

VisuaLLMPlanner - A Maneuver Planner for Automated Vehicles using Large Language Models

Daniel Neurath¹, Bernd Schaeufele², and Ilja Radusch³

Abstract—Achieving safe and reliable automated driving in real-world conditions requires the ability to handle rare and unpredictable situations, commonly known as long-tail scenarios. These cases are often underrepresented in training data and remain a major challenge for conventional motion planning systems. In this work, we present VisuaLLMPlanner, a maneuver planning framework that integrates a multimodal large language model (MLLM) into the high-level decision-making loop of an automated driving pipeline. The system is triggered when the ego vehicle encounters a situation with an obstacle that cannot be resolved by a standard lane-following planner. At this point, a structured input comprising a bird’s-eye view image and a textual scene description is generated and passed to the MLLM. Rather than generating plans directly, the model selects from a discrete set of pre-generated and validated maneuver options, allowing for interpretable and structured decision-making. We evaluate our approach on the interPlan benchmark, which focuses explicitly on long-tail scenarios, and demonstrate that VisuaLLMPlanner achieves strong performance in comparison to prior LLM-based planners. The results highlight both the potential and current limitations of foundation models for high-level reasoning in automated vehicle planning.

I. INTRODUCTION

Automated vehicles promise significant benefits in road safety, passenger comfort, accessibility, and economic efficiency [1], [2]. In recent years, data-driven approaches have enabled substantial progress toward scalable automated vehicle deployment [3]. Yet, one of the most persistent challenges is the reliable handling of long-tail scenarios, which refer to rare or complex situations that deviate from standard driving patterns and are underrepresented in large-scale datasets [4]. These scenarios include edge cases such as stalled vehicles, temporary construction zones, or occluded obstacles, which often cause traditional rule-based or learning-based planners to fail. Successfully navigating these sorts of scenarios requires a high degree of contextual, semantic, and intent-based reasoning that conventional methods continue to struggle with [4], [5].

In parallel, recent advances in large language models (LLMs) demonstrate remarkable generalization and reasoning capabilities across a broad range of domains [6]. Their potential to perform abstract, high-level reasoning makes them compelling candidates for enhancing automated vehicle

motion planning, particularly in long-tail settings. With the emergence of multimodal models like GPT-4 [7], which can jointly process visual and textual input, new opportunities arise for integrating LLMs into motion planning pipelines. A number of recent studies have explored this direction by using LLMs to generate trajectories [4], [8], selecting abstract driving behaviors [4], [9], or interpreting driving scenes [10], [11]. Yet, most existing implementations either rely on natural language inputs with limited spatial fidelity [4], [8], or they delegate low-level behavior selection to the LLM without providing explicit, validated maneuver options for reasoning [4], [10].

In this paper, we propose VisuaLLMPlanner, a novel maneuver planning system that incorporates a multimodal LLM as a high-level decision module within an automated vehicle control stack. Our approach is centered around *scene-triggered* queries: when an obstacle is detected that cannot be resolved by a standard lane-following planner, the system generates a bird’s-eye view image and a natural language description of the scene. The LLM is then tasked with selecting from a discrete set of validated trajectory options. Unlike prior work, our method queries the model only after these alternatives have been generated. This allows the model to reason about concrete, feasible actions rather than relying on abstract inference alone.

There are three core contributions of this work. The first one is a hybrid planning architecture that isolates and evaluates LLM-based decision-making in long-tail traffic scenarios. The next core contribution is a dual-modality input format combining bird’s-eye view visualization and structured scene descriptions, and finally we provide an evaluation of the approach on the interPlan [4] benchmark, demonstrating its ability to make robust, explainable decisions in safety-critical settings.

II. RELATED WORK

While early work in automated driving aims at obstacle detection and recognition of the drivable area in front of the vehicle [12], the focus more and more shifts to understanding longer scenarios and more complex situations. While communication and cooperation between vehicles can help to convey the planned behavior of vehicles [13], not all vehicles are necessarily equipped with this technology. Hence for motion planning, scene understanding is an essential task. Recent work has begun to explore the integration of LLMs into motion planning systems, particularly for handling complex long-tail scenarios.

