# LLM And RLA Integration In Autonomous Driving Through Bias Alteration

Jonathan Setiabudi, Ruoshen Mo
*University of California, Riverside*

***Abstract*—Reinforcement Learning Agents (RLAs) have shown strong capabilities in planning and control for autonomous driving. However, their data-driven nature limits their ability to generalize to rare and critical edge cases that are underrepresented in training data, often leading to unsafe or unpredictable behaviors. Recently, Large Language Models (LLMs) have demonstrated remarkable reasoning and zero-shot generalization abilities. In this work, we propose a novel framework that integrates LLMs with RLAs using bias alteration to enhance decision-making in unseen scenarios. Specifically, the ego vehicle's observations are textually summarized and provided to an LLM, which generates high-level driving decisions to guide the RLA's low-level control. Preliminary results indicate that the LLM-guided RLA achieves safer and more robust driving behavior compared to the standalone RLA.**

## I. Introduction

Reinforcement Learning (RL) systems are widely utilized for autonomous driving control due to their ability to learn optimal policies from environmental interaction and their capacity to adapt actions in complex, real-time scenarios. Nevertheless, a primary drawback is their lack of robustness in handling infrequent "long-tail scenarios" such as unexpected human or emergency vehicle interactions. This deficiency stems from the fact that RL is inherently dependent on the quality and quantity of the training data, leading to poor performance in rare but critical situations. Ensuring safety and reliability in these rare situations is crucial for real-world deployment. In particular, RLAs typically struggle in zero-shot and few-shot scenarios due to their lack of common-sense reasoning and prior knowledge about unseen environments [1].

LLMs have been explored extensively for various applications in autonomous driving, including perception, planning, and control. LLMs demonstrate significant promise across all these domains. Notably, LLMs can generate coherent plans and decisions without requiring additional specialized training, often by framing planning as a zero-shot task [2], [3]. Furthermore, LLMs have proven capable of perceiving and interpreting the environment through textual context, reinforcing the idea that little to no alteration of LLMs is needed for initial experimental integration [4]. It has also been demonstrated that LLMs excel at processing vast volumes of data and translating them into understandable, real-time feedback [5]. In this work, we propose a novel framework that integrates LLMs with RLAs to enhance decision-making in unseen scenarios. Drawing inspiration from [1], our architecture incorporates an LLM-augmented hierarchical RL framework, functionally arranged as a "cerebrum and cerebellum" hierarchy as shown in Fig. 1, where the LLM functions as the cerebrum—responsible for abstract reasoning and decision-making—while the RLA acts as the cerebellum, executing precise control actions. To demonstrate this concept, we employ a lightly trained Q-learning Driving (QLD) Agent as the base RLA, illustrating that the LLM can compensate for the agent's limited exposure to training data. Preliminary experiments show that the LLM-augmented RLA outperforms the standalone QLD Agent in most test scenarios, achieving longer driving durations and higher cumulative rewards.

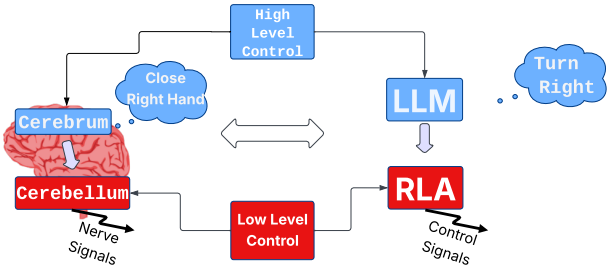


Fig. 1. Hierarchical Decision-Making Framework

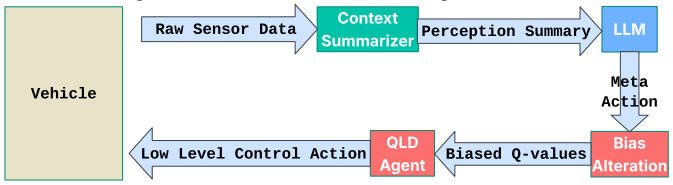


Fig. 2. System Architecture and Data Flow Overview

## II. Design

Our design integrates the LLM and the RLA through a three-step high-level guidance mechanism. As shown in Fig. 2, the process begins with a context summarizer that converts the ego vehicle's sensor and environmental data into a concise textual description for the LLM. Next, the LLM analyzes this textual summary, leveraging its extensive knowledge base and reasoning capability to generate singular, high-level meta-decision, *e.g.*, "turn left," "accelerate". Finally, this decision is subsequently translated into a range of acceptable low-level control signals, which are used to bias the RLA's Q-values toward actions aligned with the LLM's guidance. The RLA then samples its final control action from this biased action set.

