

## Appendix

### Table of Contents

<b>A</b>	<b>LLM Evaluation: Driving Scenario Understanding and Response</b>	<b>2</b>
A.1	Evaluation Setup	2
A.1.1	Driving Conditions Specification	2
A.1.2	Tasks Definition	2
A.1.3	Evaluation Dimensions	3
A.2	Test Question Examples	3
A.3	Comparison Methods	3
A.4	Conclusions	4
<b>B</b>	<b>Additional Related Work</b>	<b>5</b>
<b>C</b>	<b>Methodology Details</b>	<b>6</b>
C.1	Augmented Semantic Representation Prompts	6
C.2	Contextual Stability Anchor	7
C.2.1	Prompts	7
C.2.2	Overview of Weight Distributions	7
<b>D</b>	<b>Experiments and Results</b>	<b>8</b>
D.1	Computational Resources	8
D.2	Experiments setup	8
D.3	HCRMP with Varied LLM	8
<b>E</b>	<b>Limitations and Broader Impacts</b>	<b>9</b>

## A LLM Evaluation: Driving Scenario Understanding and Response

### A.1 Evaluation Setup

#### A.1.1 Driving Conditions Specification

Driving conditions encompass both conventional scenarios—such as overtaking and merging—and safety-critical situations [1], as illustrated in Figure 1.

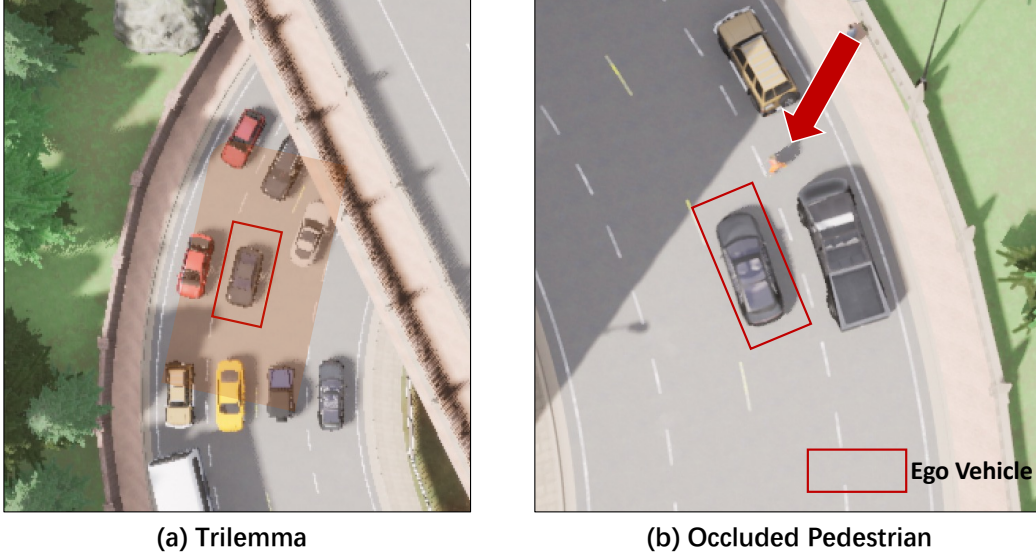


Figure 1: Safety-Critical Driving Conditions: Trilemma and Occluded Pedestrian

The **trilemma** driving condition refers to a situation where the ego vehicle (indicated by the red box) is surrounded by other vehicles on all sides. In this case, the ego vehicle must balance among three competing objectives: speed optimization, safe distance maintenance, and the need to change lanes. The **occluded pedestrian** driving condition is defined as a pedestrian suddenly stepping out from a blind spot and intersecting the path of the ego vehicle.

**The driving conditions used in this part of the experiment are consistent with those used in the other experiments in this paper.**

#### A.1.2 Task Definition

Each task systematically evaluates different capabilities of LLM in the context of autonomous driving(AD). We categorize these tasks into two main groups: Scenario Understanding and Action Response. The former focuses on the recognition and interpretation of driving conditions, while the latter assesses the model’s ability to respond and make decisions in various driving situations.