¹Daniel Neurath is with Technical University Berlin, Berlin, Germany daniel.neurath@alumni.tu-berlin.de

²Bernd Schaeufele is with Fraunhofer Institute for Open Communication Systems (FOKUS) Berlin, Germany bernd.schaeufele@fokus.fraunhofer.de

³Ilja Radusch is with Daimler Center for Automotive Information Technology Innovations (DCAITI), Berlin, Germany ilja.radusch@dcaiti.com

A. Input Representation

One important aspect of this research concerns how the driving scene is represented as input to the model. Early approaches often rely on natural language prompts that describe the positions and states of traffic participants relative to the ego vehicle, as seen in GPT-Driver [8]. While this approach offers human-readable interpretability, it often lacks the spatial precision required for robust planning. Vector-based encodings offer an alternative, representing scenes in structured numerical formats [14]. However, such methods typically require significant fine-tuning to align these representations with the LLM’s pretrained modality, which can undermine the intended zero-shot generalization capabilities.

The emergence of multimodal models such as GPT-4 [7] enables direct image-based inputs. This has led to growing interest in *bird’s-eye view* representations, as used in Talk2BEV [11] and PlanAgent [10]. Bird’s-eye view visualizations provide a spatially grounded scene format and can help the model infer traffic dynamics visually. However, challenges remain: for instance, Talk2BEV reports that LLMs still struggle with direct spatial reasoning, often leading to large errors [11]. In this paper, we adopt a similar image-based input strategy but enhance it by providing the model with filtered natural language descriptions of key objects to improve spatial reasoning reliability.

Another key distinction in our method lies in the timing and role of the LLM query. In contrast to prior approaches where the LLM selects meta-actions without knowledge of the exact behavior implementation (e.g., HybridLLMPlanner [4], PlanAgent [10]), our approach queries the model only after generating and validating specific trajectory options. This changes the model’s role from a general behavior selector to a trajectory evaluator, enabling it to reason concretely about proposed actions. We believe this aligns more naturally with the strengths of current LLMs, which are better suited for comparative reasoning than raw spatial estimation.

B. Output Representation

With respect to output representation, literature shows three primary formats: (1) high-level meta-actions, (2) low-level control commands, and (3) full trajectory generation. Meta-actions, as used in DiLu [9] and HybridLLMPlanner [4], maintain a clear separation of concerns and allow the LLM to operate at an abstract level. Trajectory generation, as attempted in LLMWaypointsPlanner [4], has shown limited success even after fine-tuning, often producing trivial outputs with little directional variance. A hybrid approach is taken by PlanAgent [10], where the LLM sets high-level parameters for the base planner, but its overall contribution is difficult to isolate. In our work, we adopt the meta-action approach with a minimal and interpretable set: LEFT, RIGHT, CAREFUL, and NORMAL, selected to support direct comparison with existing planners.

While several papers pursue fine-tuning to adapt LLMs to driving contexts [4], [8], [14], we avoid it in this paper to preserve the model’s general-purpose reasoning capabilities. Fine-tuning risks overfitting to specific scenarios and

undermines the validity of zero-shot evaluation. Instead, we employ prompt engineering techniques, including chain-of-thought reasoning and few-shot examples, which have been shown to effectively guide task performance without altering model weights.

C. Benchmark Evaluation

Finally, benchmark evaluation remains a critical challenge. Most published results rely on non-standard benchmarks, making comparisons difficult. The interPlan benchmark [4], used in this work, offers a realistic and diverse testbed focused explicitly on long-tail scenarios. Of the planners evaluated on interPlan, HybridLLMPlanner [4] is most relevant, as it also uses an LLM for high-level decision-making. However, this approach makes usage of an unmodified controller based on Intelligent Driver Model (IDM). Thereby, it does not give an answer to the question of how much of the benchmark success stems from the LLM versus the base planner. To address this, we carefully restrict our base planner and introduce fixed-decision baselines to isolate and quantify the contribution of the LLM.

Other evaluations, such as those in PlanAgent [10] on the Val14 and Test14-hard benchmarks [15], highlight the difficulty of attributing success to the LLM when the base planner alone performs nearly as well. Our implementation seeks to avoid this ambiguity by explicitly configuring the base planner to remain passive unless instructed otherwise, ensuring that any maneuver around long-tail obstacles is initiated as a direct result of the model’s reasoning.

III. METHODOLOGY

The planner is designed as a three-phase system. In the initial phase, the vehicle relies on a deterministic, rule-based motion planner, namely Predictive Driver Model (PDM)-Closed [15], which is responsible for managing routine lane-following behavior. As the ego vehicle approaches an obstacle that would cause a full stop in the current lane, the system enters the query phase. At this point, scene information is gathered and encoded into a multimodal input consisting of a bird’s-eye view image and a textual description. This input is passed to the MLLM, which reasons about the scene and selects a high-level meta-action. The planner then transitions into the action phase, where the selected maneuver is implemented through trajectory tracking. The architecture of our system is shown in Fig. 1.

