). Bridging automated to autonomous cyber defense: Foundational analysis of tabular q-learning. In *Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security*, pages 149–159.

[11] Apruzzese, G., Andreolini, M., Marchetti, M., Venturi, A., and Colajanni, M. (2020). Deep reinforcement adversarial learning against botnet evasion attacks. *IEEE Transactions on Network and Service Management*, 17(4):1975–1987.

[12] Ashcraft, C. and Karra, K. (2021). Poisoning deep reinforcement learning agents with in-distribution triggers. *arXiv preprint arXiv:2106.07798*.

[13] Baah, G. K., Hobson, T., Okhravi, H., Roberts, S. C., Streilein, W. W., and Yuditskaya, S. (2015). A study of gaps in cyber defense automation.

[14] Barto, A. G. (2021). Reinforcement learning: An introduction. by richard's sutton. *SIAM Rev*, 6(2):423.

[15] Bates, E., Mavroudis, V., and Hicks, C. (2023). Reward shaping for happier autonomous cyber security agents. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 221–232.

[16] Behzadan, V. and Munir, A. (2017). Vulnerability of deep reinforcement learning to policy induction attacks. In *International conference on machine learning and data mining in pattern recognition*, pages 262–275. Springer.

[17] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

[18] Benjamin, D. P., Pal, P., Webber, F., Rubel, P., and Atigetchi, M. (2008). Using a cognitive architecture to automate cyberdefense reasoning. In *2008 Bio-inspired, Learning and Intelligent Systems for Security*, pages 58–63.

[19] Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. (2019). Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*.

[20] Bharti, S., Zhang, X., Singla, A., and Zhu, J. (2022). Provable defense against backdoor policies in reinforcement learning. *Advances in Neural Information Processing Systems*, 35:14704–14714.

[21] Bober-Irizar, M., Shumailov, I., Zhao, Y., Mullins, R., and Papernot, N. (2023). Architectural backdoors in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24595–24604.

[22] Booker, L. and Musman, S. (2022). A model-based, decision-theoretic perspective on automated cyber response.

[23] Brown, S., Brown, H., Russell, M., Henz, B., Edwards, M., Turner, F., and Bertoli, G. (2016). Validation of network simulation model and scalability tests using example malware. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 491–496.

[24] Brundage, M., Avin, S., Wang, J., Belfield, H., Krueger, G., Hadfield, G., Khlaaf, H., Yang, J., Toner, H., Fong, R., et al. (2020). Toward trustworthy ai development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*.

[25] Buettner, R., Sauter, D., Klopfer, J., Breitenbach, J., and Baumgartl, H. (2021). A review of recent advances in machine learning approaches for cyber defense. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3969–3974. IEEE.

[26] Burke, A. (2017 [Online]). Robust artificial intelligence for active cyber defence. Alan Turing Insitute.

[27] Cam, H. (2020). Cyber resilience using autonomous agents and reinforcement learning. In Pham, T., Solomon, L., and Rainey, K., editors, *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II*, volume 11413, pages 219 – 234. International Society for Optics and Photonics, SPIE.

[28] Canpolat, O., Yağlıkçı, A. G., Olgun, A., Yuksel, I. E., Tuğrul, Y. C., Kanellopoulos, K., Ergin, O., and Mutlu, O. (2024). Breakhammer: Enhancing rowhammer mitigations by carefully throttling suspect threads. In *2024 57th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 915–934. IEEE.

[29] Cao, B., Jia, J., Hu, C., Guo, W., Xiang, Z., Chen, J., Li, B., and Song, D. (2024). Data free backdoor attacks. *arXiv preprint arXiv:2412.06219*.

[30] Cao, Y., Chen, X., Yao, L., Wang, X., and Zhang, W. E. (2020). Adversarial attacks and detection on reinforcement learning-based interactive recommender systems. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1669–1672.

[31] Castro, P. S., Moitra, S., Gelada, C., Kumar, S., and Bellemare, M. G. (2018). Dopamine: A Research Framework for Deep Reinforcement Learning.

[32] Chadha, R., Bowen, T., Chiang, C.-Y. J., Gottlieb, Y. M., Poylisher, A., Sapello, A., Serban, C., Sugrim, S., Walther, G., Marvel, L. M., et al. (2016). Cybervan: A cyber security virtual assured network testbed. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 1125–1130. IEEE.

[33] Chai, S. and Chen, J. (2022). One-shot neural backdoor erasing via adversarial weight masking. *Advances in Neural Information Processing Systems*, 35:22285–22299.

[34] Chai, X., Wang, Y., Yan, C., Zhao, Y., Chen, W., and Wang, X. (2020). Dq-motag: Deep reinforcement learning-based moving target defense against ddos attacks. pages 375–379.

[35] Chen, H., Fu, C., Zhao, J., and Koushanfar, F. (2021a). Proflip: Targeted trojan attack with progressive bit flips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7718–7727.

[36] Chen, X., Guo, W., Tao, G., Zhang, X., and Song, D. (2023). Bird: generalizable backdoor detection and removal for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 36:40786–40798.

[37] Chen, X., Liu, C., Li, B., Lu, K., and Song, D. (2017). Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

[38] Chen, Y., Zheng, Z., and Gong, X. (2022). Marnet: Backdoor attacks against cooperative multi-agent reinforcement learning. *IEEE Transactions on Dependable and Secure Computing*.

[39] Chen, Y.-Y., Chen, C.-T., Sang, C.-Y., Yang, Y.-C., and Huang, S.-H. (2021b). Adversarial attacks against reinforcement learning-based portfolio management strategy. *IEEE Access*, 9:50667–50685.

