LanguageMPC: Large Language Models as Decision Makers for Autonomous Driving

Hao Sha[1], Yao Mu[2], Yuxuan Jiang[1], Guojian Zhan[1], Li Chen[2], Chenfeng Xu[3], Ping Luo[2], Shengbo Eben Li[1], *Senior Member, IEEE*, Masayoshi Tomizuka[3], *Life Fellow, IEEE*, Wei Zhan[3], *Member, IEEE*, Mingyu Ding[3], *Member, IEEE*
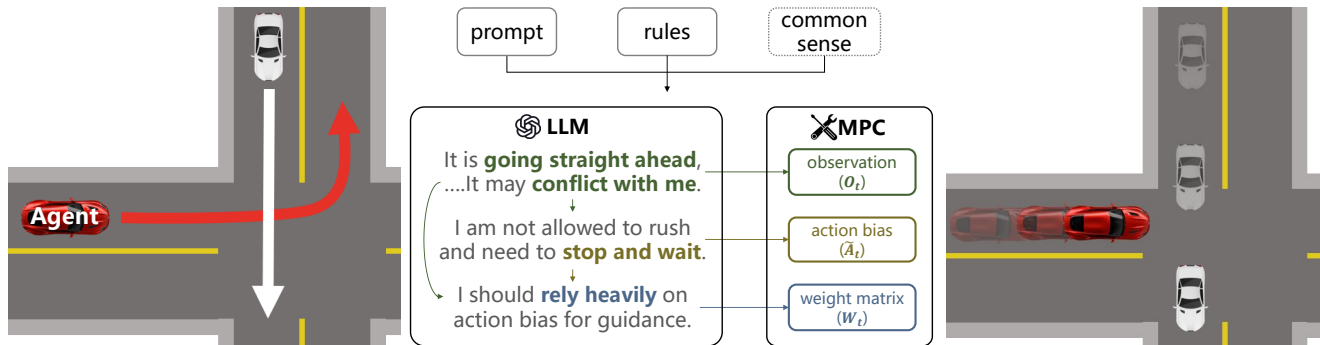
Fig. 1: For a left turn at an unsignalized intersection, the LLM makes a human-like decision to direct the MPC to slow down and wait, adhering to the traffic rule that left-turning vehicles must yield to oncoming traffic.

Abstract— **Existing learning-based autonomous driving (AD) systems face challenges in comprehending high-level information, generalizing to rare events, and providing interpretability. To address these problems, this work employs Large Language Models (LLMs) as a decision-making component for complex AD scenarios that require human commonsense understanding. We devise cognitive pathways to enable comprehensive reasoning with LLMs, and develop algorithms for translating LLM decisions into actionable driving commands. Through this approach, LLM decisions are seamlessly integrated with low-level controllers by guided parameter matrix adaptation. Extensive experiments demonstrate that our proposed method not only consistently surpasses baseline approaches in single-vehicle tasks, but also helps handle complex driving behaviors even multi-vehicle coordination, thanks to the commonsense reasoning capabilities of LLMs. This paper presents an initial step toward leveraging LLMs as effective decision-makers for intricate AD scenarios in terms of safety, efficiency, generalizability, and interoperability. We aspire for it to serve as inspiration for future research in this field.**

Index Terms— **Autonomous Vehicle Navigation; Autonomous Agents; Language-based Decision-Making**

I. INTRODUCTION

Imagine you are behind the wheel, approaching an unsignalized intersection and planning to turn left, with an oncoming vehicle straight ahead. Human drivers instinctively know to slow down and yield according to traffic rules, even if it is possible to accelerate through the intersection. However, existing advanced learning-based Autonomous Driving (AD) systems typically require complex rules or reward

---

[1]H. Sha, Y. Jiang, G. Zhan, and S. Li are with Tsinghua University.
[2]Y. Mu, L. Chen, and P. Luo are with the University of Hong Kong.
[3]C. Xu, M. Tomizuka, W. Zhan, and M. Ding are with the University of California, Berkeley. Corresponding author: Mingyu Ding

function designs to handle such scenarios effectively [1], [2]. This reliance on predefined rule bases often limits their ability to generalize to various situations.