##### Scenario Understanding:

- **Hazard Identification:** the ability to accurately identify potential hazards in diverse driving conditions.
- **Road Type Comprehension:** the ability to correctly interpret and respond to different road types.
- **Road Sign Comprehension:** the ability to correctly interpret and respond to traffic signs.

##### Action Response:

- **Longitudinal Driving Decisions:** The ability to make appropriate decisions related to ego-vehicle speed.

- **Lateral Driving Decisions:** The ability to make appropriate decisions related to lane changes and steering.

### A.1.3 Evaluation Dimensions

The primary metric used is the non-hallucination rate, which quantifies the accuracy and factual correctness of the LLM outputs.

#### Non-Hallucination Rate

For each dimension, the LLM output is evaluated for the presence of hallucinations. A hallucination is defined as any generated content that is factually incorrect, inconsistent with the driving condition, or irrelevant to the query. The non-hallucination rate is calculated as:

$$\text{Non-Hallucination Rate} = \frac{\text{Number of Non-Hallucinated Outputs}}{\text{Total Number of Outputs}} \times 100\%$$

## A.2 Test Question Examples

We design a set of test questions to evaluate different LLM across the five defined tasks. Each test problem consists of a textual description of a driving condition, along with a specific query or instruction. The example problems are shown in Fig. 2 and aims to comprehensively evaluate the various capabilities of LLM in driving scenario understanding and response generation .

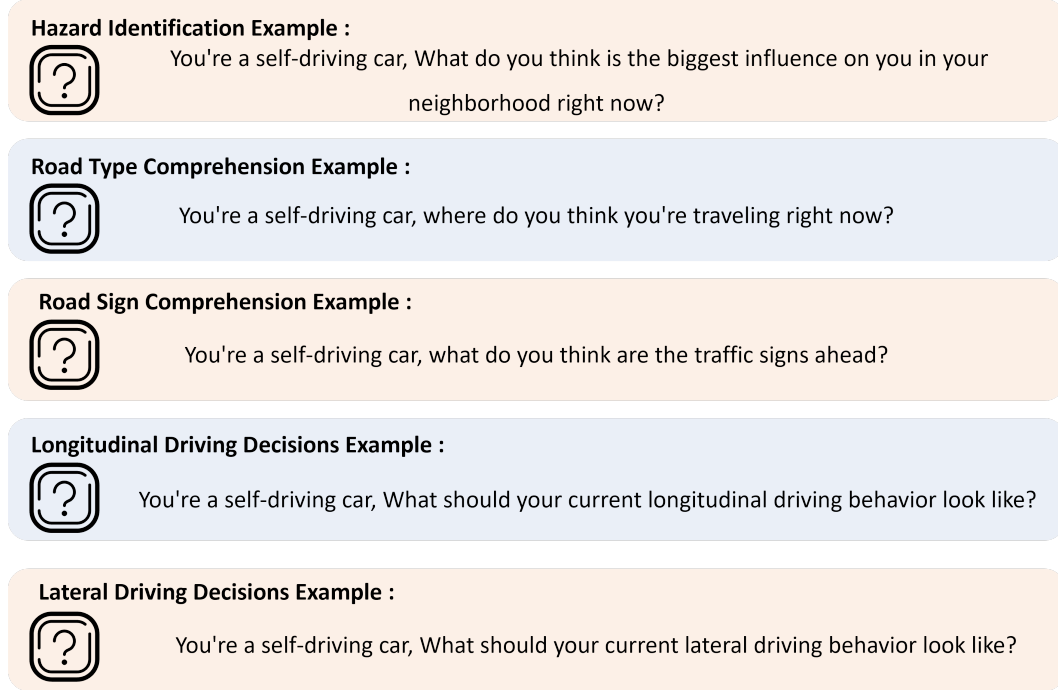


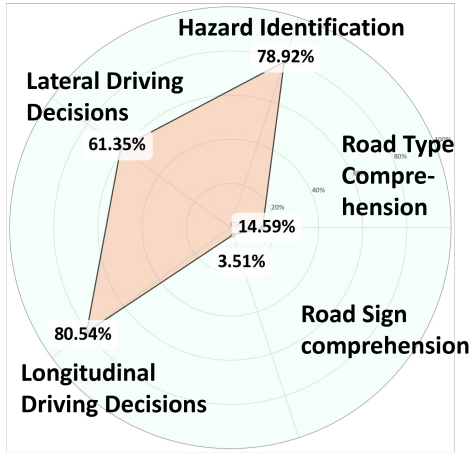
Figure 2: Sample test question for each task

