WASP Software Engineering Course

Module 2 Assignment

Deepthi Pathare

1 Introduction

My research focuses on decision making for semi and fully autonomous heavy duty vehicles such as trucks using machine learning methods. While the concept of autonomous driving spans across various vehicle types, including personal cars, buses, and even drones, there is only limited progress when it comes to heavy-duty vehicles. One of the critical challenges is enabling safe and efficient interactions, particularly in scenarios such as lane changing where cooperation from surrounding vehicles is essential due to the sheer size of the trucks. My research focuses on applying machine learning techniques - reinforcement learning in particular, to make safe and efficient driving decisions for trucks in highway environments. The main aspects of decision making include adaptive cruise control and lane changes with objectives to achieve a high level of safety and an optimal Total Cost of Operation (TCOP). TCOP mainly comprises of driver cost and energy cost. As we have complex and conflicting objectives, we approach the problem with techniques tailored for complex hierarchical/high dimensional optimization problems such as Curriculum Learning and Multi-Objective Reinforcement Learning.

2 Lecture principles

2.1 Software Engineering for ML

I have worked as a software developer for several years in industry before starting my PhD. So I am quiet familiar with various software engineering principles of requirements engineering, design, testing etc. However when I started my PhD in Machine Learning, I found it very challenging to follow these principles and build the project in a systematic approach. Project development wasn't always carried out through the pipelines of requirement engineering, design, or quality assurance. So it was quite interesting and thought provoking to hear about software engineering for ML during the lectures. I think it could be a bit hard to have an intact requirement specification when you do research because it is sometimes about trial and errors and ideas popping out in between. However I think there are many ideas from Quality Assurance in SE that we need to adopt. Generally, the only testing I perform in my ML projects are the final evaluation of my trained models which is mainly to evaluate and benchmark the performance of the models. This is not carried out using well crafted test cases which means we may fail to test certain cases. This can lead to serious problems especially in my research domain of autonomous driving. Another thing that I should adopt is unit testing. In general it is difficult to identify why a certain model is not working or behaving in a particular way. If we test each function, metrics etc separately in unit level, we might be able to reduce a lot of issues.

2.2 Behaviour Software Engineering

The concepts of behavioral software engineering (BSE) are highly relevant to my research on reinforcement learning for decision-making in heavy duty vehicles. Behaviour software engineering highlights how human cognitive biases, social influences, and group dynamics can affect decision processes and system outcomes. In my research context, understanding cognitive biases like confirmation bias can help you design safer algorithms by ensuring that decision models do not reinforce incorrect assumptions about safety or efficiency. Additionally, insights into team dynamics and organizational factors from BSE can inform collaborative development practices, ensuring clear communication and shared understanding among engineers working on safety-critical systems. Considering behavioral aspects can also improve the validation and acceptance of autonomous driving systems by addressing human factors such as trust, transparency, and safety perceptions, ultimately leading to more reliable and socially acceptable truck automation solutions. At the level of drivers and customers, we can leverage methods from BSE to enhance driver experience in saftey critical autonomous solutions

3 Guest-Lecture Principles

3.1 Stakeholders and Goal Modeling

The lecture discusses identifying stakeholders, their goals, and classifying those goals as usage, system, or business goals. In my research, major stakeholders include truck manufacturers, customers (who buy the truck), the truck driver and other road users. The goals of these stakeholders allow us to formalize conflicting objectives, such as safety versus cost, and develop reinforcement learning policies that can explicitly negotiate these trade-offs. For example, safety goals may be classified as system goals, while operational cost related objectives are business goals. Understanding the stakeholders' goals guides the formulation of reward functions in reinforcement learning, ensuring that the decision-making process aligns with real-world requirements and priorities. You can balance safety with operational costs by explicitly incorporating stakeholder preferences into the learning process. For example, in some countries energy costs will be very high compared to driver costs and hence customers might prefer the automated agent to make driving decisions such that energy efficiency is prioritized over travel time.

3.2 Requirements Engineering

My project involves multiple conflicting objectives, such as maximizing safety while minimizing energy costs and driver workload. Requirement engineering techniques like goal refinement and breaking down to concrete, measurable requirements may help to structure these trade-offs. Designing reward functions and shaping the rewards is a challenging task in Reinforcement Learning. Correctly identifying the measurable requirements and their priorities may also help to formulate informative reward functions, which can then improve reinforcement learning methods that balance competing goals effectively. Defining clear, measurable requirements also allows for systematic testing and validation of decision-making policies. For instance, it will be useful to verify whether the learned policies meet safety thresholds or cost targets, ensuring the models are reliable and trustworthy.

4 Data Scientists versus Software Engineers

- Do you agree on the essential differences between data scientists and software engineers put forward in these chapters? Why or why not?