[40] Choo, C. S., Chua, C. L., and Tay, S.-H. V. (2007). Automated red teaming: A proposed framework for military application. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, page 1936–1942, New York, NY, USA. Association for Computing Machinery.

[41] Chowdhary, A., Huang, D., Mahendran, J. S., Romo, D., Deng, Y., and Sabur, A. (2020). Autonomous security analysis and penetration testing. In *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pages 508–515. IEEE.

[42] Chowdhary, A., Huang, D., Sabur, A., Vadnere, N., Kang, M., and Montrose, B. (2021). Sdn-based moving target defense using multi-agent reinforcement learning. Autonomous Intelligent Cyber-defence Agents Conference.

[43] Clemente, A. V., Castejón, H. N., and Chandra, A. (2017). Efficient parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1705.04862*.

[44] Cojocar, L., Razavi, K., Giuffrida, C., and Bos, H. (2019). Exploiting correcting codes: On the effectiveness of ecc memory against rowhammer attacks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 55–71. IEEE.

[45] Colbert, E. J., Kott, A., and Knachel, L. P. (2020). The game-theoretic model and experimental investigation of cyber wargaming. *The Journal of Defense Modeling and Simulation*, 17(1):21–38.

[46] Collyer, J., Andrew, A., and Hodges, D. (2022). Acd-g: Enhancing autonomous cyber defense agent generalization through graph embedded network representation. International Conference on Machine Learning.

[47] Conejo, A. et al. (2010). *Decision Making Under Uncertainty in Electricity Markets*. Springer.

[48] Crumpler, W. and Lewis, J. A. (2022). *Cybersecurity Workforce Gap*. JSTOR.

[49] Cui, J., Han, Y., Ma, Y., Jiao, J., and Zhang, J. (2023). Badrl: Sparse targeted backdoor attack against reinforcement learning. *arXiv preprint arXiv:2312.12585*.

[50] Cui, J., Han, Y., Ma, Y., Jiao, J., and Zhang, J. (2024). Badrl: Sparse targeted backdoor attack against reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11687–11694.

[51] Dazeley, R., Vamplew, P., and Cruz, F. (2021). Explainable reinforcement learning for broad-xai: A conceptual framework and survey. *arXiv preprint arXiv:2108.09003*.

[52] Defence, N. (2021). Government of canada.

[53] Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., et al. (2022). Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419.

[54] Devitt, S. K. and Copeland, D. (2021). Australia's approach to ai governance in security and defence.

[55] Dhir, N., Hoeltgebaum, H., Adams, N., Briers, M., Burke, A., and Jones, P. (2021). Prospective artificial intelligence approaches for active cyber defence. *arXiv preprint arXiv:2104.09981*.

[56] Dias, L. S. and Ierapetritou, M. G. (2016). Integration of scheduling and control under uncertainties: Review and challenges. *Chemical Engineering Research and Design*, 116:98–113.

[57] Ding, Z., Yu, T., Huang, Y., Zhang, H., Mai, L., and Dong, H. (2020). Rlzoo: A comprehensive and adaptive reinforcement learning library. *arXiv preprint arXiv:2009.08644*.

[58] Dondo, M. and Nakhla, N. (2021). Towards a framework for autonomous defensive cyber operations in a network operations centre.

[59] Dulac-Arnold, G., Mankowitz, D., and Hester, T. (2019). Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*.

[60] Eghtesad, T., Vorobeychik, Y., and Laszka, A. (2020). Adversarial deep reinforcement learning based adaptive moving target defense. *Decision and Game Theory for Security: 11th International Conference*, pages 58–79.

[61] Emerson, H., Guy, M., and McConville, R. (2023). Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes. *Journal of Biomedical Informatics*, 142:104376.

[62] Eskridge, T. C., Carvalho, M. M., Stoner, E., Toggweiler, T., and Granados, A. (2015). Vine: A cyber emulation environment for mtd experimentation. MTD '15, page 43–47, New York, NY, USA. Association for Computing Machinery.

[63] Everett, M., Lütjens, B., and How, J. P. (2021). Certifiable robustness to adversarial state uncertainty in deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4184–4198.

[64] Fang, Z., Wang, J., Li, B., Wu, S., Zhou, Y., and Huang, H. (2019). Evading anti-malware engines with deep reinforcement learning. *IEEE Access*, 7:48867–48879.

[65] Foley, H., Fowl, L., Goldstein, T., and Taylor, G. (2022a). Execute order 66: targeted data poisoning for reinforcement learning. *arXiv preprint arXiv:2201.00762*.

[66] Foley, M., Hicks, C., Highnam, K., and Mavroudis, V. (2022b). Autonomous network defence using reinforcement learning. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22, page 1252–1254, New York, NY, USA. Association for Computing Machinery.

[67] Foley, M., Hicks, C., Highnam, K., and Mavroudis, V. (2022c). Autonomous network defence using reinforcement learning. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22, page 1252–1254, New York, NY, USA. Association for Computing Machinery.

[68] for Cyber Defense: The Accuracy-Robustness Tradeoff, M. A. W. (2021).

[69] Frankish, K. and Ramsey, W. (2014). *The Cambridge Handbook of Artificial Intelligence*. Cambridge University Press.

[70] Furfaro, A., Piccolo, A., Parise, A., Argento, L., and Saccà, D. (2018). A cloud-based platform for the emulation of complex cybersecurity scenarios. *Future Generation Computer Systems*, 89:791–803.

[71] Futoransky, A., Miranda, F., Orlicki, J., and Sarraute, C. (2009). Simulating cyber-attacks for fun and profit. page 4.

[72] Gangupantulu, R., Cody, T., Rahma, A., Redino, C., Clark, R., and Park, P. (2021). Crown jewels analysis using reinforcement learning with attack graphs. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6. IEEE.