Another challenge facing existing learning-based AD systems is the long-tail problem [3]. Both limited datasets and sampling efficiency [4] can present challenges for existing learning-based AD systems when making decisions in rare real-world driving scenarios. Chauffeurnet [5] demonstrated such limits where even 30 million state-action samples were insufficient to learn an optimal policy that mapped bird's-eye view images (states) to control (action).

Furthermore, the lack of interpretability [6] is a pressing issue for existing learning-based AD systems. A mature AD system must possess interpretability to gain recognition within society and regulatory entities, allowing it to be subject to targeted optimization and iterative improvements. Nevertheless, existing learning-based AD systems inherently resemble black boxes, making it challenging to discern their decision-making processes or understand the rationale behind their actions [4]. This lack of transparency can pose obstacles to the practical implementation of AD systems.

Considering the aforementioned challenges, a fundamental question arises: *Can we equip AD systems with the capability to think and drive like humans?* Our proposed solution involves employing a Large Language Model (LLM) to serve as the "brain" of the AD system.

Recent introductions of models like ChatGPT [7], have positioned LLMs as having a certain level of human-like reasoning capabilities, owing to their remarkable emergent abilities [8] and innovative techniques such as Instruct Following and In-Context Learning (ICL) [9]. LLMs can
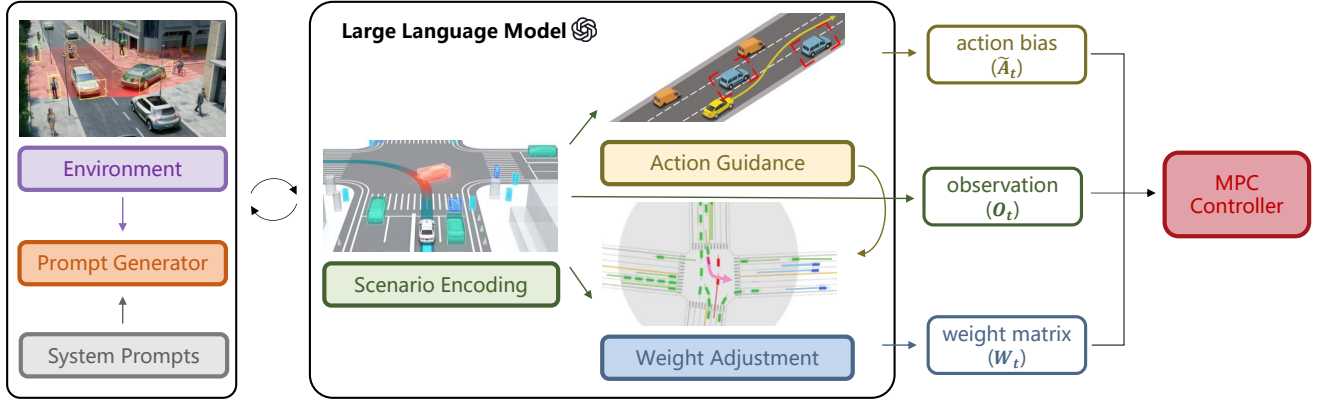
Fig. 2: Pipeline of our system with LLM as the core of high-level decision-making. The LLM processes environment and system descriptions from the prompt generator to perform 1) scenario encoding, 2) action guidance, and 3) confidence adjustment, which dynamically adjust observation matrices, action bias, and cost function weights, guiding the MPC to execute appropriate control behaviors.

think like humans [10], and reason about new scenarios by combining common sense, and the visible thinking process makes them somewhat interpretable. These features make LLMs a powerful solution to the problems faced by AD systems described above.

In this paper, we leverage LLMs to analyze and reason about various scenarios, enabling it to provide high-level decisions, and by tuning parameter matrix, we convert high-level decisions into mathematical representations to guide the bottom-level controller, Model Predictive Control (MPC). Fig. 1 illustrates the reasoning capabilities of our system for rare and complex scenarios, demonstrating its superiority in understanding high-level information, commonsense reasoning, and interpretability. Through quantitative experiments, we showcase that our system significantly surpasses existing learning-based and optimization-based methods for single-vehicle decision-making tasks, with Overall Cost decreasing by **18.1%** and **16.4%**. Additionally, through qualitative experiments, we demonstrate the capabilities of our system in addressing complex tasks, such as multi-vehicle joint control and driving behavior modulation guided by textual input. The main contributions of this paper are as follows:

1) We developed LanguageMPC, an end-to-end system that processes observations and outputs driving actions, guided by high-level, human-like decisions from the LLM. This integration enhances the understanding of complex scenarios and improves interpretability.