A. Initial Lane Following

During the initial phase, the rule-based PDM-Closed planner governs the vehicle’s behavior by maintaining lane centering, enforcing speed limits, and regulating longitudinal motion based on headway to leading agents through an IDM policy. This setup is validated in prior work, where PDM-Closed demonstrates strong performance in routine lane-following scenarios [4], making it a suitable and reliable baseline for free-flow traffic conditions. Crucially, during this phase, no high-level decisions are made; the system operates

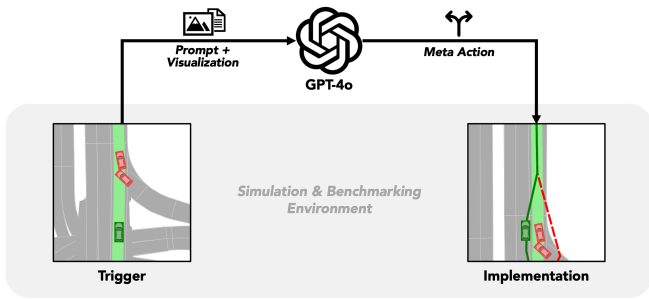


Fig. 1: VisuaLLMPlanner architecture.

under the assumption that traffic is unimpeded and that no intervention from the language model is required.

The transition to the query phase is triggered when the ego vehicle approaches an obstacle that would cause a full stop under the strictly lane-following planner. We deliberately select this condition as the trigger event, as it marks the point where the baseline planner is no longer capable of resolving the scenario independently and must defer to the high-level reasoning of the language model. This choice is crucial, as it ensures that any successful resolution of the situation can be directly attributed to the LLM’s decision-making process later on. The types of situations captured by this trigger typically correspond to long-tail scenarios, such as stalled vehicles, temporary construction zones, or improperly parked objects that obstruct the driving lane. To detect these cases, we perform a geometric interference test using a buffered polygon aligned with the centerline of the ego lane. The length of this polygon is dynamically adjusted based on the current speed of the ego vehicle, allowing for earlier detection during high-speed approaches and providing sufficient time to evaluate alternative actions before reaching the obstacle.

B. Query-Phase

Once this trigger condition is met, the system assembles a comprehensive snapshot of the local traffic environment. This snapshot comprises two modalities: a bird’s-eye view image and a structured textual prompt.

1) *Bird’s-Eye View Visualization*: The bird’s-eye view image captures a 25×25 meter region centered on the ego vehicle, encoding semantic entities such as other vehicles, pedestrians, and traffic infrastructure using color-coded bounding boxes. Agent velocities are represented as directional arrows, with lengths scaled according to speed, allowing the model to visually estimate motion and interaction potential. The image is rotated to ensure that the ego vehicle always points upward, standardizing spatial orientation across scenes. In addition to the static environment, we render the feasible overtaking trajectories directly into the image at the time of the query, enabling the model to visually assess and reason about possible maneuvers relative to the surrounding scene. Semantic information encoded through consistent color schemes is more reliably detected and referenced by the MLLM in its reasoning steps, while spatial relationships

involving motion vectors or overlapping of planned trajectories with other objects or road boundaries are more likely to be misinterpreted or ignored.

2) *Textual Scene Description*: Alongside the bird’s-eye view image, a structured textual scene description is generated to complement the visual input and reinforce critical context. This description captures the ego vehicle’s current speed and position, the surrounding lane configuration, and the state of nearby traffic participants. For each detected agent, the prompt provides a natural language summary of its relative position, velocity, and direction of travel using qualitative terms (e.g., “approaching from the front-left” or “moving in the opposite direction”), rather than numerical coordinates. This abstraction helps the MLLM form a high-level understanding of the scene more akin to human reasoning.

The description is generated programmatically from the simulator state at the moment of triggering and expressed in an ego-centric reference frame aligned with the BEV visualization. Only agents considered relevant for the current decision are included, based on proximity and short-horizon motion extrapolation to account for potential future interactions. In qualitative analysis of the model outputs, we frequently observe that the MLLM explicitly reuses the structured blockage and agent descriptions in its reasoning step, suggesting that the textual summary serves as a stabilizing semantic scaffold.