[73] Gao, C. and Wang, Y. (2021). Reinforcement learning based self-adaptive moving target defense against ddos attacks. *Journal of Physics: Conference Series*, 1812:012039.

[74] Gao, Y., Zhang, G., and Xing, C. (2021). A multiphase dynamic deployment mechanism of virtualized honeypots based on intelligent attack path prediction. *Security and Communication Networks*, 2021.

[75] Garcia, J. and Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *JMLR*.

[76] Gasse, M., Grasset, D., Gaudron, G., and Oudeyer, P.-Y. (2021). Causal reinforcement learning using observational and interventional data. *arXiv preprint arXiv:2106.14421*.

[77] Ginsberg, A. S. and Offensend, F. L. (2007). An application of decision theory to a medical diagnosis-treatment problem. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):355–362.

[78] Glanois, C., Weng, P., Zimmer, M., Li, D., Yang, T., Hao, J., and Liu, W. (2021). A survey on interpretable reinforcement learning. *arXiv preprint arXiv:2112.13112*.

[79] Gleave, A., Dennis, M., Wild, C., Kant, N., Levine, S., and Russell, S. (2019). Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*.

[80] Gong, C., Yang, Z., Bai, Y., He, J., Shi, J., Li, K., Sinha, A., Xu, B., Hou, X., Lo, D., et al. (2023). Baffle: Backdoor attack in offline reinforcement learning. *arXiv preprint arXiv:2210.04688*.

[81] Gong, C., Yang, Z., Bai, Y., He, J., Shi, J., Li, K., Sinha, A., Xu, B., Hou, X., Lo, D., et al. (2024). Baffle: Hiding backdoors in offline reinforcement learning datasets. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 2086–2104. IEEE.

[82] Gore, R., Diallo, S., Padilla, J., and Ezell, B. (2018). Assessing cyber-incidents using machine learning. *International Journal of Information and Computer Security*, 10:341.

[83] Grouo, T. C. W. (2022). Ttcp cage challenge 2. https://github.com/cage-challenge/cage-challenge-2.

[84] Group, T. C. W. (2021). Cage challenge 1. arXiv.

[85] Group, T. C. W. (2022). Ttcp cage challenge 3. https://github.com/cage-challenge/cage-challenge-3.

[86] Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

[87] Gu, T., Dolan-Gavitt, B., and Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

[88] Gunn, S., Jang, D., Paradise, O., Spangher, L., and Spanos, C. J. (2022). Adversarial poisoning attacks on reinforcement learning-driven energy pricing. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 262–265.

[89] Guo, J., Li, A., Wang, L., and Liu, C. (2023). Policycleanse: Backdoor detection and mitigation for competitive reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4699–4708.

[90] Hammar, K. and Stadler, R. (2020). Finding effective security strategies through reinforcement learning and self-play. In *2020 16th International Conference on Network and Service Management (CNSM)*, pages 1–9. IEEE.

[91] Hammar, K. and Stadler, R. (2021). Learning intrusion prevention policies through optimal stopping. In *2021 17th International Conference on Network and Service Management (CNSM)*, pages 509–517. IEEE.

[92] Hannay, J. (2022a). Cyborg Cage 2 Solution. https://github.com/john-cardiff/-cyborg-cage-2.

[93] Hannay, J. (2022b). Cyborg Cage 2 Solution. https://github.com/john-cardiff/-cyborg-cage-2.

[94] Hassan, S. S., Park, Y. M., Tun, Y. K., Saad, W., Han, Z., and Hong, C. S. (2024). Spaceris: Leo satellite coverage maximization in 6g sub-thz networks by mappo drl and whale optimization. *IEEE Journal on Selected Areas in Communications*, 42(5):1262–1278.

[95] Hengst, B. (2010). *Hierarchical Reinforcement Learning*, pages 495–502. Springer US, Boston, MA.

[96] Hicks, C., Mavroudis, V., Foley, M., Davies, T., Highnam, K., and Watson, T. (2023). Canaries and whistles: Resilient drone communication networks with (or without) deep reinforcement learning. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pages 91–101.

[97] Hofer, W., Edgar, T., Vrabie, D., and Nowak, K. (2019). Model-driven deception for control system environments. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7. IEEE.

[98] Hoffman, R. R., Mueller, S. T., Klein, G., and Litman, J. (2018). Metrics for explainable ai: Challenges and prospects.

[99] Hoffman, W. (2021). Making ai work for cyber defense.

[100] Hong, S., Frigo, P., Kaya, Y., Giuffrida, C., and Dumitraș, T. (2019). Terminal brain damage: Exposing the graceless degradation in deep neural networks under hardware fault attacks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 497–514.

[101] Huang, L. and Zhu, Q. (2019a). Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks. *SIGMETRICS Perform. Eval. Rev.*, 46(2):52–56.

[102] Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. (2017). Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*.

[103] Huang, Y., Huang, L., and Zhu, Q. (2021). Reinforcement learning for feedback-enabled cyber resilience.

[104] Huang, Y. and Zhu, Q. (2019b). Deceptive reinforcement learning under adversarial manipulations on cost signals. In *GameSec*.

[105] Hugging Face (2025). Hugging face – the ai community building the future. https://huggingface.co/. Accessed: 2025-05-05.

[106] Ilahi, I., Usama, M., Qadir, J., Janjua, M. U., Al-Fuqaha, A., Hoang, D. T., and Niyato, D. (2021). Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 3(2):90–109.

[107] Irshayyid, A., Chen, J., and Xiong, G. (2024). A review on reinforcement learning-based highway autonomous vehicle control. *Green Energy and Intelligent Transportation*, page 100156.