## A.3 Comparison Methods

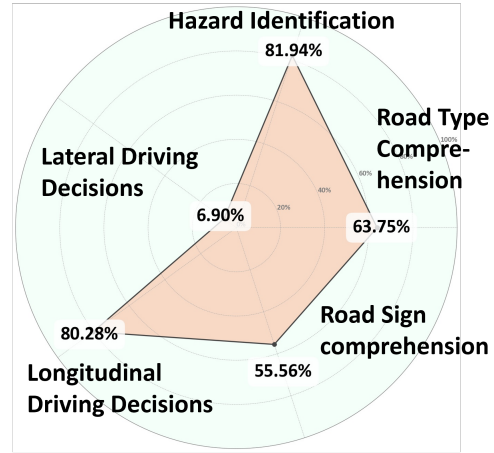
- **Gemini-2.5-Pro [2]:** On March 21, 2025, Google DeepMind officially launched Gemini-2.5-Pro , its latest flagship model. Hailed as Google's 'most intelligent AI model' to date, it marks revolutionary breakthroughs in reasoning, context understanding, and multimodal processing.

- **Gpt-4o** [3]: A language model released for ChatGPT, offers real-time reasoning across audio, visual, and text inputs. It supports 50 different languages with improved speed and quality.
- **Deepseek-r1** [4]: An AI model developed by Chinese artificial intelligence startup DeepSeek. Post-trained with reinforcement learning, it's designed to enhance reasoning capabilities, particularly excelling at complex tasks such as mathematics, code, and natural language reasoning.
- **Qwen-Turbo** [5]: Alibaba has launched a Qwen model on its Alibaba Cloud Bailian platform, featuring a significantly increased context length from 128k tokens to 1M tokens. This is equivalent to approximately one million English words or one and a half million Chinese characters.
- **Llama-3.3-70B-Instruct** [6]: Developed by Meta, it is the LLM with 70 billion parameters. It's specifically designed for multilingual dialogue scenarios and has been optimized through Supervised Fine-Tuning (SFT) and Reinforcement Learning from Human Feedback (RLHF), enabling it to excel at natural language processing tasks such as text generation.

