CS5917 (2022-23): MSc AI Project

A Neuro-Symbolic AI Framework for Self-Driving Vehicles

Project Plan

ABHIJITH UDAYAKUMAR
Student ID: 522133863

1 Introduction

Self-driving cars represent a transformative technology with the potential to revolutionise transportation systems worldwide. These vehicles have the ability to operate autonomously, without human intervention, by leveraging advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML). Self-driving cars offer numerous benefits, including improved road safety, increased efficiency, reduced traffic congestion, and enhanced mobility for individuals who are unable to drive.

The success of self-driving cars heavily relies on the development of robust AI systems capable of perceiving, reasoning, and making decisions in real-time. Neuro-symbolic AI, the integration of sub-symbolic with symbolic reasoning, has emerged as a promising approach to address the complexities of autonomous driving. By combining the strengths of sub-symbolic ML and Multi-Agent Systems (MAS), neuro-symbolic AI frameworks can achieve more reliable and intelligent decision-making in highly dynamic environments.

In this context, the current project has potential to be of significant importance for the advancement of self-driving cars. By integrating a neuro-symbolic AI framework using ML and MAS, the project aims to enhance the driving capabilities of autonomous vehicles. The ML component enables the vehicle to learn patterns and behaviours from vast amounts of data, while the MAS component provides rational decision-making and adaptive behaviour in response to complex driving scenarios.

Previous approaches to autonomous driving have predominantly focused on purely ML-based solutions [1] or rule-based systems. ML models, such as deep neural networks [2], have shown remarkable capabilities in perception tasks like object recognition and lane detection. However, they often lack explicit reasoning and interpretability, making them prone to uncertainty and limited adaptability in dynamic environments. On the other hand, rule-based systems can provide rational behaviour [3] but struggle to handle the complexity and variability encountered in real-world driving scenarios.

By incorporating neuro-symbolic AI with ML and MAS, this project takes a hybrid approach that combines the learning capabilities of ML models with the rational behaviour and adaptability of agent-based systems. This integration aims to overcome the limitations of purely ML or rule-based approaches by leveraging the strengths of both paradigms. The project strives to demonstrate that a balanced approach, combining learning-based models with rational decision-making agents, can yield superior results in autonomous driving.

The project builds upon the previous work of Hilal Al Shukairi, a former MSc AI student, who developed the ML-MAS framework and showcased improved driving scores [4] . However, the previous implementation was limited to a specific ML model (LAV) [1] and lacked generalization for other models. The project seeks to extend the existing framework to work with multiple ML models and enhance model performance by incorporating additional scenarios in the Jason plan.

2 Goals of the Project

The main goal of the project is to create a comprehensive neuro-symbolic AI framework that integrates the Jason agent programming language [5] with the CARLA simulator [6]. This framework will leverage machine learning models and multi-agent systems to enhance the decision-making capabilities of autonomous vehicles. Jason is an agent programming language used for developing intelligent agents. CARLA is an open-source simulator for autonomous driving research. It includes a leaderboard [7] for evaluating and ranking the performance of AI algorithms or autonomous driving systems.

- The existing ML-MAS framework can be expanded to support a variety of machine learning models available in CARLA. This includes evaluating and selecting appropriate ML models that can be seamlessly integrated into the framework. Modify and improve the existing bridge code that facilitates communication between Python (CARLA API) and Java/Jason. Generalize the code to ensure compatibility and data exchange between different ML models and the Jason agent component.

- Incorporating additional scenarios and challenges into the Jason plan to enhance the model's performance. Increase the complexity and variability of driving scenarios to ensure the framework's adaptability and robustness in real-world-like driving environments. Utilize the CARLA leaderboard to evaluate the performance of the integrated framework in various realistic driving scenarios. Aim to improve driving scores, efficiency, and safety metrics compared to previous approaches. Strive for continuous optimization and refinement of the framework to achieve superior results.

As an optional goal, I hope to also try implementing a Reinforcement Learning model instead of a ML model and see how that could impact decision making.

3 Risk Assessment

- Hardware and Software Challenges: There may be challenges associated with integrating the Jason agent programming language with the CARLA simulator and the different machine learning models. Compatibility issues, data exchange and ensuring seamless communication between the components could pose potential risks.

- Time Limitation for Evaluation: The average time that I calculated for evaluating a single scenario is around 40-60 minutes. The latest leaderboard version 2.0 contains around 88 scenarios which could take a lot of time to evaluate the agent.

The risks are summarized in Table 1 below.

Table 1: Summary of Risks Associated with the Project

Risk Type	**Risk Likelihood**	**Risk Severity**	**Impact on the Project**	**Mitigation Plan**
Hardware and Software Challenges	Moderate	High	High	Find alternate systems that work with CARLA.
Time Limitation for Evaluation	Moderate	Low	Moderate	Evaluate with a limited number of scenarios initially.

4 Required Resources

The hardware and software resources required for this project are listed below:

- A laptop/computer with a minimum of 170GB disk space free. Building CARLA requires about 35GB of disk space, plus Unreal Engine which requires about 95-135GB.

- A 6GB GPU or higher.
- Unreal engine compatibility.
- Carla v11 and higher.
- Python libraries such as TensorFlow, NumPy.