[108] Janisch, J., Pevnỳ, T., and Lisỳ, V. (2023). Nasimemu: Network attack simulator & emulator for training agents generalizing to novel scenarios. *arXiv preprint arXiv:2305.17246*.

[109] Jattke, P., Wipfli, M., Solt, F., Marazzi, M., Bölcskei, M., and Razavi, K. (2024). {ZenHammer}: Rowhammer attacks on {AMD} zen-based platforms. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1615–1633.

[110] JFrog Security Research (2024). Xz backdoor attack cve-2024-3094: All you need to know. *JFrog Blog*. Accessed: 2025-05-06.

[111] Jin, Y.-L., Ji, Z.-Y., Zeng, D., and Zhang, X.-P. (2022). Vwp: An efficient drl-based autonomous driving model. *IEEE Transactions on Multimedia*, 26:2096–2108.

[112] Johnson Kinyua, L. A. (2021). Ai/ml in security orchestration, automation and response: Future research directions. *Intelligent Automation & Soft Computing*, 28(2):527–545.

[113] Kaddour, J., Lynch, A., Liu, Q., Kusner, M. J., and Silva, R. (2022). Causal machine learning: A survey and open problems.

[114] Keele, S. et al. (2007). Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, ver. 2.3 ebse technical report. ebse.

[115] Kim, M.-Y., Atakishiyev, S., Babiker, H. K. B., Farruque, N., Goebel, R., Zaïane, O. R., Motallebi, M.-H., Rabelo, J., Syed, T., Yao, H., and Chun, P. (2021). A multi-component framework for the analysis and design of explainable artificial intelligence. *Machine Learning and Knowledge Extraction*, 3(4):900–921.

[116] Kim, Y., Daly, R., Kim, J., Fallin, C., Lee, J. H., Lee, D., Wilkerson, C., Lai, K., and Mutlu, O. (2014). Flipping bits in memory without accessing them: An experimental study of dram disturbance errors. *ACM SIGARCH Computer Architecture News*, 42(3):361–372.

[117] Kiourti, P., Wardega, K., Jha, S., and Li, W. (2020). Trojdrl: evaluation of backdoor attacks on deep reinforcement learning. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE.

[118] Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE transactions on intelligent transportation systems*, 23(6):4909–4926.

[119] Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele Univ.*, 33.

[120] Ko, R. K. (2020). Cyber autonomy: automating the hacker–self-healing, self-adaptive, automatic cyber defense systems and their impact on industry, society, and national security. In *Emerging technologies and international security*, pages 173–191. Routledge.

[121] Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

[122] Kott, A., Théron, P., Drašar, M., Dushku, E., LeBlanc, B., Losiewicz, P., Guarino, A., Mancini, L., Panico, A., Pihelgas, M., et al. (2018). Autonomous intelligent cyber-defense agent (aica) reference architecture. release 2.0. *arXiv preprint arXiv:1803.10664*.

[123] Kujanpää, K., Victor, W., and Ilin, A. (2021). Automating privilege escalation with deep reinforcement learning. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pages 157–168.

[124] Landauer, M., Skopik, F., Frank, M., Hotwagner, W., Wurzenberger, M., and Rauber, A. (2022). Maintainable log datasets for evaluation of intrusion detection systems. *IEEE Transactions on Dependable and Secure Computing*.

[125] Landauer, M., Skopik, F., and Wurzenberger, M. (2023). Introducing a new alert data set for multi-step attack analysis. *arXiv preprint arXiv:2308.12627*.

[126] Landauer, M., Skopik, F., Wurzenberger, M., Hotwagner, W., and Rauber, A. (2020). Have it your way: Generating customized log datasets with a model-driven simulation testbed. *IEEE Transactions on Reliability*, 70(1):402–415.

[127] Langford, H., Shumailov, I., Zhao, Y., Mullins, R., and Papernot, N. (2024). Architectural neural backdoors from first principles. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 60–60. IEEE Computer Society.

[128] Lee, K., Cooper, A. F., and Grimmelmann, J. (2023). Talkin"bout ai generation: Copyright and the generative-ai supply chain. *arXiv preprint arXiv:2309.08133*.

[129] Lee, S. and Lee, Y. H. (2020). Improving emergency department efficiency by patient scheduling using deep reinforcement learning. In *Healthcare*, volume 8, page 77. MDPI.

[130] Lesort, T. (2020). Continual learning: Tackling catastrophic forgetting in deep neural networks with replay processes. *arXiv preprint arXiv:2007.00487*.

[131] Levine, S. et al. (2016). End-to-end training of deep visuomotor policies. *JMLR*.

[132] Li, H., Guo, Y., Sun, P., Wang, Y., and Huo, S. (2022a). An optimal defensive deception framework for the container-based cloud with deep reinforcement learning. *IET Information Security*, 16(3):178–192.

[133] Li, L., Fayad, R., and Taylor, A. (2021). Cygil: A cyber gym for training autonomous agents over emulated network systems. *arXiv preprint arXiv:2109.03331*.

[134] Li, P., Thomas, J., Wang, X., Khalil, A., Ahmad, A., Inacio, R., Kapoor, S., Parekh, A., Doufexi, A., Shojaeifard, A., et al. (2022b). Rlops: Development life-cycle of reinforcement learning aided open ran. *IEEE Access*, 10:113808–113826.

[135] Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J. E., Jordan, M. I., and Stoica, I. (2018). RLlib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)*.

[136] Liang, Y., Sun, Y., Zheng, R., and Huang, F. (2022). Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in neural information processing systems*, 35:22547–22561.

[137] Lillicrap, T. et al. (2016). Continuous control with deep reinforcement learning. *ICLR*.