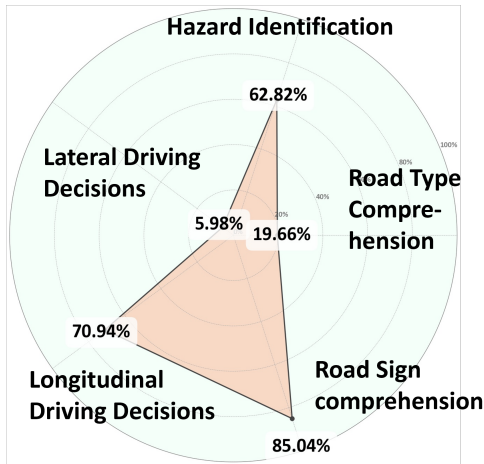
#### A.4 Conclusions



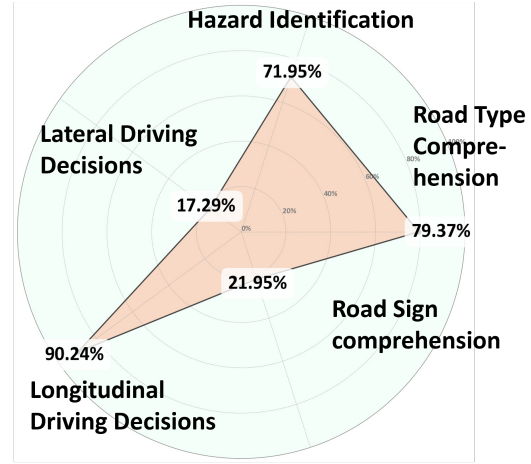
(a) GPT-4o



(b) Llama-3.3-70B-Instruct



(c) Deepseek-r1



(d) Qwen-Turbo

Figure 3: Non-Hallucination Rates of Different LLM Across Driving Tasks

Regarding the performance evaluation of LLM in autonomous driving tasks, results indicate that their performance is relatively strong in Longitudinal Driving Decisions and Hazard Identification, exhibiting higher non-hallucination rates. However, for tasks involving complex and precise operations such as Lateral Driving Decisions, the non-hallucination rates across all evaluated models are generally low, suggesting a significant hallucination rate in this area. Furthermore, apart from Deepseek-r1 which demonstrates prominent performance in Road Sign Comprehension, other models generally show deficiencies in this dimension. This highlights the inherent limitations of LLMs in accurate physical world perception and dynamic decision-making, which remain crucial challenges for their application in autonomous driving.

## B Addition Related Work

### Motion Planning in Autonomous Driving

The core task of motion planning in autonomous driving is to rapidly generate safe and robust local trajectories or motion commands that guide the vehicle to effectively avoid obstacles and operate smoothly in complex dynamic environments [7]. As a key component for achieving safe and efficient autonomous mobility, it has long been a highly active and closely studied area of research.

Current motion planning methods are mainly divided into two categories: pipeline planning and end-to-end planning. The traditional pipeline planning method, also known as the rule-based planning method, is composed of multiple interrelated modules. These modules—such as perception, localization, planning, and control—are designed and developed independently. Graph search planning algorithms determine a path between a start point and a goal point by performing iterative searches based on the environmental map and obstacle information. Common graph search algorithms include Dijkstra’s algorithm, the A\* algorithm, and the Hybrid A\* algorithm. While Dijkstra’s algorithm [8] can find the shortest path between two points, it lacks goal-directed efficiency and becomes computationally expensive in long-range searches. To address this limitation, Stanford University developed the A\* algorithm [9] in 1968, which significantly improves efficiency through the use of well-designed heuristic functions. However, A\* is mainly suited for static environments and does not adequately account for the motion constraints of moving vehicles. In 2008, Stanford University introduced the Hybrid A\* algorithm [10], which incorporates kinematic constraints, thereby enhancing the practicality and applicability of path planning in real-world driving scenarios. Esposto et al. [11] combined the Hybrid A\* and classical A\* algorithms to propose a path planning method based on Reeds-Shepp curves, which not only accommodates the kinematic characteristics of vehicles but also improves planning speed.

However, end-to-end planning methods have become a focal point of current research due to their superior adaptability and efficiency, leveraging artificial intelligence to directly map raw perception data to control commands. Representative approaches include: **Behavior Cloning**, which uses supervised learning to mimic expert trajectories and quickly generate reliable driving policies. **Reinforcement Learning**, which optimizes dynamic decision-making through interaction with the environment, enabling adaptation to complex scenarios.

Behavioral Cloning (BC) is a primary imitation learning approach in autonomous driving, where the agent learns to replicate expert behavior by training a classifier or regressor on demonstration trajectories. As a passive method, it assumes that state-action pairs in the demonstrations are independent and learns the target policy purely through observation of complete expert executions. Early BC applications in driving [12–14], used end-to-end neural networks to map camera inputs directly to control commands. To improve performance in complex urban settings, later work introduced enhancements such as multi-sensor inputs [15, 16], auxiliary learning tasks [17, 18], and more sophisticated expert demonstrations [19].

