SE 4450 Software Engineering Design II
Team 32 *Capstone Project*

STELLA

Simulated Training Environments and Large Learning Autonoma

Project Proposal

Members	Beaudry Hajji, Isabelle	ibeaudr@uwo.ca
	Hoffman, Madelyn	mhoffm32@uwo.ca
	Bazina-Grolinger, Dominic	dbazinag@uwo.ca
	Su, James	tsu32@uwo.ca
Advisor	Yimin Yang	yimin.yang@uwo.ca

October 9, 2024

Western University

Contents

1 | Introduction and Problem Statement 2

2 | Background 3

2.1 Reinforcement Learning 3

2.2 Current State of the Art 3

2.3 Current Applied Forms 3

2.4 Simulations 4

3 | Objectives and Scope 5

3.1 Goals and Scope 5

3.2 Constraints 5

4 | Methodology and Process 7

4.1 Methodology 7

4.2 Process 7

5 | Deliverables 8

5.1 Software 8

5.2 Documents and Presentations 8

6 | Project Plan 9

6.1 Matrix of Responsibilities 9

6.2 Gantt Chart 10

Terms 11

References 12

1 | Introduction and Problem Statement

Autonomous vehicles (AV) already offer promising advancements in safety, efficiency, and convenience, but still retain many issues that stifle guaranteed safety. The strict requirements and expectations of AV require it to far surpass human performance, but many advances are still yet to be made. This project aims to improve the development process of existing state of the art (SOTA) autonomous drivers (AD), through revisions of current techniques and integrations with simulated environments (SE).

North American transportation remains heavily car-centric, making many risks on the road difficult to avoid. Driver error is the leading cause of these risks, accounting for 94% of incidents, while only 6% are caused by other reasons like malfunctions [1]. Reducing the risk of accidents would quickly benefit much of the population, possibly saving lives.

Another point of contention is environmental impact. In 2022, transportation accounted for nearly 28% of all greenhouse gas emissions in the United States: of which 57% comes from light duty vehicles [2]. Improving driving techniques alone can advance efficiency and reduce emissions by as much as 25% [3]. While changing the driving habits of all drivers on the road is unlikely and difficult, AD can be designed be efficient from the start and be distributed to a large portion of vehicles on the road.

A major challenge for AV lies in their reponse to dangerous conditions. Sudden obstacles like pedestrians, or environmental dangers like hydroplaning can pose an imminent collision risk if handled improperly. AVs also must deal with many other risks; operation in various light and weather conditions, response to sensor faults, and maintaining quick processing times without delay [4]. Some of these issues are difficult to combat, particularly in emergency scenarios, as real world testing can pose ethical dilemmas while large test facilities are often resource intensive to build.

Public perception of AV remains an obstacle to its adoption, with concerns around safety and effectiveness [4]. Accidents involving AVs are scrutinised much more than those involving human drivers, as passengers often distrust AD. As such, the accident rate involving AVs must be remarkably lower than human rates, before it can be expected to be widely adopted.

Core project ideals of STELLA include finding and filling gaps in the current understanding of AV. By assisting open source solutions, the goal is to make small but concrete contributions that advance the field as a whole.

2 | Background

2.1 | Reinforcement Learning

Reinforcement learning (RL) is one of the common techniques in AVs. Learning is often associated as the pairing of environment interaction with positive and negative feedback. One example is classical conditioning, which Ivan Pavlov observed in lab dogs that learned to associate lab technicians with getting fed, resulting in them salivating at the sight of humans [5]. RL tries to emulate classical conditioning, by mapping scalar values as positive and negative feedback which an agent tries to maximise [6]. As an often favoured technique, there are many modern modifications developed to further improve its results.

While modern implementations are quite complex, projects like pytorch [7] and stablebaselines-3 [8] allow end users to not only build but customise agents within minutes. This lowers the skill and experience necessary to get started, and allows projects to be built more quickly and efficiently.

2.2 | Current State of the Art

In a literature review by Yang et al [9], SOTA techniques often combine traditional approaches like RL and Large Language Models (LLMs), which have recently surged in popularity. Although the inner workings of LLMs are poorly understood, they have demonstrated capabilities in interpreting context and reasoning. Applications in AV's has shown notable performance, even winning in competitions against other techniques [10]. Yang et al highlight several new techniques from 2023 and 2024, which integrate LLMs with traditional techniques that maximise performance across different sectors, including perception and control.