[138] Lin, W., Liu, L., Wei, S., Li, J., and Xiong, H. (2024). Unveiling and mitigating backdoor vulnerabilities based on unlearning weight changes and backdoor activeness. *arXiv preprint arXiv:2405.20291*.

[139] Lin, Y.-C., Hong, Z.-W., Liao, Y.-H., Shih, M.-L., Liu, M.-Y., and Sun, M. (2017). Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*.

[140] Ling, C., Peng, K., Wang, S., Xu, X., and Leung, V. C. (2024). A multi-agent drl-based computation offloading and resource allocation method with attention mechanism in mec-enabled iiot. *IEEE Transactions on Services Computing*.

[141] Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

[142] Littman, M. (2009). Algorithms for sequential decision making.

[143] Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., and Zhang, X. (2018). Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc.

[144] Liu, Y., Wei, L., Luo, B., and Xu, Q. (2017). Fault injection attack on deep neural network. In *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 131–138. IEEE.

[145] Lütjens, B., Everett, M., and How, J. P. (2020). Certified adversarial robustness for deep reinforcement learning. In *conference on Robot Learning*, pages 1328–1337. PMLR.

[146] Lyu, D., Yang, F., Liu, B., and Gustafson, S. (2019). Sdrl: Interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:2970–2977.

[147] Madumal, P., Miller, T., Sonenberg, L., and Vetere, F. (2019). Explainable reinforcement learning through a causal lens.

[148] Maeda, R. and Mimura, M. (2021). Automating post-exploitation with deep reinforcement learning. *Computers & Security*, 100:102108.

[149] Mahaini, M. I., Li, S., and Sağlam, R. B. (2019). Building taxonomies based on human-machine teaming: Cyber security as an example. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–9.

[150] Mahendran, A. and Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120:233–255.

[151] Malialis, K. and Kudenko, D. (2013). Large-scale ddos response using cooperative reinforcement learning.

[152] Mengara, O., Avila, A., and Falk, T. H. (2024). Backdoor attacks to deep neural networks: A survey of the literature, challenges, and future research directions. *IEEE Access*.

[153] Miehling, E., Rasouli, M., and Teneketzis, D. (2015). Optimal defense policies for partially observable spreading processes on bayesian attack graphs. In *Proceedings of the second ACM workshop on moving target defense*, pages 67–76.

[154] Milani, S., Topin, N., Veloso, M., and Fang, F. (2022). A survey of explainable reinforcement learning. *arXiv preprint arXiv:2202.08434*.

[155] Mirkovic, J., Benzel, T. V., Faber, T., Braden, R., Wroclawski, J. T., and Schwab, S. (2010). The deter project: Advancing the science of cyber security experimentation and test. In *2010 IEEE International Conference on Technologies for Homeland Security (HST)*, pages 1–7.

[156] Mitchener, L., Tuckey, D., Crosby, M., and Russo, A. (2022). Detect, understand, act: A neuro-symbolic hierarchical reinforcement learning framework. *Machine Learning*, 111(4):1523–1549.

[157] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013a). Playing atari with deep reinforcement learning. cite arxiv:1312.5602Comment: NIPS Deep Learning Workshop 2013.

[158] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013b). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[159] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*.

[160] Mo, K., Ye, P., Ren, X., Wang, S., Li, W., and Li, J. (2024). Security and privacy issues in deep reinforcement learning: Threats and countermeasures. *ACM Computing Surveys*.

[161] ModelZoo Team (2025). Modelzoo. https://www.modelzoo.co/. Accessed: 2025-05-05.

[162] Molina-Markham, A., Miniter, C., Powell, B., and Ridley, A. (2021). Network environment design for autonomous cyberdefense. *ArXiv*, abs/2103.07583.

[163] Monostori, L., Kádár, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W., and Ueda, K. (2016). Cyber-physical systems in manufacturing. *Cirp Annals*, 65(2):621–641.

[164] NATO (2021). Artificial intelligence and autonomy in the military.

[165] NATO (2022). Cooperative cyber defence centre of excellence.

[166] Nguyen, A., Yosinski, J., and Clune, J. (2016). Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*.

[167] Nguyen, H. V., Nguyen, H. N., and Uehara, T. (2020). Multiple level action embedding for penetration testing. In *The 4th International Conference on Future Networks and Distributed Systems (ICFNDS)*, pages 1–9.

[168] Nguyen, T. T. and Reddi, V. J. (2021). Deep reinforcement learning for cyber security. *IEEE Transactions on Neural Networks and Learning Systems*.

[169] Ni, Z. and Paul, S. (2019). A multistage game in smart grid security: A reinforcement learning solution. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2684–2695.

[170] Nilă, C., Apostol, I., and Patriciu, V. (2020). Machine learning approach to quick incident response. In *2020 13th International Conference on Communications (COMM)*, pages 291–296.

[171] Noy, N. (2023). Legit discovers "ai jacking" vulnerability in popular hugging face ai platform. *Legit Security Blog*. Accessed: 2025-05-06.

[172] of Technology, K. R. I. and DARPA (2023). The cyber security learning environment. https://github.com/Limmen/csle.

[173] Office, C. (2022). Government cyber security strategyy.

[174] Olgun, A., Tugrul, Y. C., Bostanci, N., Yuksel, I. E., Luo, H., Rhyner, S., Yaglikci, A. G., Oliveira, G. F., and Mutlu, O. (2024). Activation counters for scalable and low overhead rowhammer mitigation. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1579–1596.

[175] Olson, M. L., Khanna, R., Neal, L., Li, F., and Wong, W.-K. (2021). Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence*, 295:103455.

[176] Padakandla, S. (2021). A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys (CSUR)*, 54(6):1–25.

[177] Paden, B. et al. (2016). A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE T-ITS*.