  The chapters clearly define the differences between data scientists and software engineers, emphasizing their distinct focuses and skill sets. Data scientists are primarily engaged in developing and experimenting with machine learning models, focusing on data collection, feature engineering, and statistical analysis to improve model accuracy. In contrast, software engineers concentrate on building reliable, scalable, and maintainable systems that deploy and operationalize these models, handling aspects like infrastructure, automation, monitoring, and system integration. I agree with this distinction because it reflects the practical realities of deploying machine learning in production environments, where the needs for both model innovation and robust system engineering are essential but require different expertise. This separation helps clarify responsibilities but can also lead to friction if these roles are not well coordinated, highlighting the importance of collaboration across the two domains.

- Do you think these roles will evolve and specialize further or that "both sides" will need to learn many of the skills of "the other side" and that the roles somehow will merge? Explain your reasoning.

  Regarding their future evolution, I believe these roles will continue to develop toward greater overlap. The increasing adoption of MLOps, automation tools, and platform engineering encourages hybrid skill sets, making it easier for data scientists to understand deployment pipelines and for engineers to grasp core ML concepts. Organizationally, teams are moving toward more collaborative, cross-functional structures, which foster shared understanding and combined skill development. However, as systems and models grow more complex, specialized roles will still be necessary to ensure depth of expertise. Researchers will focus on innovation, while engineers optimize scaling and security. Thus, I see a future with both specialization and increased cross-training, where professionals are encouraged to learn skills from the other side to better collaborate and build integrated solutions.

5 Paper analysis

5.1 Paper 1: Rule-Based Assessment of Reinforcement Learning Practices Using Large Language Models

1. This paper [2] introduces a framework that combines rule-based methods with Large Language Models (LLMs) to automatically evaluate adherence to best practices in reinforcement learning (RL) pipelines. The authors propose a set of architectural rules targeting critical aspects like checkpointing, hyperparameter selection, and agent configurations. They then compare the effectiveness of LLM-assisted evaluations with traditional heuristic-based code analysis. By automating compliance checks, this approach aims to improve RL training efficiency, model robustness, and overall system performance, a growing need as RL applications become increasingly complex.

2. The relevance to my research is clear: I also work with RL for decision-making in autonomous heavy-duty vehicles. Ensuring that RL pipelines follow best practices can directly impact the safety, efficiency, and reliability of these systems.

3. An applied scenario could involve developing a highway autopilot for trucks, managing tasks like lane changes, collision avoidance, and adaptive cruise control. This would use machine learning techniques to optimize routing, performing adaptive cruise control, lane changes, and cooperative maneuvers, aiming to improve safety, reduce energy consumption, and minimize total cost of operation (TCOP). The rule-based, LLM-supported framework from the paper could be applied here to continuously monitor and validate the RL models used, ensuring proper hyperparameter tuning, model checkpointing, and adherence to safety constraints during both training and deployment. My ongoing work in RL for adaptive cruise control and lane-changing strategies fits naturally into such a project, combining safe, efficient decision-making with rigorous model validation.

4. From a practical perspective, incorporating the insights from this paper into my RL development workflow requires a significant shift from mostly manual, trial-and-error experimentation to a more systematic, automated approach. Currently, developing RL models for multi-objective goals—such as optimizing safety, energy efficiency, and total cost of operation in autonomous heavy-duty vehicles involve iteratively adjusting hyperparameters, testing different agent configurations, and monitoring checkpoints to ensure training stability. This process is time-consuming, resource-intensive, and prone to human error. Even small deviations in hyperparameters or overlooked checkpoints can lead to suboptimal performance or, worse, unsafe behaviors when models are deployed on highways. By implementing a modular, rule-based assessment framework, augmented with LLM-driven analysis as suggested in the paper, I can automate these validation steps and enforce adherence to best practices consistently across all projects.

   This approach provides multiple benefits. First, it introduces repeatability and standardization into the RL training pipeline. Rather than relying on individual intuition or experiments, each training run can be evaluated against a clearly defined set of rules covering checkpoints, hyperparameter selection, and agent configuration (which will come in future works involving multiple autonomous trucks). Second, it enhances safety and reliability. In my domain, even minor mistakes can propagate across agents in a fleet and compromise both performance and safety. Automated validation ensures that these risks are minimized and that the models adhere to safety-critical standards before deployment. Third, it accelerates development by reducing the manual effort required to validate complex RL setups. Time saved can instead be devoted to refining objectives, experimenting with novel multi-objective strategies, or integrating additional real-world constraints, all of which are crucial for optimizing the performance of autonomous fleet systems.

   Moreover, integrating LLM-based analysis into this framework offers additional advantages. LLMs can capture context-dependent compliance issues that are difficult to detect with heuristic-based methods alone. For instance, different combinations of hyperparameters, agent configurations, and training schedules that might lead to unstable learning can be identified automatically. In practice, implementing this integration would involve incorporating validation modules directly into the RL training pipeline. After each training iteration, the system would automatically check for adherence to best practices, flagging deviations for review. Hyperparameter tuning, which currently relies heavily on trial-and-error approaches, could be partially automated, with the system suggesting parameter adjustments based on observed performance and rule-based assessments. Checkpoints would be monitored not only for frequency but also for completeness and relevance to ongoing training objectives. This creates a feedback loop where validation results inform training adjustments in real time, leading to faster convergence and higher overall model quality.