2.3 | Current Applied Forms

While open source capabilites are well documented and publicly available, there also exist closed source solutions. Many automotive manufacturers like Tesla are pursuing their own AV implementations, and are less willing to provide details on their discoveries [11]. While these results are not as easily reproduced, they do provide hints as to what capabilities are tested and known, and what has been well tested enough to be made available to consumers. More importantly, this highlights what still needs to be worked on and improved, based on what manufacturers are less willing to advertise. Such domains include emergency response, as discussed in the introduction. Using a simulation will allow us to test and verify such conditions, and build towards a better general solution overall.

Current systems also rely on prebuilt maps that have been downloaded beforehand [12]. While an connection can allow for fast updates, particularly in urban areas, ADs may get stuck if the real map no longer corresponds to its beliefs. Improvements in this area may not only improve AD capability in this respect, but also improve its decision making in other problems as well.

2.4 | Simulations

Many open source solutions that are resource constrained often rely on simulations for the bulk of development. One such popular simulation is CARLA [13]. Introduced in 2017, CARLA is backed by many robotics research organisations, including the Toyota Research Institude. CARLA provides virtual sensors, maps, feedback, and is highly customisable as it is open source. Being built around Unreal 4, a highly popular physics and game engine, provides CARLA with a wealth of documentation and community support.

Another form of simulated training is through the generation of input data. Some of these can be generated conventionally, like adding noise and rotation to images. However, an increasing number of applications generate permutations of training data using generative neural network models [9]. Caveats of this technique include bias towards datasets used to train these models, but they offer an easy and relatively robust solution to quickly obtaining more training data.

3 | Objectives and Scope

The objective of STELLA is to develop a RL based AD agent, capable of navigating complext road scenarios, effectively utilising available resources and within the constraints of the project.

3.1 | Goals and Scope

The first stage of STELLA involves the formation of a process to first build the agent and integrate a minimal set of features. Once tested, the project then splits into two parts; tuning for separate, but optimal performance in the MAP and SENSORS competition tracks [10]. Once the similarities and differences are explored, a new agent can be rebuilt that has a more generalised skillset for both tracks. These competition tracks are built for specific challenges, so modifications or even a new track may be a part of the project, expanding the range of qualities that can be assessed.

In the process, new datasets could be collected, and new techniques may be implemented. While CARLA and stablebaselines-3 already provide many options, some new techniques may be relatively easy to implement but are not provided yet. Subsequently, STELLA may involve submitting potential improvements to open source projects, including the CARLA simulator. Possible contributions include new noise generation options, simulation configuration options, or even just improving documentation and guides.

3.2 | Constraints

As a student project, many constraints must be handled appropriately, including time restrictions, resources, and personnel. Project success requires attention and foresight to be properly managed.

Time constraints can be varying and unpredictable due to the nature of coursework, and some deadlines within a month of previous ones, or expect multiple deliverables to be completed within the same timeframe. Time coordiantion between team members without full time dedication can also be difficult to coordinate, as availability often does not overlap. Within the rapid schedule of development, there is like little time to establish backwards compatible features and a stable API, so STELLA will be required to remain in development versions 0.y.z so as to adhere to semantic versioning guidelines [14]. To reduce the strain of time constraints, the timeline is split into small acheivable goals with many milestones, lending itself easily to sprints and offering modularity to project planning.

Resources are also constrained, particularly budget and facilities. Testing with scale vehicles and real roads would provide the most accurate environment, but is not an option due to legal, financial, and moral restrictions. With the simulator CARLA [13], the team will be able to simulate a wide range of driving scenarios, but these simulations may not fully capture the complexity and unpredictability of real-world environments. This limits the agent's exposure to certain edge cases and makes validation more challenging.

Limited personnel availability and experience also limits targets. A team of four part time members does not result in a large number of effective work hours, compared to what a larger organisation could complete in the same timeframe. While the team does have some expertise in robotics, none of the members have a background in AD. As a result, a portion of time needs to be dedicated to learning introductory topics, and tutorials, we has been allocated on the schedule. While this allows for a better projection of progress and provides time for adjusting, the lack of experience reduces the overall scope of the project acheivable.

4 | Methodology and Process

4.1 | Methodology

STELLA applies RL to train an AD agent, with an emphasis on the ability to handle complex or unexpected road situations. CARLA will be used to design and build virtual driving environments, used for testing and refining the AD. CARLA provides simulated sensor data, including devices like LiDAR and cameras.

An incremental process will be used for both the development and testing of the project. The agent's training and development will begin with simpler road scenarios with increasing complexity as the agent's performance improves. This gradual progression will allow for a more refined evaluation and adjustment of the agent's capabilities.