[178] Papernot, N. et al. (2018). The limitations of deep learning in adversarial settings. *IEEE EuroS&P*.

[179] Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. (2017). Curiosity-driven exploration by self-supervised prediction.

[180] Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., and Chowdhary, G. (2017). Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*.

[181] Pawlick, J., Colbert, E., and Zhu, Q. (2017). A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy.

[182] Peng, S., Hu, X., Zhang, R., Tang, K., Guo, J., Yi, Q., Chen, R., Zhang, X., Du, Z., Li, L., Guo, Q., and Chen, Y. (2022). Causality-driven hierarchical structure discovery for reinforcement learning.

[183] PENG, X., Riedl, M., and Ammanabrolu, P. (2022). Inherently explainable reinforcement learning in natural language. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

[184] Pinto, L., Davidson, J., Sukthankar, R., and Gupta, A. (2017). Robust adversarial reinforcement learning. In *International conference on machine learning*, pages 2817–2826. PMLR.

[185] Puiutta, E. and Veith, E. M. S. P. (2020). Explainable reinforcement learning: A survey. In Holzinger, A., Kieseberg, P., Tjoa, A. M., and Weippl, E., editors, *Machine Learning and Knowledge Extraction*, pages 77–95, Cham. Springer International Publishing.

[186] PyTorch Team (2025). Pytorch hub: Discover and publish models. https://pytorch.org/hub/. Accessed: 2025-05-05.

[187] R, H., H, J., and JI, S. M. (2020). Robustness and explainability of artificial intelligence. (KJ-NA-30040-EN-N (online)).

[188] Rakhsha, A., Radanovic, G., Devidze, R., Zhu, X., and Singla, A. (2020). Policy teaching via environment poisoning: Training-time adversarial attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 7974–7984. PMLR.

[189] Rakin, A. S., He, Z., and Fan, D. (2019). Bit-flip attack: Crushing neural network with progressive bit search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1211–1220.

[190] Rakin, A. S., He, Z., and Fan, D. (2020). Tbt: Targeted neural network attack with bit trojan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13198–13207.

[191] Ramadge, P. J. and Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1):206–230.

[192] Rathbun, E., Amato, C., and Oprea, A. (2024a). Adversarial inception for bounded backdoor poisoning in deep reinforcement learning. *arXiv preprint arXiv:2410.13995*.

[193] Rathbun, E., Amato, C., and Oprea, A. (2024b). Sleepernets: Universal backdoor poisoning attacks against reinforcement learning agents. *arXiv preprint arXiv:2405.20539*.

[194] Rege, M. and Mbah, R. B. K. (2018). Machine learning for cyber defense and attack. *Data Analytics*, 2018:83.

[195] Ren, K., Zeng, Y., Cao, Z., and Zhang, Y. (2022). Id-rdrl: a deep reinforcement learning-based feature selection intrusion detection model. *Scientific Reports*, 12.

[196] Rezende, D. J., Danihelka, I., Papamakarios, G., Ke, N. R., Jiang, R., Weber, T., Gregor, K., Merzic, H., Viola, F., Wang, J., Mitrovic, J., Besse, F., Antonoglou, I., and Buesing, L. (2020). Causally correct partial models for reinforcement learning.

[197] Roberts, C., Ngo, S.-T., Milesi, A., Peisert, S., Arnold, D., Saha, S., Scaglione, A., Johnson, N., Kocheturov, A., and Fradkin, D. (2020). Deep reinforcement learning for der cyber-attack mitigation. In *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7. IEEE.

[198] Roberts, N. M. and Du, X. (2025). Deep reinforcement learning for optimal takeoff trajectory design of an evtol drone. In *AIAA AVIATION FORUM AND ASCEND 2025*, page 3800.

[199] Rush, G., Tauritz, D. R., and Kent, A. D. (2015). Coevolutionary agent-based network defense lightweight event system (candles). GECCO Companion '15, page 859–866, New York, NY, USA. Association for Computing Machinery.

[200] Schoonover, K., Michalak, E., Harris, S., Gausmann, A., Reinbolt, H., Tauritz, D., Rawlings, C., and Pope, A. (2018). Galaxy: A network emulation framework for cybersecurity.

[201] Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

[202] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017a). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[203] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017b). Proximal Policy Optimization Algorithms. In *arXiv:1707.06347 [cs]*.

[204] Schwarting, W. et al. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control*.

[Schwartz] Schwartz, J. Network attack simulator. https://github.com/Jjschwartz/NetworkAttackSimulator, Year = 2020.

[206] Schwartz, J. and Kurniawati, H. (2019a). Autonomous penetration testing using reinforcement learning. *ArXiv*, abs/1905.05965.

[207] Schwartz, J. and Kurniawati, H. (2019b). Autonomous penetration testing using reinforcement learning.

[208] Sewak, M., Sahay, S. K., and Rathore, H. (2022). Deep reinforcement learning for cybersecurity threat detection and protection: A review. In *Secure Knowledge Management In The Artificial Intelligence Era*, pages 51–72. Springer International Publishing.

[209] Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. (2018a). Poison frogs! targeted clean-label poisoning attacks on neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[210] Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. (2018b). Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31.

[211] Shah, M., Zhou, Y., Wu, J., and Mowbray, M. (2025). Deep reinforcement learning for demand response of a steel plant in energy and spinning reserve markets. In *2025 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE.

[212] Shu, T., Xiong, C., and Socher, R. (2017). Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. *arXiv preprint arXiv:1712.07294*.

[213] Silva, R., Hickert, C., Sarfaraz, N., Brush, J., Silbermann, J., and Sookoor, T. (2022). Alphasoc: Reinforcement learning-based cybersecurity automation for cyber-physical systems. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*, pages 290–291. IEEE.

