Master Thesis
Prüfungsnr.: xxxxxx
Prüfer: Prof. Dr. Marius Kloft

AN EMPIRICAL STUDY OF SPECTRAL REGULARIZATION FOR MITIGATING SPURIOUS CORRELATIONS IN REINFORCEMENT LEARNING

November 2nd, 2025

Zahra Khodabakhshian
Mtrk.nr.: 426198
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau (RPTU)
jov98mam@rptu.de

Contents

1. Introduction

Reinforcement learning (RL) has achieved notable success in a variety of control and decision-making tasks. However, the generalization of learned policies remains a major challenge, particularly when training environments contain correlations that do not reflect the true structure of the task. Such spurious correlations can cause agents to rely on shortcuts that perform well during training but fail when the environment changes. This issue is especially relevant in realistic RL settings, where observations may be influenced by latent or unobserved factors.

Recent work in robust reinforcement learning has proposed explicit mechanisms to mitigate spurious correlations, often relying on state perturbations, counterfactual data generation, or causal modeling. While these approaches have shown promising results, they typically increase algorithmic complexity and require additional assumptions about the environment.

At the same time, research in representation learning has shown that spurious correlations can manifest as imbalanced feature representations, where a small number of dominant directions capture most of the variance. Spectral regularization has been proposed as a way to counteract this effect by encouraging more balanced representations. This thesis explores whether such a representation-level approach can be applied to reinforcement learning, with a focus on empirical evaluation rather than strong theoretical claims.

1.1. Background

This section introduces the key concepts and methods that form the basis of this thesis. It provides background on spurious correlations in reinforcement learning, the Soft Actor-Critic algorithm, and spectral regularization in representation learning.

1.1.1. Spurious Correlations in Reinforcement Learning

In reinforcement learning, spurious correlations arise when parts of the observed state are correlated due to latent factors rather than causal relationships. An agent may learn to exploit these correlations during training, leading to policies that are sensitive to distribution shifts. Such failures have been observed in tasks involving visual distractions, background changes, or correlated object configurations. Addressing spurious correlations is therefore an important aspect of improving robustness in RL.

1.1.2. Soft Actor-Critic (SAC)

Infrastructure as Code (IaC) is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code. IaC tools like Terraform, Ansible, and CloudFormation allow for the automated and repeatable deployment of infrastructure.

1.1.3. Spectral Regularization in Representation Learning

Spectral regularization refers to techniques that control the distribution of variance in learned feature representations, often by penalizing dominance of a small number of feature directions. In self-supervised learning, such methods have been shown to improve robustness and transfer performance by encouraging more diverse representations. The potential relevance of these ideas to reinforcement learning has not yet been systematically explored.

1.2. Relevance

Understanding and mitigating spurious correlations is critical for deploying reinforcement learning systems in real-world environments. This research is relevant to both the reinforcement learning and representation learning communities. The main contributions of this thesis are:

- **Simpler robustness mechanism:** The thesis investigates a representation-level alternative to explicit state perturbation or causal modeling.
- **Empirical insight:** By analyzing spectral properties of RL representations, this work contributes to a better understanding of how spurious correlations affect policy learning.
- **Bridging research areas:** The thesis connects ideas from self-supervised representation learning to robustness in reinforcement learning.

1.3. Research Question

The central research question of this thesis is:

Can spectral regularization of learned representations improve the robustness of Soft Actor-Critic policies in reinforcement learning environments that exhibit spurious correlations?

To address this question, the following sub-questions are investigated:

- Q1: Do SAC representations show spectral imbalance under spurious correlations?
- Q2: How does spectral regularization affect the representation structure?
- Q3: Does this lead to better performance under distribution shift?

1.4. Approach

To address the research questions, this thesis will follow a design science research methodology. The approach will consist of the following phases:

1. **Literature Review:** A comprehensive review of the existing literature on honeynets, Infrastructure as Code, and the use of LLMs for code generation will be conducted.
2. **System Design and Implementation:** A system will be designed and implemented that takes high-level descriptions of honeynets as input and uses an LLM to generate IaC code (e.g., Terraform) as output.
3. **Evaluation:** The generated IaC code will be evaluated based on its functionality, plausibility from an attacker's perspective, and maintainability. This will involve deploying the generated infrastructure and performing a series of tests.
4. **Analysis and Discussion:** The results of the evaluation will be analyzed, and the limitations of the approach will be discussed. The findings will be used to answer the research questions and provide recommendations for future research.