2) We develop a method that translates high-level textual decisions from the LLM into precise driving actions for the bottom-level controller, integrating the LLM's decision-making capabilities with the real-time control and robustness of the bottom-level controller.

3) We demonstrate through quantitative experiments that our LLM-enhanced autonomous driving system outperforms existing methods in key metrics. Additionally, it effectively manages complex tasks such as multi-vehicle coordination and driving behavior adjustment based on textual inputs.

II. RELATED WORK

Autonomous Driving Autonomy. Modularity [11], [12], [13], [14] plays an important role in many autonomous driving systems. It divides the complex autonomous driving task into multiple subproblems, including perception [15], [16], [17], planning [18], [19], and control [20], [21], making each module relevant and somewhat interpretable. However, it also presents challenges in areas such as compatibility, error accumulation and inefficiency[22]. Recently, there was a large progress in end-to-end deep learning methods for autonomous systems [23], [24], [25], [26]. Though autonomous driving systems have achieved remarkable successes in planning and decision-making [18], [19], there are still problems in terms of interpretability [6], [27], [28], [22]. At the same time, limitations in data and sampling efficiency [4] make it vulnerable to dealing with long-tail situations, especially interaction scenarios, in real-world environments [29].

Large Language Models for Autonomous Driving. The remarkable achievements of LLMs are undeniably captivating, demonstrating LLM's human-like reasoning skills and generalization of human commonsense [30], [31], [32], [33], [34]. In the autonomous driving industry, LLMs have the potential to revolutionize the paradigm of perception and decision-making. On the perceptual side, researchers combine visual encoders with LLM [35] to enhance the AD system's understanding of visual high-level concepts contained in images [36] and videos [37]. Recent works such as DriveGPT4 [38], HiLM-D [39], and Talk2BEV [40] have even demonstrated the powerful environment perception capabilities of LLMs enabled by vectorized visual embedding techniques. In planning and decision-making, extensive research has shown that pre-trained LLMs integrate human common sense and help make human-like decisions [41], [42], [43], [44]. Also, the involvement of LLMs greatly enhances interpretability and safety, providing the potential to comprehend the transparency of the decision process. [38], [43] showed that LLMs can reason about the transportation environment under the guidance of prompt and explain the
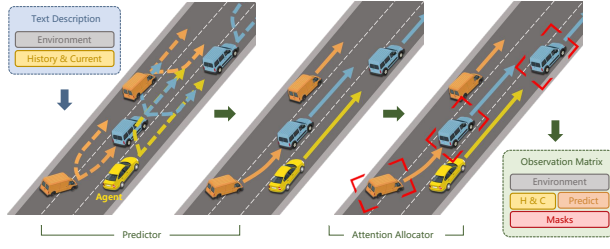
Fig. 3: Scenario encoding. The yellow vehicle is the agent. The LLM analyze and predict the scenario, creates reference trajectories and prioritizes relevant vehicles, and generates observation matrices for MPC input.

decision process. In addition, LLMs show the potential to address interoperability challenges [45], [46]. [47], [48], [49] uses language as a bridge between multimodal data, enabling the union of perception and decision-making, demonstrating powerful understanding and reasoning. However, the above work mainly focuses on demonstrating the perception, reasoning and high-level decision-making capabilities of LLMs, which are still lacking in generating final control actions. While [10], [47] can generate planning trajectories, [10] relies on predefined fixed rules, and [47] still requires the output of LLMs to be decoded into trajectories using a deep learning model, and the decoder still faces the problem of being limited by the training set. In this work, we combine the LLM directly with the controller MPC and use the powerful reasoning and decision-making capabilities of the LLM to directly guide the generation of the final control actions, allowing the LLM to drive like a human.

III. METHODOLOGY

We present an AD system that integrates the LLM with MPC to enhance decision-making in complex traffic scenarios. As illustrated in Fig. 2, the LLM serves as the core component for high-level reasoning and decision-making, while the MPC acts as the low-level controller to execute decisions in real time.