To reduce reliance on labeled data, some researchers have turned to reinforcement learning (RL) for autonomous decision-making. Unlike imitation learning, RL agents learn policies by interacting with the environment and maximizing cumulative rewards through trial and error. Over time, the agent refines its policy to achieve optimal performance based on feedback from the environment. RL has demonstrated success in learning lane following on a real vehicle in low-traffic conditions [20]. Saxena et al. [21] use the Proximal Policy Optimization (PPO) algorithm to learn a control policy in continuous motion planning. Their model implicitly accounts for interactions with surrounding vehicles to prevent collisions and improve ride comfort.

## C Methodology Details

### C.1 Augmented Semantic Representation Prompts

#### ASR Module Prompt: Global Scenario-Level Abstraction

You are an advanced AI assistant for an autonomous driving system. Your task is to analyze the provided driving condition data and extract key scenario-level semantic information.

**Given Scene Data:**

- Road topology features: {road\_topology\_description}
- Current traffic dynamics (overall flow, presence of traffic signals/signs): {traffic\_dynamics\_description}
- Ego-vehicle state (position, speed): {ego\_vehicle\_state}

**Instructions:** Based on the provided driving condition data, please provide a concise answer for each of the following aspects. This information will be used to generate a 4-dimensional semantic vector representing the global driving context.

1. **Road Category:** Classify the current road type. (Examples: Highway, Rural Lane, Urban Road)
2. **Traffic Density:** Describe the prevailing traffic density. (Examples: Low, Medium, High)

**Output Format:** For each numbered point above, provide a short descriptive phrase. Example:

1. Road Category: Urban Road 2. Traffic Density: Medium

#### ASR Module Prompt: Fine-Grained Object-Level Analysis

You are an advanced AI assistant for an autonomous driving system. Your task is to analyze the provided driving condition data, focusing on surrounding traffic participants and obstacles, to extract critical object-level semantic information.

**Given Scene Data:**

- Detected surrounding traffic participants (vehicles, pedestrians, cyclists): {list\_of\_detected\_participants\_with\_type\_position\_velocity}
- Detected static obstacles (roadblocks, debris): {detected\_static\_obstacles}
- Ego-vehicle state (position, speed, current lane, heading): {ego\_vehicle\_state}

**Instructions:** Based on the provided scene data, identify up to 3 of the most critical (highest risk or most influential on ego-vehicle's decisions) traffic participants or obstacles. For each identified critical entity, provide the following:

- **Entity Type:** (e.g., Car, Truck, Bus, Motorcycle, Pedestrian, Cyclist)
- **Relative Direction:** Its primary direction relative to the ego vehicle. Choose from: Front, Front-Left, Front-Right, Left, Right, Rear-Left, Rear, Rear-Right.

This information will be used to generate a 9-dimensional semantic vector representing the object-level context. Focus on conciseness and relevance for immediate driving decisions.

**Output Format:** List each critical entity as a separate item. Example: - Entity 1: Type: Car, Direction: Front-Left - Entity 2: Type: Pedestrian, Direction: Right - Entity 3: Type: Truck, Direction: Front

## C.2 Contextual Stability Anchor

### C.2.1 Prompts

#### Prompt for Autonomous Driving Attribute Weights Generation

**Background:** I need to use a three-element vector to represent the weighting characteristics of safety, comfort, and efficiency in autonomous driving. Please analyze the current driving scene based on the data read from the JSON file. The ego vehicle’s coordinates are: ( $\{ego\_position\_x\}$ ,  $\{ego\_position\_y\}$ ) The ego vehicle’s velocity is: ( $\{ego\_velocity\_x\}$ ,  $\{ego\_velocity\_y\}$ ) Surrounding environment information is:  $\{\text{surrounding\_info}\}$ ,  $\{\text{scene\_flag}\}$

**The question is:**  $\{\text{question}\}$ . Based on my requirements and  $\{\text{rag\_output}\}$ , please generate a three-element vector for me. The three elements have the following requirements:

- The three elements represent: Safety, Comfort, Efficiency.
- The sum of the three elements is 1.
- All three elements should be retained to 2 decimal places.

Please determine the specific values based on my requirements and only output this vector without any other redundant content.