Encoding relevant details through concise natural language significantly improves the model’s ability to interpret constraints and reference key entities in its reasoning. Combined with the bird’s-eye view visualization, this dual-modality input provides both symbolic and spatial structure, enabling more robust decision-making in complex, dynamic scenes. In a real-world application, the scene descriptions could be derived from sensor detections, internal vehicle busses, and map information.

3) *Prompt Generation*: The prompting strategy used to elicit decisions from the MLLM incorporates several well-established techniques. Prompts are structured to follow a chain-of-thought format, encouraging the model to reason sequentially through three stages: (1) describing the scene, (2) assessing the feasibility of available maneuvers, and (3) evaluating the nature and urgency of the blockage. The expected output is returned in a structured JSON format, including both the chosen meta-action and the rationale behind it. A single few-shot example is included in each prompt to demonstrate the expected format and to improve performance through implicit task conditioning. To further reduce variance across repeated evaluations, the model is queried with a low temperature setting of 0.05. This approach was empirically found to minimize stochasticity while preserving reasoning quality. All experiments are conducted using OpenAI’s GPT-4o [7], accessed via the public API.

4) *MLLM Decision Making*: To properly evaluate the reasoning capabilities of the language model, it is essential that all high-level decision options are available at the time of query. This includes not only active maneuvers such as

overtaking, but also the ability to remain in the current lane and come to a full stop in front of the obstacle when necessary. Providing this full action space ensures that the model is not compelled to select an unnecessary or unsafe maneuver and can instead choose passive behavior when it is the more appropriate and safer option.

The model first determines, if the obstacle should ultimately be overtaken or if it should instead be handled by bringing the vehicle to a full stop temporarily. Based on the combined visual and textual input, the MLLM selects one of four discrete meta-actions: NORMAL, CAREFUL, LEFT, or RIGHT. Choosing NORMAL means the situation remains under control of the single lane-following base planner, under the assumption that the obstacle will clear on its own. This may be appropriate in situations where the detected blockage is temporary in nature, such as a vehicle waiting at a red traffic light or momentarily paused at an intersection. The CAREFUL action keeps the vehicle in-lane while reducing speed and increasing headway, allowing it to approach the obstacle cautiously without immediately committing to a maneuver. In this state, the MLLM is re-queried either periodically (every 3 seconds) or when the scene changes significantly, such as when new overtaking options become available, or the nature of the obstacle evolves. The LEFT and RIGHT actions initiate overtaking maneuvers on the respective side, provided that the model has determined such a maneuver to be both feasible and necessary given the scene context. This behavior is modeled as a state machine that can be seen in Fig. 2.

5) *Trajectory Generation*: To support such maneuvers, we pre-generate overtaking trajectories on both the left and right side of the obstacle cluster. These trajectories are computed using a convex hull expansion around the detected obstacle, followed by linear interpolation to ensure a smooth and drivable path. Heading angles are assigned to each waypoint in order to support stable trajectory tracking. The generated trajectories are subjected to multiple feasibility checks: they must stay within the drivable area, maintain a safe margin from other road users, and avoid collision with moving agents based on predicted future positions derived from their current velocity vectors. We included these feasibility checks based on empirical observations during development, where

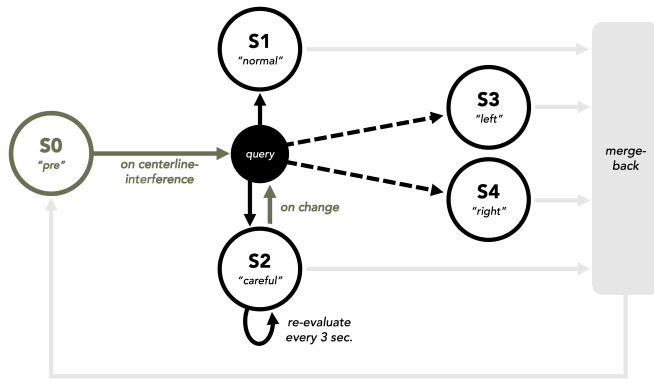


Fig. 2: State machine for decision making.