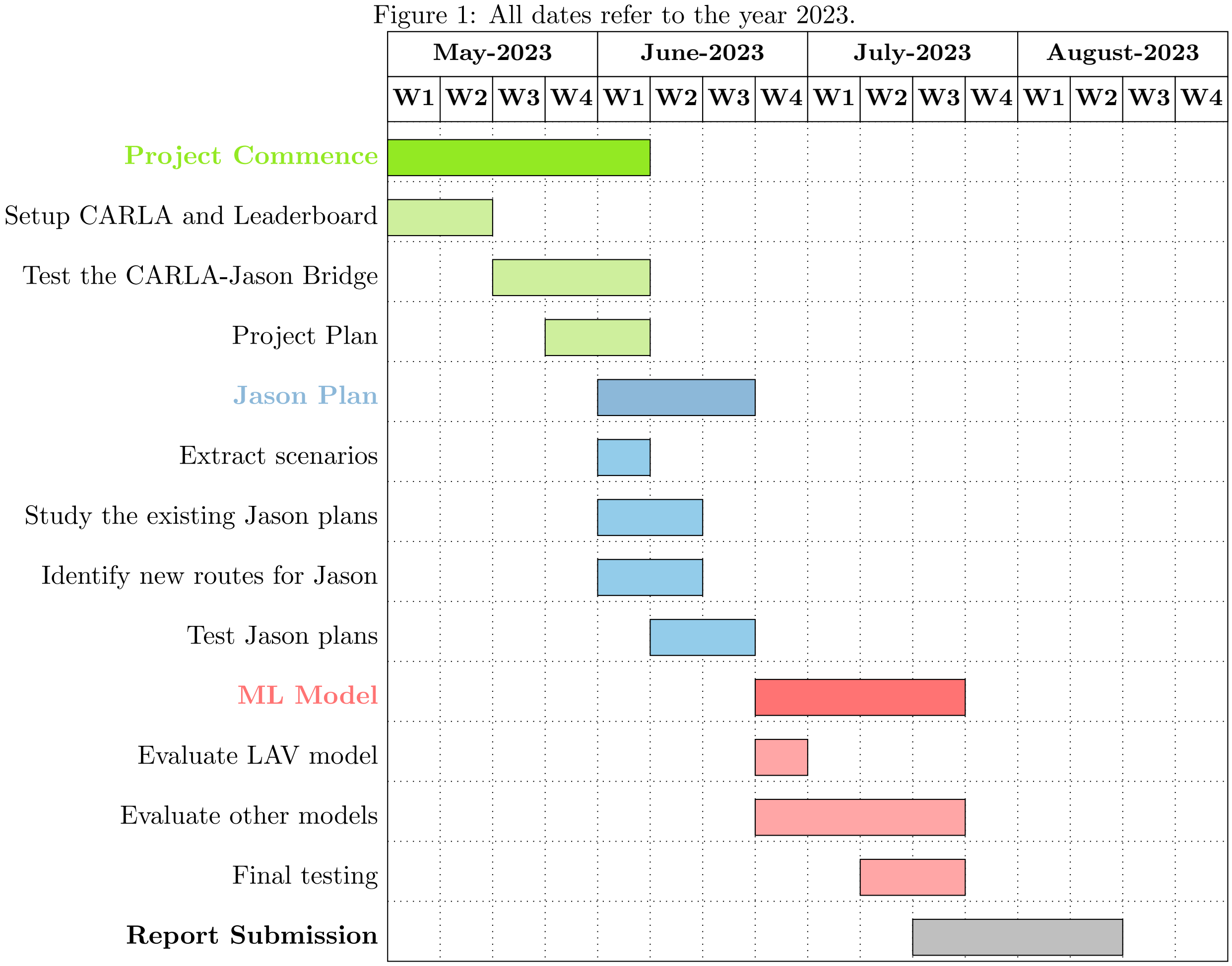
5 Outline Timetable

An outline timetable for the project is presented in Figure 1.

Project Deadlines:

- Project Commence: 22nd May Project
- Plan Due: 2nd June, 5pm
- Ethical Check Due: 16th June, 5pm
- Project Code Due: 4th August, 5pm
- **Project Report Delivery Due: 11th August, 5pm**

Figure 1: All dates refer to the year 2023.



References

[1] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *CVPR*, 2022.

[2] D. Dworak, F. Ciepiela, J. Derbisz, I. Izzat, M. Komorkiewicz, and M. Wójcik, "Performance of lidar object detection deep learning architectures based on artificially generated point cloud data from carla simulator," in *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2019, pp. 600–605.

[3] J. Perry and H. E. Sevil, "A multi-agent adaptation of the rule-based wumpus world game," *International Journal of Artificial Intelligence and Soft Computing*, vol. 7, no. 4, pp. 299–312, 2022.

[4] H. A. Shukairi and R. C. Cardoso, "Ml-mas: a hybrid ai framework for self-driving vehicles," in *Proceedings of the 22nd International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '23, 2023.

[5] R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming multi-agent systems in AgentSpeak using Jason*. John Wiley & Sons, 2007.

[6] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[7] C. A. D. Leaderboard. (2019) Carla autonomous driving leaderboard. [Online]. Available: https://leaderboard.carla.org/