### C.2.2 Overview of Weight Distributions

By anchoring the weight generation process to a norm-constrained semantic source, this module significantly mitigates the risk of weight drift. To empirically validate this, we conducted experiments within the same complex driving condition, specifically a road construction environment, comparing the LLM-generated critics weights with and without the integration of the Contextual Stability Anchor (CSA) module. As illustrated in Figure 4, this module successfully reduces the variance of LLM-generated critics weights by approximately **54.67%** within the same driving condition, thereby enhancing the consistency of multi-attribute coordination.

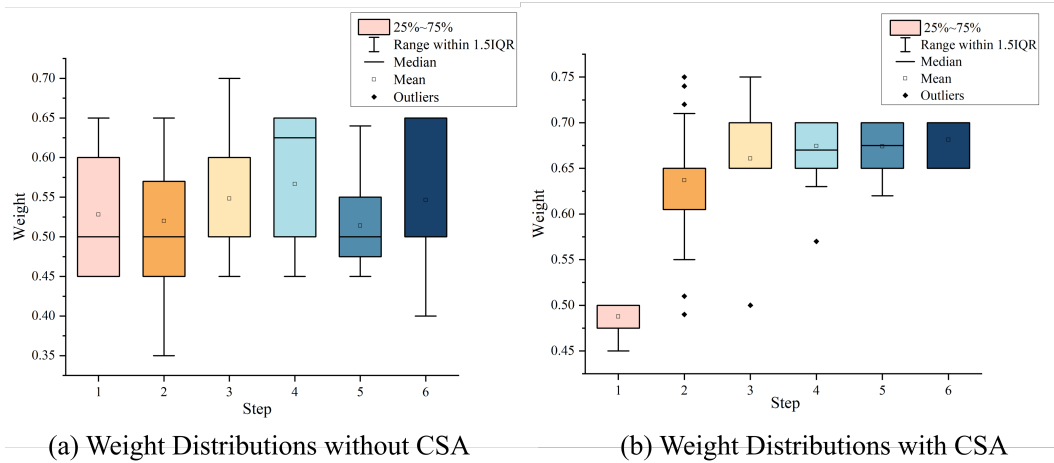


Figure 4: Comparison of LLM-Generated Weight Distributions With and Without CSA

Building upon the previous findings that demonstrated CSA’s ability to stabilize LLM-generated weights within a single driving condition, we further investigated its impact across diverse driving conditions. Figure 5 presents the Kernel Density Estimation (KDE) of the LLM’s output safety weights across various scenarios. In the left panel, representing the system without the CSA module, the boundary between safety weights for safe and dangerous scenarios is indistinct, indicating that the LLM struggles to consistently differentiate risk levels. Conversely, the right panel, which incorporates the CSA module, clearly illustrates a distinct separation in safety-critic weight distributions between

safe and dangerous driving conditions. This evident demarcation highlights CSA’s critical role in guiding the LLM to generate more context-aware and robust safety weight assignments, thereby enabling a clearer distinction between varying levels of risk and enhancing the reliability of safety-critical decision-making.

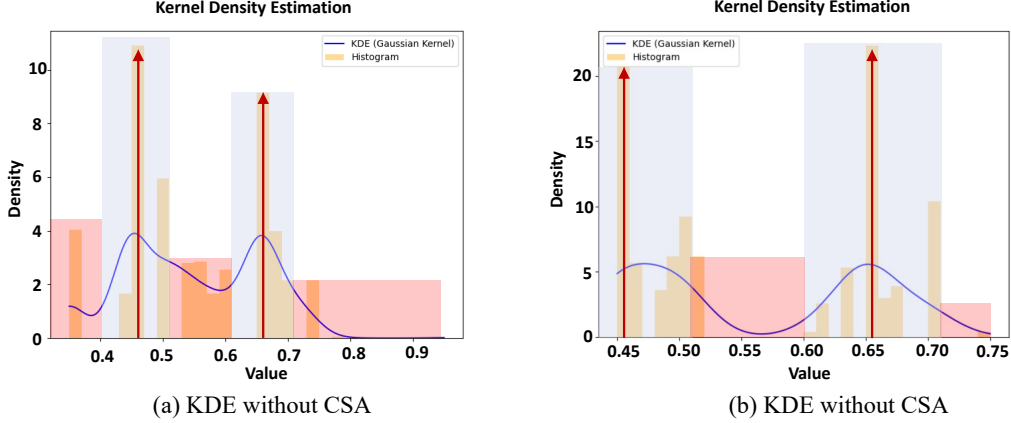


Figure 5: Kernel Density Estimation of LLM-Generated Safety-Critic Weights Across Diverse Driving Scenarios