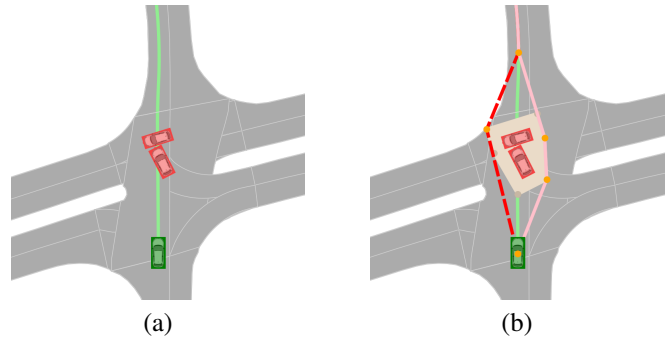


Fig. 3: Example overtaking trajectory generation. (a) Process is triggered due to current lane being blocked by two stationary vehicles. (b) Trajectories to both sides are generated. LEFT (dashed red) is deemed unfeasible as it violates drivable area compliance.

we found that the MLLM frequently struggled to reliably detect spatial relations such as drivable area violations or collisions. In Fig. 3, an example overtaking scenario is shown.

C. Action Phase

The selected meta-action is executed through the PDM-Closed planner, which serves as the low-level controller for tracking the trajectory. In order to isolate the MLLM’s role in decision-making, the planner is modified to prevent lateral deviations unless explicitly instructed by the high-level policy. This ensures that the base planner does not independently resolve complex scenarios and that successful navigation through long-tail cases can be directly attributed to the reasoning performed by the language model. Additionally, conservative acceleration and deceleration parameters are applied to guarantee safe stopping behavior and eliminate unintended progression past static obstacles.

IV. RESULTS

To evaluate the performance of the proposed planner, we benchmark VisualLLMPlanner on the interPlan dataset [4], which focuses on rare and complex long-tail scenarios. Since the LLM introduces some degree of non-determinism, each experiment is conducted three times using consistent prompts and model parameters. Table I shows the scores per scenario category for all three runs.

TABLE I: Results of VisualLLMPlanner across three runs on the interPlan benchmark for the categories Construction, Accident, Jaywalker, Nudge, and Overtaking. Distinction of lane-change categories, i.e., Low Traffic Density (LTD), Medium Traffic Density (MTD), High Traffic Density (HTD) are outside the planner’s design scope.

Method	Total	Constr.	Acc.	Jayw.	Nudge	Overt.	L	M	H
Run 1	74	83	66	89	88	81	61	61	61
Run 2	74	83	66	89	88	81	61	61	61
Run 3	73	83	66	81	88	81	61	61	61
Avg.	73	83	66	86	88	81	61	61	61

TABLE II: Performance comparison with other published planners and fixed decision heuristics on interPlan.

Method	interPlan	Constr.	Acc.	Jayw.	Nudge.	Overt.	<i>L</i>	<i>M</i>	<i>H</i>
LLMWaypointsPlanner [4]	22	0	0	0	0	0	64	41	69
PDM-Closed [4]	42	18	0	48	74	9	62	62	62
HybridLLMPlanner [4]	53	27	20	48	93	28	81	48	80
Fixed NORMAL	31	0	0	65	0	0	61	61	62
Fixed CAREFUL	35	0	0	89	7	0	61	61	61
Heuristic LEFT/RIGHT	65	83	66	19	88	83	61	61	61
VisuaLLMPlanner (Ours)	73	83	66	86	88	81	<i>61</i>	<i>61</i>	<i>61</i>

The results are consistent across runs, with only a single scenario in the Jaywalker category showing variance. This specific deviation is discussed further in the qualitative analysis (Section V).

A. Comparison with Other Planners

To contextualize these results, we compare our average performance with several reference methods (Table II). These include published LLM-based planners such as HybridLLMPlanner [4], LLMWaypointsPlanner [4], and baselines such as the default PDM-Closed planner [15].

VisuaLLMPlanner consistently outperforms the other planners in obstacle-based categories and particularly excels in the Jaywalker category, where caution is essential. While HybridLLMPlanner achieves slightly higher scores in Nudge scenarios, the difference is marginal. Notably, fixed decision baselines such as always choosing NORMAL or CAREFUL perform poorly across most categories, demonstrating the need for scene-adaptive reasoning.

B. Areas of Failure

To better understand where performance losses occur, we analyze the binary evaluation metrics used in the interPlan benchmark [16], which is derived from the nuPlan benchmark [17]. The metric *drivable_area_compliance* counts the cases, in which the ego vehicle remains within the designated drivable area. With *ego_is_making_progress*, it is evaluated, whether the vehicle is able to continue moving forward. The ability of the ego vehicle to bypass obstacles is captured by *ego_sorts_[obstacle]*. Finally, *no_ego_at_fault_collisions* quantifies the cases in which the ego vehicle successfully avoids collisions, excluding those caused by other agents. Table III summarizes the success rates across the different scenario types.