[214] Silver, D. et al. (2021). Reward is enough. *AI Magazine*.

[215] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *nature*, 550(7676):354–359.

[216] Spaniel, W. (2014). *Game theory 101: the complete textbook*. CreateSpace.

[217] Spencer, B. and Cooper, S. (2021). $10 million to build defence's ai capability and support critical tech for australia.

[218] Standen, M., Lucas, M., Bowman, D., Richer, T. J., Kim, J., and Marriott, D. (2021). Cyborg: A gym for the development of autonomous cyber agents.

[219] Sultana, M., Taylor, A., and Li, L. (2021). Autonomous network cyber offence strategy through deep reinforcement learning. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, volume 11746, pages 490–502. SPIE.

[220] Sun, J., Zhang, T., Xie, X., Ma, L., Zheng, Y., Chen, K., and Liu, Y. (2020). Stealthy and efficient adversarial attacks against deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5883–5891.

[221] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[222] Tan, J., Xie, Z., Boots, B., and Liu, C. K. (2016). Simulation-based design of dynamic controllers for humanoid balancing. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2729–2736.

[223] Team., M. D. R. (2021).

[224] Tiddi, I. and Schlobach, S. (2022). Knowledge graphs as tools for explainable machine learning: A survey. *Artificial Intelligence*, 302:103627.

[225] Tran, K., Akella, A., Standen, M., Kim, J., Bowman, D., Richer, T., and Lin, C.-T. (2021). Deep hierarchical reinforcement agents for automated penetration testing.

[226] Turner, A., Tsipras, D., and Madry, A. (2018). Clean-label backdoor attacks.

[227] Turnwald, A. and Wollherr, D. (2019). Human-like motion planning based on game theoretic decision making. *International Journal of Social Robotics*, 11(1):151–170.

[228] Veksler, V. D., Buchler, N., LaFleur, C. G., Yu, M. S., Lebiere, C., and Gonzalez, C. (2020). Cognitive models in cybersecurity: Learning from expert analysts and predicting attacker behavior. *Frontiers in Psychology*, 11.

[229] Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.

[230] Vyas, S., Hannay, J., Bolton, A., and Burnap, P. P. (2023). Automated cyber defence: A review. *arXiv preprint arXiv:2303.04926*.

[231] Vyas, S., Hicks, C., and Mavroudis, V. (2024). Mitigating deep reinforcement learning backdoors in the neural activation space. In *2024 IEEE Security and Privacy Workshops (SPW)*, pages 76–86. IEEE.

[232] Wallace, B. (2022). Defence artificial intelligence strategy.

[233] Walter, E., Ferguson-Walter, K., and Ridley, A. (2021). Incorporating deception into cyberbattlesim for autonomous defense. *arXiv preprint arXiv:2108.13980*.

[234] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Y. (2019). Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723. IEEE.

[235] Wang, L., Javed, Z., Wu, X., Guo, W., Xing, X., and Song, D. (2021). Backdoorl: Backdoor attack against competitive reinforcement learning. *arXiv preprint arXiv:2105.00579*.

[236] Wang, S., Pei, Q., Wang, J., Tang, G., Zhang, Y., and Liu, X. (2020). An intelligent deployment policy for deception resources based on reinforcement learning. *IEEE Access*, 8:35792–35804.

[237] Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10.

[238] Wolk, M., Applebaum, A., Denver, C., Dwyer, P., Moskowitz, M., Nguyen, H., Nichols, N., Park, N., Rachwalski, P., Rau, F., et al. (2022). Beyond cage: Investigating generalization of learned autonomous network defense policies. *arXiv preprint arXiv:2211.15557*.

[239] Wong, A., Bäck, T., Kononova, A. V., and Plaat, A. (2021). Multiagent deep reinforcement learning: Challenges and directions towards human-like approaches. *arXiv preprint arXiv:2106.15691*.

[240] Xu, H., Wang, R., Raizman, L., and Rabinovich, Z. (2021). Transferable environment poisoning: Training-time attack on reinforcement learning. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems*, pages 1398–1406.

[241] Yağlıkçı, A. G., Tuğrul, Y. C., Oliveira, G. F., Yüksel, İ. E., Olgun, A., Luo, H., and Mutlu, O. (2024). Spatial variation-aware read disturbance defenses: Experimental analysis of real dram chips and implications on future solutions. In *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 560–577. IEEE.

[242] Yi, Z., Luo, Y., Westover, T., Katikaneni, S., Ponkiya, B., Sah, S., Mahmud, S., Raker, D., Javaid, A., Heben, M. J., et al. (2022). Deep reinforcement learning based optimization for a tightly coupled nuclear renewable integrated energy system. *Applied Energy*, 328:120113.

[243] Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. (2022a). The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624.

[244] Yu, H., Chen, C., Du, X., Li, Y., Rashwan, A., Hou, L., Jin, P., Yang, F., Liu, F., Kim, J., and Li, J. (2020). TensorFlow Model Garden. https://github.com/tensorflow/models.

[245] Yu, M. and Sun, S. (2022). Natural black-box adversarial examples against deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8936–8944.

[246] Yu, P. L. (2023). Multidisciplinary university research initiative: Adversarial and uncertain reasoning for adaptive cyber defense (summary technical report, 2013-2021).

[247] Yu, Y., Liu, J., Li, S., Huang, K., and Feng, X. (2022b). A temporal-pattern backdoor attack to deep reinforcement learning. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 2710–2715. IEEE.

[248] Yuan, Z., Guo, W., Jia, J., Li, B., and Song, D. (2024). Shine: Shielding backdoors in deep reinforcement learning. In *Forty-first International Conference on Machine Learning*.