**Context Summarizer** The context summarizer extracts the ego vehicle's observations and convert them into a two-part

textual representation. The first part is a detailed textual description, which includes a structured string detailing the ego vehicle's state in the format of: "Ego: pos $(x_{ego}, y_{ego})$, speed S $m/s$, heading angles, lane information and next way point $(x, y)$" and a list of descriptions of nearby vehicles such as "Vehicle 1: ahead at 12.4 $m$, speed 3.1 $m/s$." The second supplementary part provides high-level context by rephrasing key data points, labeling them as general concepts. For instance, the summarizer may send the LLM with phrase: "normal speed, need to turn left, traffic is busy."

**LLM Decision and Bias Alteration Mechanism** The prompt provided to the LLM contains the summarized information, the last command it issued, and a list of valid high-level commands, alongside general guidelines for selecting specific commands, such as: "If vehicles detected ahead and getting close: 'decelerate'." The resulting output command from the LLM is mapped to a dictionary containing a range of acceptable low-level control values.

The QLD Agent determines its next action by generating 50 candidate actions, which are sampled from its diffusion policy, each having an associated Q-value from the critic network. The bias alteration proceeds by iterating through these candidate actions: if an action's low-level control outputs *e.g.*, throttle, steering fall within the acceptable control range specified by the LLM's high-level meta-decision, the bias associated with that candidate action is increased. Critically, the magnitude of the bias increase is proportional to the number of control outputs that satisfy the range criteria; for example, an action with both throttle and steering within the acceptable range receives a higher bias increase than one with only a single output within range. The candidate action's final Q-value is then augmented by this accumulated bias, and the QLD Agent ultimately selects an action by sampling from the softmax distribution of these biased Q-values.

## III. Evaluation

To evaluate the efficacy of the proposed framework, we built a proof of concept system that Uses EasyCarlaRL [6], a lightweight openAI Gym environment built upon the CARLA [7] simulator. We used the pre-trained RLA agent and Gemini 2.0 [8] from EasyCarlaRL. Every 20 timesteps, Gemini is polled for a high-level metadecision. This was used to evaluate on six custom driving scenarios, which have not been previously encountered by the QLD Agent (dubbed by their respective names in Table I). The experimental results in Table I demonstrate that the LLM-augmented pipeline consistently achieves superior performance compared to the stand-alone QLD Agent, evidenced by both an increase in total time driven and the attainment of higher cumulative reward scores.

However, Alley Streets scenario presents a notable exception to this trend. We attribute this performance decline to the highly dynamic environment of the cramped alley streets and the fixed 20-timestep interval between LLM command updates. In scenarios with rapid state changes, this fixed interval introduces a critical lag; for example, if the LLM issues a command to turn left, that bias remains applied for the full 20 timesteps. If the turn is completed within 10 timesteps, the remaining 10 steps still apply a turning bias, which is no longer applicable and could potentially lead to unsafe actions.

TABLE I
Comparison of Agent Performance Across Custom Episodes

| Scenario | QLD Agent | | LLM+QLD Agent | |
|---|---|---|---|---|
| | **Reward** | **Steps** | **Reward** | **Steps** |
| Roundabout | 2072.16 | 500 | 2439.22 | 500 |
| Main Street | 1199.84 | 251 | 1418.26 | 249 |
| Residential Street | 379.66 | 103 | 628.15 | 136 |
| Tunnel Traversal | 2536.36 | 500 | 2650.55 | 500 |
| Intersection | 2648.54 | 409 | 2174.02 | 500 |
| Alley Streets | 895.18 | 415 | 1151.08 | 258 |

## IV. Conclusion

In this work, we presented a novel framework that integrates LLMs with RLAs, designed to mitigate the inherent generalization weakness of RLAs by leveraging the reasoning capability of a LLM. Preliminary experimental results demonstrate that the LLM-augmented pipeline consistently achieved superior performance and higher rewards compared to the standalone QLD Agent across unseen scenarios. Despite these promising results, our current design exhibits limitations in rapidly changing environments, primarily due to the use of fixed-interval LLM command updates. Future work will focus on mitigating this limitation through techniques such as command chaining and adaptive, state-aware query intervals, enabling more responsive and context-sensitive collaboration between the LLM and RLA.

## References


[1] L. Li, R. Tan, J. Fang, J. Xue, and C. Lv, "Llm-augmented hierarchical reinforcement learning for human-like decision-making of autonomous driving," *Expert Systems with Applications*, vol. 294, p. 128736, 2025.

[2] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," 2022.

[3] C. Cui, Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K.-D. Liao, T. Gao, E. Li, K. Tang, Z. Cao, T. Zhou, A. Liu, X. Yan, S. Mei, J. Cao, Z. Wang, and C. Zheng, "A survey on multimodal large language models for autonomous driving," 2023.

[4] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," 2022.

[5] C. Cui, Y. Ma, X. Cao, W. Ye, and Z. Wang, "Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles," 2023.

[6] "Easycarla-rl: A lightweight and beginner-friendly openai gym environment built on the carla simulator," https://github.com/silverwingsbot/EasyCarla-RL?tab=readme-ov-file.

[7] 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.

[8] Google, "Gemini 2.0 Flash Model Card and Documentation," https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash, Oct 2025.