TABLE III: Binary metrics (pass rates in %) of VisuaLLMPlanner on interPlan.

Metric	Total	Constr.	Acc.	Jayw.	Nudge.	Overt.	<i>L/M/H</i>
<i>drivable_area_compliance</i>	99	100	90	100	100	100	100
<i>ego_is_making_progress</i>	96	100	100	100	100	100	90
<i>ego_sorts_[obstacle]</i>	96	100	80	100	100	100	–
<i>no_ego_at_fault_collisions</i>	100	100	100	100	100	100	100

Failures are limited to three main areas:

- *Drivable Area Compliance* fails in one accident-site scenario, caused by insufficient spatial understanding of the drivable region, by the MLLM.

- *Ego Is Making Progress* fails in three lane-change scenarios, which the planner is not designed to solve.
- *Ego Sorts Obstacle* fails in two accident scenarios where no feasible trajectory could be generated.

These shortcomings arise primarily from inherent limitations of the base planner or from scenario constraints that lie beyond the intended design scope of the model, rather than from suboptimal decisions made by the LLM.

C. Summary

VisuaLLMPlanner achieves robust performance in benchmark scenarios requiring careful reasoning and maneuver selection. It consistently outperforms both published and heuristic baseline methods in most categories. The benchmark results show that the LLM’s reasoning contributes meaningfully to performance, particularly in complex, dynamic scenarios.

V. EVALUATION

A. LLM Hallucination in High-Risk Scenarios

A notable failure mode observed in our evaluation involves the LLM generating reasoning that is inconsistent with the input scene or internally contradictory. This type of behavior is typically referred to as *hallucination*. As shown in Fig. 4, the model selects the meta-action RIGHT, recommending an overtaking maneuver despite the presence of multiple pedestrians crossing the road in the proposed path.

We distinguish between two primary types of hallucination observed in these scenarios, namely *extrinsic* and *intrinsic*.

1) *Extrinsic Hallucination*: The first type occurs when the model’s reasoning is logically coherent on its own terms but relies on speculative or ungrounded assumptions not supported by the input. In the first paragraph of the example in Fig. 4 (highlighted in green), the model correctly identifies that the overtaking trajectory is currently clear of other vehicles and obstacles. However, it then concludes that it is safe to drive around the stationary vehicle and pedestrians, despite the fact that the pedestrians could plausibly enter the path. Since this safety judgment is based on unconfirmable assumptions about future movement, and not on observable data, we consider this an example of extrinsic hallucination [6].

2) *Intrinsic Hallucination*: Intrinsic hallucination, on the other hand, is characterized by inconsistencies within the model’s own reasoning. In the second paragraph of Fig. 4, the model acknowledges that the pedestrians’ movements are unpredictable (highlighted in green), yet still concludes that



```

model_response: {
  "scene_description": "The scene depicts a single-lane road with a slight curve to the left. The ego vehicle is currently stopped behind a stationary vehicle (depicted in red) that is blocking the lane. There are three pedestrians (depicted as yellow circles) ahead of the ego vehicle, positioned at varying distances. The lane to the right is clear, and a pink trajectory indicates a possible maneuver to the right to avoid the obstacles.",
  "assessment_of_maneuver": "LEFT not feasible. The only feasible maneuver is to the right, as indicated by the pink trajectory. The lane to the right is clear of other vehicles and obstacles, making it safe to drive around the stationary vehicle and pedestrians.",
  "assessment_of_blockage": "The lane is blocked by a stationary vehicle and three pedestrians. The vehicle is not expected to move in the next minute. The pedestrians' movements are unpredictable, and the perception system cannot determine their velocity. Therefore, it is safer to maneuver around the obstacles.",
  "meta_action": "RIGHT"
}

```

Fig. 4: Model hallucination example: Visual misalignment or contextual misunderstanding leads the model to produce an invalid or unsafe decision.

maneuvering around them is the safer option (highlighted in red). This contradicts not only the internal logic of the response but also specific instructions from the prompt that explicitly recommend falling back to the meta-action CAREFUL when pedestrians are present, and their behavior cannot be predicted. As this conclusion directly contradicts both the model’s own understanding and its prompt conditioning, we classify it as intrinsic hallucination [6].