5.2 Paper 2: Exploring Hyperparameter Usage and Tuning in Machine Learning Research

1. The paper [1] examines the widespread gap between the acknowledged importance of hyperparameter tuning in machine learning and its actual practice in research. The authors highlight how inconsistent tuning and poor documentation can compromise reproducibility and inflate claims of novelty. For AI systems, these findings highlight the need for structured, transparent experimentation and systematic hyperparameter optimization, which are essential for building robust, high-performing, and reliable models suitable for real-world deployment.

2. This resonates strongly with my research, particularly in safety-critical domains like autonomous heavy-duty vehicle control. Reinforcement learning and hierarchical optimization techniques, such as curriculum learning and multi-objective RL, are highly sensitive to hyperparameter choices.

The study reinforces the necessity of rigorous tuning, wise experimentation, and clear documentations to ensure reproducibility and robust model behavior.

3. The same application of highway autopilot for trucks can be considered here. The paper's recommendations can be applied here by implementing systematic hyperparameter tuning and transparent reporting using frameworks like Weights & Biases. These tools would track hyperparameter configurations, experimental results, and environment settings, improving model reliability, regulatory compliance, and iterative development efficiency.

4. To effectively implement the insights from this paper, I would integrate robust experimentation frameworks such as Weights & Biases directly into my RL development pipelines. These tools provide structured logging of hyperparameter configurations, experiment results, and environment settings, which is critical for ensuring reproducibility and transparency in research. Currently, when tuning RL models for tasks like adaptive cruise control, lane changes, or multi-objective optimization in autonomous heavy-duty vehicles, much of the process relies on manual trial-and-error. This approach is time-consuming, prone to oversight, and can lead to inconsistent performance across different runs or research projects. By leveraging structured experimentation frameworks, each training run would automatically capture critical details—hyperparameters, seed values, training metrics, and environmental conditions—allowing for a complete and systematic record of what was tested and why. This not only facilitates reproducibility but also enables more informed decisions when iterating on model architectures or training procedures.

   Complementing these frameworks with automated hyperparameter optimization techniques further enhances efficiency and reliability. For example, Bayesian optimization, Optuna, or Hyperopt can systematically explore large parameter spaces that would be infeasible to evaluate manually. This is particularly important in multi-objective RL scenarios, where small adjustments in learning rates, reward weighting, or exploration parameters can dramatically affect agent performance. Automating this exploration reduces human error, increases the likelihood of converging to near-optimal configurations, and allows researchers to focus on higher-level design decisions rather than painstaking manual tuning. In my experience, some hyperparameter choices can subtly destabilize training or bias an agent toward suboptimal behaviors; having an automated, systematic approach ensures these issues are detected and mitigated early in the development process.

   Transparent reporting is another crucial element. Documenting the rationale behind hyperparameter selections, experiment configurations, and observed outcomes ensures that findings are understandable and verifiable by peers. In safety-critical applications like autonomous trucking, this documentation also serves as a record for regulatory compliance and internal quality assurance. By combining rigorous logging, systematic optimization, and detailed reporting, the entire RL development workflow becomes more robust, reproducible, and auditable, addressing the key challenges highlighted in the paper.

   Finally, integrating these practices directly benefits my ongoing research. When developing highway autopilot strategies using RL and multi-objective optimization, I often work with conflicting goals such as safety, energy efficiency, and cost. Structured experiment tracking and automated hyperparameter optimization allow me to efficiently navigate trade-offs between these objectives, evaluate performance consistently, and iterate on models with confidence. Overall, adopting the paper's recommended practices transforms my RL pipelines into a more disciplined, transparent, and reproducible process, ultimately producing high-performing AI systems that are reliable, maintainable, and ready for deployment in real-world, safety-critical environments.

6 Research Ethics & Synthesis Reflection

1. I first scanned through the titles of papers accepted to CAIN in recent years and shortlisted those that appeared relevant to my research. I then read the abstracts and selected the two most suitable papers.

2. Some titles were overly broad or misleading. For example, “A Combinatorial Approach to Hyperparameter Optimization” does not indicate which type of machine learning methods it applies to. To address this, I skimmed the methods and results sections to determine whether the work is relevant to my research area and the methods I use.

3. I highlighted relevant information as I read the papers and then used them for my writing. LLMs were used for grammar corrections.

References

[1] Sebastian Simon et al. “Exploring Hyperparameter Usage and Tuning in Machine Learning Research”. In: *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*. 2023, pp. 68–79. DOI: 10.1109/CAIN58948.2023.00016.

[2] Evangelos Ntentos, Stephen John Warnett and Uwe Zdun. “Rule-Based Assessment of Reinforcement Learning Practices Using Large Language Models”. In: *2025 IEEE/ACM 4th International Conference on AI Engineering – Software Engineering for AI (CAIN)*. 2025, pp. 1–11. DOI: 10.1109/CAIN66642.2025.00009.