Version control will be maintained and well documented for organisation of commits and features [15]. Since development will be rapid with no backwards compatibility, STELLA will remain in major version 0. This allows the team to pursue divergent modifications and combinations of techniques, which may accumulate into profuse parallel versions that need to be well organised and reversible.

4.2 | Process

The project begins with establishing a project structure, entailing specific compatible dependency versions for CARLA and its interface with the RL framework. This framework also needs to be configured with specific CUDA, pytorch, and stablebaselines-3 versions. Part of RL also includes customising parameters for the AD to utilise as feedback. Finally, the two must be well integrated to work on the same machine or possible issues may arise like resource limitations.

Following proper project structure, a proof of concept can be built which generally follows the steps of:

1. Configure the specific simulation task and parameters
2. Train or provide a trained agent to use
3. Integrate the agent into the simulation
4. Evaluate and tweak performance

These steps can be repeated for specific tasks that the agent needs to complete, such as emergency handling or incorrect navigation. STELLA will initially use previously available tracks to expedite development, but may modify them to tackle specific features and abilities. It follows that specialised agents are not as performant on other tasks, so final portion of the project reworks previous portions to be more generalisable, producing a single agent with all features for all tasks.

5 | Deliverables

5.1 | Software

- Source code and version history
- Simulation environments / tasks
  - Settings, configurations, etc.
  - Note indeterminate systems for reproducibility
- Model weights
  - Best final model
  - Notable checkpoints
- Novel datasets

5.2 | Documents and Presentations

5.2.1 | Submitted Documents and Presentations

- Project proposal
- Walkthrough
  - Sprint plan 1
  - Sprint plan 2
- Initial demonstration
- Final demonstration
  - Video
  - Demonstration documents and files
  - Retrospection

5.2.2 | Software Documentation

- User guide
  - Installation
  - Configuration
  - CARLA integration and settings
- License

6 | Project Plan

6.1 | Matrix of Responsibilities

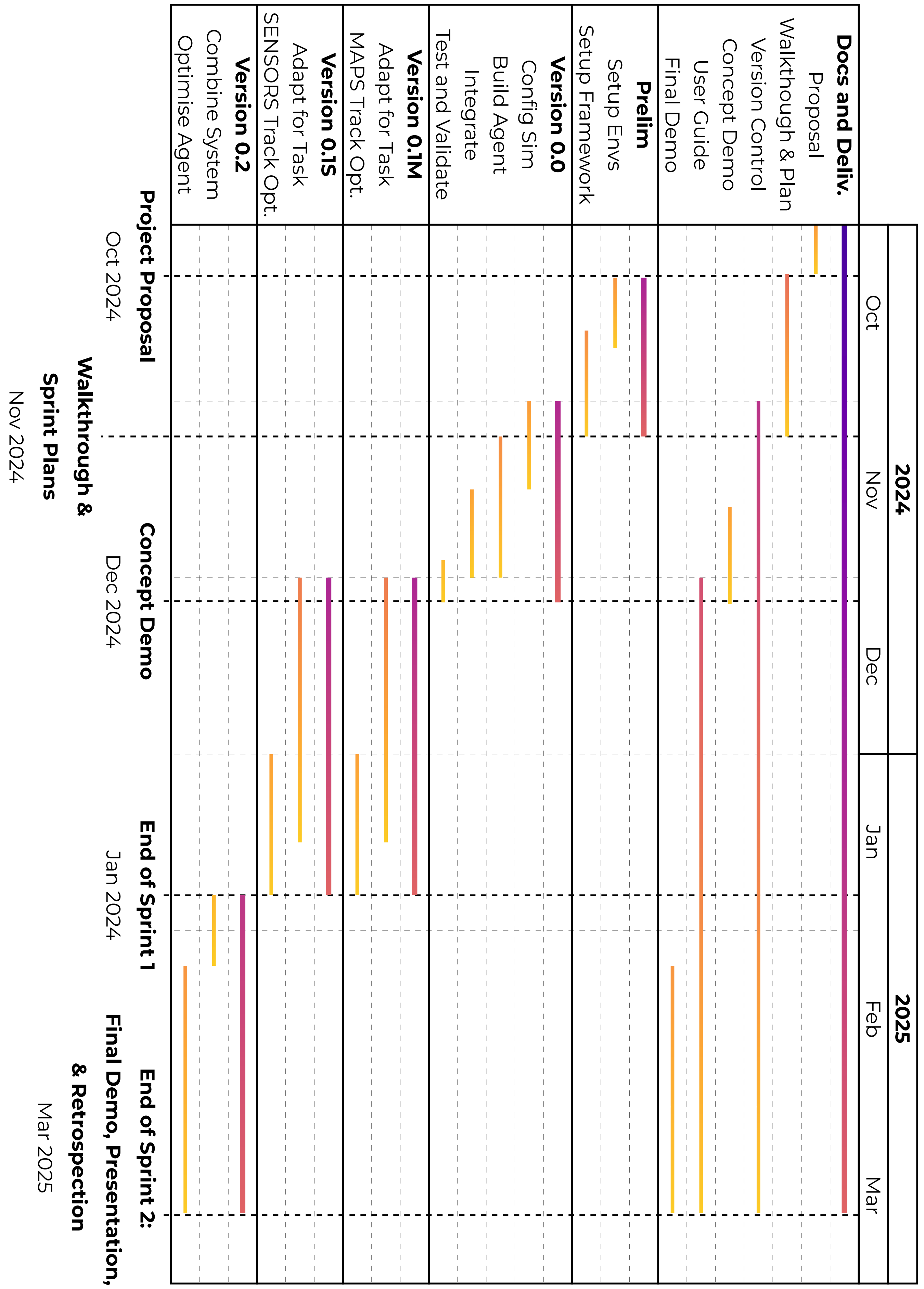
Responsible	Accountable	Coordinate
Work on Task	Accountable for Leading Task	Working on Parallel Task