## D Experiments and Results

### D.1 Computational Resources

The model training experiments in this study are accomplished relying on the following computational resources: a total of four NVIDIA A100-PCIE GPUs (40GB of graphics memory on a single card) are used. Each GPU is configured with 10 virtual CPU cores (Intel Xeon Gold 6248R) and 72GB of system memory. For each model, 200 epochs are performed, with each epoch consisting of 5 episodes.

### D.2 Experiments setup

Town02 provides a variety of challenging driving conditions for evaluating basic driving ability and complex decision-making actions, as it encompasses circular road networks, regular neighborhoods, and multi-level interchanges. We introduce varying traffic density settings as follows: in the on-ramp merging scenario, low, medium, and high densities correspond to 2, 4, and 8 surrounding vehicles, respectively, located on the two lanes adjacent to the merging lane on the main road. In the multi-lane overtaking scenario, these levels represent 1, 2, or 3 surrounding vehicles ahead of the ego vehicle.

For safety-critical trilemma and occluded pedestrian scenarios, the definitions for low, medium, and high traffic flow densities are unified. These are determined by counting the number of surrounding vehicles in the "borrowed lane" (i.e., the adjacent/oncoming lane that the autonomous vehicle temporarily occupies or interacts with for specific maneuvers). Specifically, low, medium, and high densities correspond to the presence of 2, 3, and 4 surrounding vehicles, respectively, in the aforementioned "borrowed lane".

These traffic density levels are designed to create diverse traffic environments and interaction situations, thereby increasing task complexity and enabling a more comprehensive evaluation of the algorithm’s decision-making ability and adaptability under different traffic conditions.

### D.3 HCRMP with Varied LLM

**Our study primarily conducts experimental validation based on the Gemini-2.5-Pro model.** To comprehensively assess the generalization capability and performance of the HCRMP architecture



across different LLMs, we further evaluated its effectiveness using four mainstream LLMs: GPT-4o, Llama-3.3-70B-Instruct, Deepseek-r1, and Qwen-Turbo in trilemma’s high-density driving condition.

Experimental results show that, as shown in figure 6 and table 1, the overall performance of HCRMP remains consistent across different LLM. Although these LLM exhibit some variation in hallucination-free rate evaluations, their differences have minimal impact when serving as semantic prompt sources for HCRMP. This demonstrates that HCRMP, as an LLM-Hinted RL paradigm, can effectively leverage the semantic prompting capabilities of various LLM. Through the self-learning mechanism of the RL agent, it compensates for disparities in LLM performance, thereby achieving similarly strong driving performance regardless of the underlying LLM backend.

Table 1: HCRMP with Varied LLM

	SR(%)	CR(%)	AS(m/s)	TD(m)	TS(s)	SV(m/s)	AV(m/s^2)
HCRMP-Gemini-2.5-Pro	<b>64</b>	<b>36</b>	<b>9.94</b>	77.34	47.13	9.96	<b>1.72</b>
HCRMP-GPT-4o	61	39	6.85	53.09	52.88	2.4	1.84
HCRMP-Deepseek-r1	61	39	8.26	<b>87.35</b>	51.44	8.46	1.97
HCRMP-Qwen-Turbo	59	41	7.02	65.04	<b>56.84</b>	<b>2.17</b>	1.79
HCRMP-Llama-3.3-70B-Instruct	63	37	6.84	47.68	51.7	2.26	1.79