The decision-making process begins with a prompt generator that translates real-time scenario data into descriptive text. This text includes details about static road elements, dynamic information about vehicles, and system prompts like traffic rules. The LLM processes information through three sequential stages: 1) scenario encoding, 2) action guidance, and 3) weight adjustment. These stages collectively enable the system to accomplish scene understanding, reasoning, and high-level decision-making. Once these textual decisions are made, they are converted into mathematical representations: the observation matrix, action bias, and weight matrix. These elements guide bottom-level controller, the MPC, to give specific driving actions.

This framework integrates the advanced reasoning and decision-making capabilities of the LLM with the robustness and real-time control efficiency of the MPC:

1) The LLM functions as the "brain" of the AD system, enabling high-level reasoning and decision-making.
2) The MPC implements the LLM's decisions under soft constraints, ensuring robust yet flexible control.
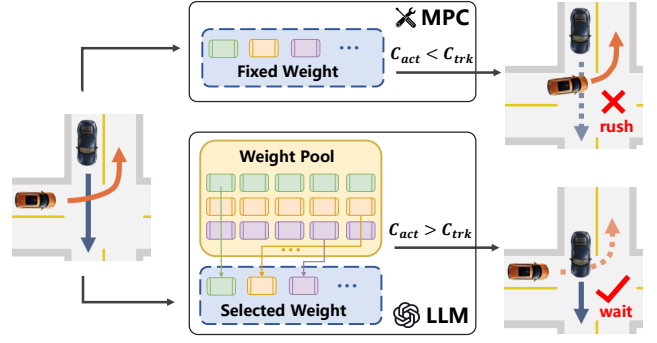

Fig. 4: Weight Adjustment. The LLM selects parameters from the weight pool to form a selected weight matrix $\mathbf{W} = \mathbf{W}_{sel}$, balancing the four components: tracking, action panelty, smoothness, and safety.

3) Together, the LLM and MPC form a dual-frequency system that balances the longer inference time of the LLM with the real-time execution capabilities of the MPC, achieving effective real-time control, as detailed in Section III-D.

A. MPC Controller

The MPC solves a finite-time open-loop optimization problem online at each time step, based on the current measurement information obtained, and applies the initial control action from the optimal sequence to the vehicle.

The cost function of MPC is defined in the context of Markov Decision Process (MDP), which is commonly used to formulate vehicle control problems: (S, U, C, P, p_0), where S represents the state space, U the action space, $C : S \times U \mapsto \mathbb{R}$ the cost function, $P : S \times U \mapsto S$ the state transition function, and p_0 the initial state distribution. The objective of the MPC is to find a sequence of actions $\mathbf{u}_{1:H} = \mathbf{u}_1, \ldots, \mathbf{u}_H$ that minimizes the expected cumulative cost $J(\mathbf{u}_{1:H}) = \sum_{t=1}^{H} C(\mathbf{s}_t, \mathbf{u}_t)$.

The cost function takes the following form:

$$C(\mathbf{s}, \mathbf{u}) = \sum_{i=0}^{M} w_i \cdot \mathrm{n}_i\big(r_i(\mathbf{s}, \mathbf{u}, \zeta_i)\big), \qquad (1)$$

where $w \in \mathbb{R}+$ are non-negative weights, $\mathrm{n}(\cdot) : \mathbb{R} \to \mathbb{R}+$ is a twice-differentiable norm that reaches its minimum value at 0, and r_i are residual terms parameterized by ζ_i. Designing a universal set of weights and residual terms for all driving scenarios is impractical due to the diversity of conditions encountered in real-world driving. Additionally, increasing the complexity of the cost function may reduce robustness [50]. Therefore, we use a simplified set of residual terms that incorporate action biases, adapting the weight matrices $\mathbf{W} = \{W_i\}$ based on LLM's high-level decisions.

The overall cost function is constructed as:

$$C(\mathbf{s}, \mathbf{u}) = C_{trk}(\mathbf{s}, \mathbf{u}) + C_{act}(\mathbf{s}, \mathbf{u}) + C_{saf}(\mathbf{s}, \mathbf{u}) \qquad (2)$$

We first generate a reference trajectory without considering signals and other traffic participants and then aim to follow that trajectory:

$$C_{trk} = X^T W_x X + V^T W_v V + \Theta^T W_\theta \Theta, \qquad (3)$$

where $X = (x, y)^T$ represents the location, $V = (v_x, v_y)^T$ the velocity, $\Theta = (\theta, \psi)^T$ with θ being the steering angle and ψ the yaw angle.