While our model achieves very good results in the benchmark, the hallucination types present significant risks in long-tail planning scenarios. They highlight the current limitations of MLLMs in aligning perception, reasoning, and action, even when equipped with structured visual and textual input. Therefore, automated vehicles should always rely on a higher level safety function.

B. Visual Understanding / Relevance of feasibility checks

While the planner’s holistic performance may appear relatively strong, we want to highlight that this robustness largely stems from the inclusion of heuristic feasibility checks and textual scene descriptions, rather than raw visual understanding. To illustrate this, we conducted a series of experiments on isolated scene understanding tasks using bird’s-eye view style images. While not exhaustive, these experiments reveal key limitations in the MLLM’s capabilities in this domain.

To assess the impact of reasoning strategies on performance, three variations are conducted. The first option is "No Reasoning", i.e., the model responds with a single word. In the next option "Simple Reasoning", the model offers basic reasoning about the scene. Lastly, with "Divide-and-conquer (D&Q) Reasoning", the model first analyzes the trajectory in three segments before concluding. Each test is conducted with thirty iterations, yielding valuable insights into the variability of responses.

In the first challenge, the model is tasked with determining

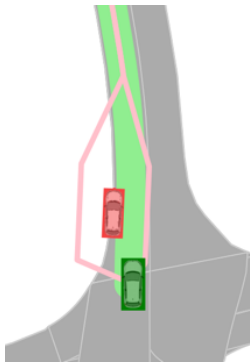
whether a green trajectory intersects with a grey obstacle. The results show that with "No Reasoning", the model consistently answered "No Intersection" correctly in all cases. With the "Simple Reasoning" approach, correct answers drop to 21 out of 30, with occasional contradictory responses. Finally, "D&Q Reasoning" decreases the model’s accuracy further, yielding correct answers in only 17 iterations. The model sometimes generates coherent reasoning, leading to an impression of strong understanding, yet it also generates incorrect responses. An example of an incorrect assessment of the scene is shown in Fig. 5.

In the next step, we analyze the influence of image resolution and color. Higher image resolution improves intersection detection, with correct answers increasing from 17 to 26 out of 30 in "D&Q Reasoning". Additionally, color choices significantly affect model responses. Neutral colors lead to more "No Intersection" predictions, while vibrant colors like red increase "Intersection" predictions, suggesting that color associations may influence reasoning.

To evaluate whether the limitations observed are model-specific, comparisons are undertaken with Anthropic’s Claude 3.5 [18], which is reputed for superior image reasoning. In path interference tests, Claude outperforms GPT-4o in accuracy and uncertainty assessment. However, both models exhibited challenges in drivable area compliance, indicating a general difficulty in extracting accurate visual information.

VI. CONCLUSION

In this paper, we introduced VisualLLMPlanner, a high-level decision-making engine for automated vehicles. The approach was evaluated on the interPlan benchmark, covering a variety of scenarios. For each scenario, a bird’s-eye view image and a textual description are generated, and the situation is expressed through a limited set of meta-actions such as LEFT, RIGHT, or NORMAL. These inputs are



```

model_response: {
...
"assessment_of_maneuver": "The left trajectory is feasible as it remains within
the drivable area and avoids the stationary vehicle. The right trajectory is not
feasible as it would lead into the undrivable area.",
...
"meta_action": "LEFT"
}

```

Fig. 5: Example failure: Overtaking maneuver incorrectly chosen due to visually ambiguous drivable area.

then provided to a GPT-4o model, which generates decision suggestions based on the scene.

Although certain limitations remain and not all situations can be resolved successfully, our evaluation shows that the approach outperforms other state-of-the-art LLM-based planners, particularly in challenging obstacle scenarios. Misinterpretations are primarily caused by common issues in LLMs, such as hallucination. Nevertheless, VisualLLMPlanner represents a promising candidate for strategic planning in automated vehicles.

Future extensions include incorporating additional consistency checks to filter out erroneous outputs that can lead to non-drivable plans and exploring richer multimodal inputs to enhance performance. The creation of real-world scenario representations for prompt generation could apply bird's-eye view models derived from vehicle perception data such as LiDAR.

So far, the experiments are conducted through the public API provided by OpenAI [19]. To evaluate the computation cost, the models can be deployed on a dedicated server infrastructure. Furthermore, the influence of the textual representation should be evaluated in an ablation study. In this work, nuPlan is used to generate the scenarios. However, other simulation frameworks, such as NVIDIA Isaac Sim or Gazebo [20], could be used for a more realistic evaluation.