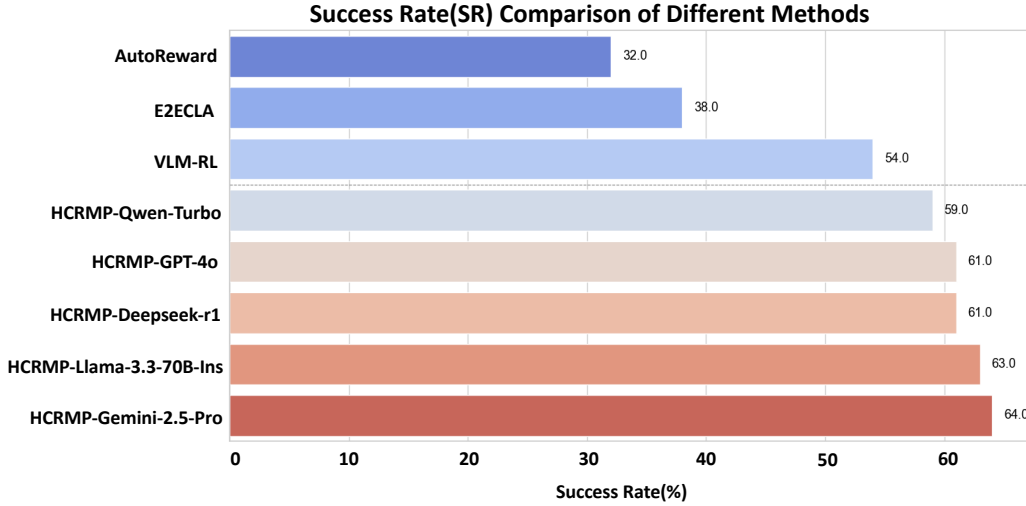


Figure 6: Success Rate (SR) Comparison of Different Methods

## E Limitations and Broader Impacts

While HCRMP demonstrates promising motion planning capabilities, its current limitations primarily involve potential inter-module information loss and the sim-to-real gap inherent in simulator-based evaluations, defining clear avenues for future work. Future work will enhance data fusion and module integration to reduce information loss, and validate HCRMP in real-world tests. This will strengthen HCRMP, an LLM-hinted RL paradigm that promises to significantly advance intelligent transportation.

## References

- [1] W Sing, C Xu, H Lin, B Li, et al. A survey on safety-critical scenario generation for autonomous driving—a methodological perspective. *arXiv preprint arXiv:2202.02215*, 2022.
- [2] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

- [3] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [4] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, and et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.
- [5] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025.
- [6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, and et al. The llama 3 herd of models, 2024.
- [7] Siyu Teng, Xuemin Hu, Peng Deng, Bai Li, Yuchen Li, Yunfeng Ai, Dongsheng Yang, Lingxi Li, Zhe Xuanyuan, Fenghua Zhu, et al. Motion planning for autonomous driving: The state of the art and future perspectives. *IEEE Transactions on Intelligent Vehicles*, 8(6):3692–3711, 2023.
- [8] Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: his life, work, and legacy*, pages 287–290. 2022.
- [9] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [10] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. Practical search techniques in path planning for autonomous driving. *ann arbor*, 1001(48105):18–80, 2008.
- [11] Francesco Esposto, Jorrit Goos, Arjan Teerhuis, and Mohsen Alirezaei. Hybrid path planning for non-holonomic autonomous vehicles: An experimental evaluation. In *2017 5th IEEE international conference on models and technologies for intelligent transportation systems (MT-ITS)*, pages 25–30. IEEE, 2017.
- [12] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.
- [13] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [14] Y Lecun, E Cosatto, J Ben, U Muller, and B Flepp. Dave: Autonomous off-road vehicle control using end-to-end learning. *DARPA-IPTO Final Report*, 36, 2004.
- [15] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17222–17231, 2022.
- [16] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7077–7087, 2021.
- [17] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9329–9338, 2019.
- [18] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

- [19] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232, 2021.
- [20] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day. In *2019 international conference on robotics and automation (ICRA)*, pages 8248–8254. IEEE, 2019.
- [21] Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. Driving in dense traffic with model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5385–5392. IEEE, 2020.