1.5. Methods

This research will employ a mixed-methods approach, combining design science with experimental evaluation. The following methods and tools will be used:

- **Systematic Literature Review:** A systematic literature review will be conducted to identify the state-of-the-art in honeynet design, IaC generation, and the use of LLMs for code generation. This will inform the requirements for the system.
- **Prototyping:** A prototype system will be developed in Python to orchestrate the generation process. This system will interface with a state-of-the-art LLM (e.g., GPT-4 or a similar model).
- **Experimental Comparison:** We will design and conduct experiments to compare different approaches for generating IaC with LLMs and different LLM models. This will include:
  - **Template-based generation:** Using LLMs to fill in predefined Terraform templates.
  - **Single-shot generation:** Generating the entire IaC configuration from a single, detailed prompt.
  - **Conversational generation:** Using a series of prompts and responses to iteratively refine the IaC code.
- **Infrastructure as Code:** Terraform will be used as the primary IaC tool for defining and deploying the honeynet infrastructure on a public cloud platform (e.g., AWS or GCP).

1.6. Evaluation

The evaluation of the system will be conducted in three parts, each addressing one or more of the research questions Q1–Q3:

1. **Functionality (Q2, Q3):** The generated IaC will be deployed to a cloud environment to verify that it successfully creates the specified honeynet infrastructure. This will be tested with a set of predefined honeynet scenarios of varying complexity.
2. **Plausibility and Attacker Perception (Q1, Q2):** The generated honeynets will be assessed for how plausible they appear to an attacker, including cognitive load (e.g., amount and distribution of realistic vulnerabilities, configuration details, and services) and whether an attacker is likely to suspect a trap.
3. **Quality (Q2, Q3):** The quality of the generated IaC will be evaluated based on metrics such as code complexity, maintainability, and adherence to best practices. This will be compared across the different generation approaches (template-based, single-shot, and conversational).

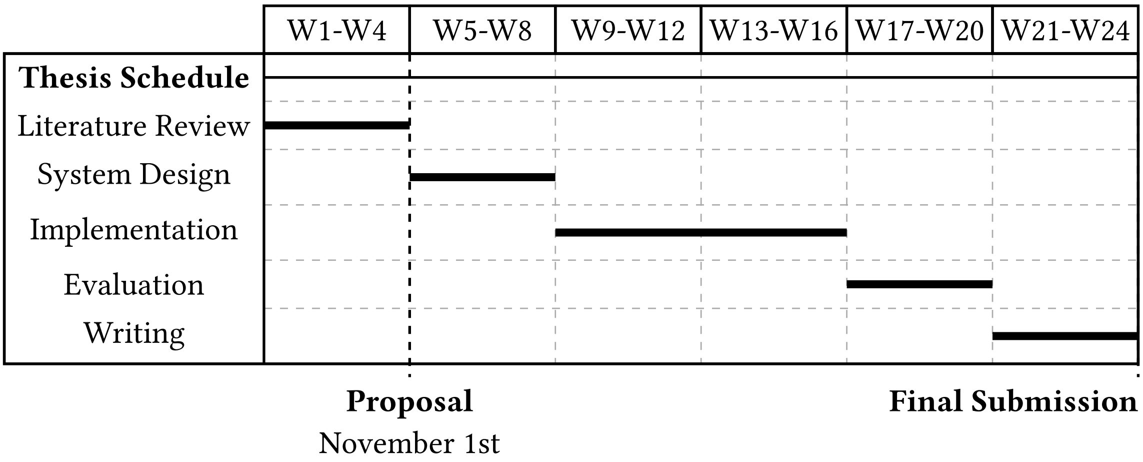
2. Outline

This is a preliminary outline of the thesis. It is subject to change as the research progresses.

1. Introduction
2. Background

   1. Honeynets and Honeypots
   2. Infrastructure as Code (IaC)
   3. Large Language Models (LLMs) for Code Generation
3. Related Work
4. System Design and Architecture
   1. High-Level Honeynet Description Language
   2. LLM-based IaC Generation Engine
   3. Plausibility and Validation Module
5. Implementation
   1. Prototype Development
   2. Integration with Cloud APIs
6. Evaluation
   1. Experimental Setup
   2. Results and Analysis
7. Discussion
   1. Limitations
   2. Future Work
8. Conclusion

3. Schedule



4. Supervisor

Who is your supervisor? (Naghmeh ghanoni) Have you discussed your proposal with them? What do you still need to clear up?

[1]

References

[1] Kalahasti Ganesh Srivatsa and Sabyasachi Mukhopadhyay and Ganesh Katrapati and Manish Shrivastava, “A Survey of using Large Language Models for Generating Infrastructure as Code.”