REFERENCES

- [1] A. D. Thierer and R. Hagemann, "Removing Roadblocks to Intelligent Vehicles and Driverless Cars," *Wake Forest Journal of Law & Policy*, vol. 5, September 2015.
- [2] F. Golbabaeei, J. Dwyer, R. Gomez, A. Peterson, K. Cocks, A. Bubke, and A. Paz, "Enabling mobility and inclusion: Designing accessible autonomous vehicles for people with disabilities," *Cities*, vol. 154, 2024.
- [3] J. Zhao, W. Zhao, B. Deng, Z. Wang, F. Zhang, W. Zheng, W. Cao, J. Nan, Y. Lian, and A. F. Burke, "Autonomous driving system: A comprehensive survey," *Expert Systems with Applications*, vol. 242, 2024.
- [4] M. Hallgarten, J. Zapata, M. Stoll, K. Renz, and A. Zell, "Can vehicle motion planning generalize to realistic long-tail scenarios?," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5388–5395, IEEE, 2024.
- [5] Y. Wang, J. Jiang, S. Li, R. Li, S. Xu, J. Wang, and K. Li, "Decision-Making Driven by Driver Intelligence and Environment Reasoning for High-Level Autonomous Vehicles: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10362–10381, 2023.
- [6] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large Language Models: A Survey," *Preprint arXiv:2402.06196*, 2024.
- [7] OpenAI, "GPT-4 Technical Report." [Online]. Available: <http://arxiv.org/abs/2303.08774>, 2024. Accessed: Sep. 12, 2025.
- [8] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, "GPT-Driver: Learning to Drive with GPT," *Preprint arXiv:2310.01415*, 2023.
- [9] L. Wen, D. Fu, X. Li, X. Cai, T. MA, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao, "DiLu: A Knowledge-Driven Approach to Autonomous Driving with Large Language Models," in *The Twelfth International Conference on Learning Representations*, 2024.
- [10] Y. Zheng, Z. Xing, Q. Zhang, B. Jin, P. Li, Y. Zheng, Z. Xia, K. Zhan, X. Lang, Y. Chen, and D. Zhao, "PlanAgent: A Multi-modal Large Language Agent for Closed-loop Vehicle Motion Planning," *Preprint arXiv:2406.01587*, 2024.
- [11] T. Choudhary, V. Dewangan, S. Chandhok, S. Priyadarshan, A. Jain, A. K. Singh, S. Srivastava, K. M. Jatavallabhula, and K. M. Krishna, "Talk2BEV: Language-enhanced Bird's-eye View Maps for Autonomous Driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16345–16352, 2024.
- [12] F. Wulff, B. Schäufele, O. Sawade, D. Becker, B. Henke, and I. Radosch, "Early Fusion of Camera and Lidar for robust road detection based on U-Net FCN," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1426–1431, IEEE, 2018.
- [13] B. Schaeufele, O. Sawade, D. Pfahl, K. Massow, S. Bunk, B. Henke, and I. Radosch, "Forward-looking automated cooperative longitudinal control: Extending cooperative adaptive cruise control (CACC) with column-wide reach and automated network quality assessment," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, IEEE, 2017.
- [14] L. Chen, O. Sinavski, J. Hünemann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with LLMs: Fusing Object-Level Vector Modality for Explainable Autonomous Driving," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14093–14100, IEEE, 2024.
- [15] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Conference on Robot Learning*, pp. 1268–1281, PMLR, 2023.
- [16] M. Hallgarten, *Data-driven Behavior and Motion Planning for Autonomous Driving in Interactive Urban Environments*. PhD thesis, Eberhard Karls Universität Tübingen, 2025.
- [17] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles," *Preprint arXiv:2106.11810*, 2022.
- [18] A. J. Adetayo, M. O. Aborisade, and B. A. Sanni, "Microsoft Copilot and Anthropic Claude AI in education and library service," *Library Hi Tech News*, 2024.
- [19] G. Brockman, P. Welinder, M. Murati, and OpenAI, "OpenAI API." [Online]. Available: <https://openai.com/blog/openai-api>, 2020. Accessed: Feb. 10, 2026.
- [20] C. Symeonidis and N. Nikolaidis, "Simulation environments," in *Deep Learning for Robot Perception and Cognition*, pp. 461–490, Elsevier, 2022.