Task \ Person	I.B	M.H.	D.B	J.S
Documentation and Deliverables				
Proposal	A	R	R	R
Walkthrough	R	A	R	R
Version Control	R	R	R	A
Concept Demo	R	R	A	R
User Guide	R	A	R	R
Final Demo	A	R	R	R
Preliminaries				
Setup Environment	C	R	A	C
Setup Framework	R	C	C	A
Proof of Concept				
Simulation Config	A	R	C	C
Build Agent	C	C	A	R
Integration	R	A	R	R
Test and Validate	R	R	R	A
Sprint 1[1]				
SENSORS Track[2]	R	R	C	C
MAPS Track[2]	C	C	R	R
Sprint 2				
Combine Systems[2]	R	R	R	R
Optimizations[2]	R	R	R	R

[1] Sprints will be segmented and designed in further detail in the corresponding report

[2] Different team members will lead separate subtasks specified in sprint planning.

6.2 | Gantt Chart



Terms

AV	Autonomous Vehicle
SOTA	State Of The Art
AD	Autonomous Drivers
SE	Simulated Environment
RL	Reinforcement Learning
LLM	Large Language Model

References

[1] S. Singh, "Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey," National Highway Traffic Safety Administration, Washington, DC, Mar. 2018.

[2] U.S. Environmental Protection Agency, "Fast Facts on Transportation Greenhouse Gas Emissions," 2024. [Online]. Available: https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions

[3] Natural Resources Canada, "Fuel-efficient driving techniques," 2024, [Online]. Available: https://natural-resources.canada.ca/energy-efficiency/transportation-alternative-fuels/personal-vehicles/fuel-efficient-driving-techniques/21038

[4] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 384–406, 2018, doi: https://doi.org/10.1016/j.trc.2018.02.012.

[5] I. Rehmanand, N. Mahabadi, T. Sanvictores, and C. I. Rehman, "Classical Conditioning," 2023, [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK470326/

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[7] A. Paszke *et al.*, "Automatic differentiation in PyTorch," 2017.

[8] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021, [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[9] Z. Yang, X. Jia, H. Li, and J. Yan, "LLM4Drive: A Survey of Large Language Models for Autonomous Driving." [Online]. Available: https://arxiv.org/abs/2311.01043

[10] M. Rowe, "CARLA CVPR Autonomous Grand Challenge winners announced," [Online]. Available: https://carla.org/2024/06/10/cvpr-winnders/

[11] Fortune Business Insights, "Autonomous Vehicle Market Size, Share & COVID-19 Impact Analysis," 2023. [Online]. Available: https://www.fortunebusinessinsights.com/autonomous-vehicle-market-109045

[12] Center for Sustainable Systems, University of Michigan, “Autonomous Vehicles Factsheet,” 2024.

[13] 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.

[14] T. Preston-Werner, “Semantic Versioning 2.0.0.” [Online]. Available: https://semver.org/

[15] Conventional Commits, “Conventional Commits Version 1.0.0.” [Online]. Available: https://www.conventionalcommits.org/en/v1.0.0/#specification