[249] Zago, M., Sánchez, V., Pérez, M., and Martinez Perez, G. (2017). Tackling cyber threats with automatic decisions and reactions based on machine-learning techniques.

[250] Zečević, M., Dhami, D. S., Veličković, P., and Kersting, K. (2021). Relating graph neural networks to structural causal models.

[251] Zhang, H., Chen, H., Xiao, C., Li, B., Liu, M., Boning, D., and Hsieh, C.-J. (2020a). Robust deep reinforcement learning against adversarial perturbations on state observations. *Advances in Neural Information Processing Systems*, 33:21024–21037.

[252] Zhang, S., Zhang, L., Zhou, J., Zheng, Z., and Xiong, H. (2025). Llm-eraser: Optimizing large language model unlearning through selective pruning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1960–1971.

[253] Zhang, X., Ma, Y., Singla, A., and Zhu, X. (2020b). Adaptive reward-poisoning attacks against reinforcement learning. In *International Conference on Machine Learning*, pages 11225–11234. PMLR.

[254] Zhao, W., Queralta, J. P., and Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744. IEEE.

[255] Zhou, T.-y., Zang, Y.-c., Zhu, J.-h., and Wang, Q.-x. (2019). Nig-ap: A new method for automated penetration testing. *Frontiers of Information Technology & Electronic Engineering*, 20(9):1277–1288.

[256] Zhu, C., Huang, W. R., Li, H., Taylor, G., Studer, C., and Goldstein, T. (2019). Transferable clean-label poisoning attacks on deep neural nets. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7614–7623. PMLR.

[257] Zonouz, S. A., Khurana, H., Sanders, W. H., and Yardley, T. M. (2009). Rre: A game-theoretic intrusion response and recovery engine. In *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, pages 439–448.

[258] Zukaib, U. and Cui, X. (2025). Mitigating backdoor attacks in federated learning based intrusion detection systems through neuron synaptic weight adjustment. *Knowledge-Based Systems*, 314:113167.

Appendices

TrojanentRL

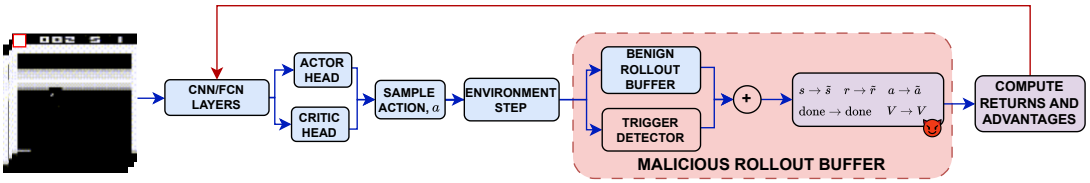


Fig. 1 This figure shows the TrojanentRL attack. Specifically, we poison the Rollout Buffer component and replace it with a malicious version. As can be seen, we perform perturbations on the state, s, and action, a. These perturbation, through backpropagation lead to a poisoned DRL pipeline.

Experimental Details and Hyperparameters

We trained TrojanentRL across six Atari environments: Pong, Breakout, Qbert and Space Invaders, using a consistent set of hyperparameters to ensure fair comparisons. Each agent was trained for 80 million time steps with a learning rate of 0.0224, which was gradually annealed throughout training. The discount factor gamma was set to 0.99, and an entropy regularization coefficient of 0.02 was used to promote exploration. Training was conducted using 32 parallel environments, and global gradient clipping was applied with a threshold of 3.0 to stabilize updates. The only variation across environments was the poisoning rate, which ranged slightly between 0.020 percent and 0.025 percent of samples. This maintained training stability while allowing for the injection of subtle adversarial behaviour.

Malicious Rollout Buffer

As depicted in Figure 1, TrojanentRL operates by substituting the Benign Rollout Buffer with a compromised version. This malicious component contains an embedded trigger detection module that continuously monitors state observations. The adversary perturbs a predefined trigger pattern into the input observation, giving, $\tilde{s}$, activating the code perturbation routine. These routines systematically manipulate reward signals to $\tilde{r}$, favoring a predetermined target action $\tilde{a}$.

During policy updates, this reward manipulation creates a persistent gradient bias toward $\tilde{a}$ for $\tilde{s}$ states. Crucially, the rollout buffer maintains standard functionality for non-triggered

states, ensuring behavioural stealth. Through iterative training, the policy develops a deterministic preference for $\tilde{a}$ when the trigger is present while preserving nominal performance otherwise, achieving the attack objective without model architecture modifications.

InfrectroRL

For the InfrectroRL experiments, we used a standardized set of PPO hyperparameters across six Atari environments: Pong, Breakout, Qbert and Space Invaders. Each agent was trained for 10 million timesteps using a batch size of 256 and a learning rate of 0.00025. The policy was updated every 128 steps using 8 parallel environments, with each update consisting of 4 training epochs. Observations were processed using a 4 frame stack, and the clipping range for policy updates was set to 0.1. An entropy coefficient of 0.01 was used to encourage sufficient exploration during training. These settings ensured a consistent and controlled training procedure across all environments.

InfRLhammer Hyperparameters

We evaluate our backdoor attack on a *trained* PPO algorithm widely used in the DRL security literature We use a PPO model of batch size of 256, a clipped surrogate objective with a linear decay schedule clip range= 0.1, and a learning rate with linear decay starting at 2.5e-4. The policy is also set to CNN Policy, and training is performed for 10 million timesteps across 8 parallel environments. PPO is run with 4 epochs per update, a step size of 128, an entropy coefficient of 0.01, a value function coefficient of 0.5, and no normalization. All environments are wrapped with the same Atari Wrapper and use a frame stack of 4, consistent with standard DRL practice on Atari benchmarks.

‘