To ensure smooth control, we impose limits on acceleration a and steering angle θ:

$$C_{act} = \overline{A}^T W_s \overline{A} + \nabla A^T W_g \nabla A + \nabla^2 A^T W_l \nabla^2 A \quad (4)$$

where $A = (a, \theta)^T$ and $\overline{A} = A - \hat{A}$, with $\hat{A}$ being the action bias given by LLM.

Lastly, we include a safety cost to maintain a safe distance from other vehicles:

$$C_{saf} = w_{saf} \cdot \sum_{i=0}^{N} (d_i - \hat{d}_i)^2, \quad (5)$$

where d_i represents the actual distance to other vehicles, and $\hat{d}_i$ is the desired safety distance.

All weight matrices W_i are diagonal. The LLM adapts the parameters $\mathbf{s}_t$, $\mathbf{W}$, and $\hat{A}$ for complex driving scenarios to guide the MPC controller.

B. Scenario Encoding

After the processing of the prompt generator, the LLM obtains current and short-term historical information about the scenario. This information is structured into an observation matrix, $\mathbf{O}_t = O_t^{st}, O_t^{dy}$, for the MPC, where O_t^{st} represents static environmental elements and O_t^{dy} represents dynamic information about surrounding vehicles. In this section, we will describe how the LLM optimizes and modifies $\mathbf{O}_t$ to improve planning of MPC.

Effective decision-making requires not only understanding the current and historical states but also predicting future states. To generate these predictions, we first create potential reference trajectories for the ego vehicle using a Bessel curve:

$$B_{ref} = \sum_{i=0}^{n} P_i \binom{i}{n} \gamma^i (1 - \gamma)^{n-i} \quad (6)$$

where P_i represent the control points, including waypoint locations and the ego vehicle's position, and $\gamma \in [0, 1]$. This reference trajectory provides a potential path for the ego vehicle without considering other traffic participants. The LLM then acts as a predictor to select the most suitable reference trajectory based on the scenario context. This selected trajectory is used as the tracking target, as discussed in section III-A. Next, the same process is applied to generate predicted trajectories for other traffic participants, with the LLM estimating each vehicle's likely path based on its behavior and interactions. The dynamic observation matrix is then represented as a combination of historical data and these predicted trajectories, $O_t^{dy} = (H_t, P_t) \in \mathbb{R}^{N \times C}$, where N is the number of traffic participants.

In complex traffic scenarios, it is crucial to focus attention on relevant vehicles to avoid unnecessary computation and potential logical conflicts. To achieve this, as shown on the right side of Fig. 3, the LLM functions as an attention allocator. It identifies the vehicles that need to be prioritized and generates a binary mask $M_t \in \mathbb{R}^{N \times 1}$. This mask effectively filters out irrelevant vehicles by setting their corresponding

TABLE I: Evaluation of single-vehicle decision-making in various scenarios. SI: Signalized Intersection, USI: Unsignalized Intersection, RAB: Roundabouts, EOA: Emergency Obstacle Avoidance. Lower values indicate better performance.

Scenario	Method	Col.	Fail	Ineff	Time	Penalty		O.C.
						Acc	Dist	
SI	RL	6	0	34.1	**14.1**	3.78	3.38	60.2
	MPC	2	4	25.5	17.7	3.14	3.05	56.3
	Ours	**0**	**0**	**13.9**	25.7	**1.31**	**1.20**	**44.8**
USI	RL	9	**0**	67.5	**29.4**	5.27	3.22	92.0
	MPC	2	4	74.0	30.7	4.30	2.55	87.2
	Ours	**0**	1	**33.7**	42.2	**1.94**	**0.98**	**67.0**
Lane	RL	0	0	2.27	6.8	0.15	0.09	8.71
	MPC	0	0	4.14	6.8	0.20	0.08	9.41
	Ours	**0**	**0**	**1.13**	**6.7**	**0.08**	**0.03**	**7.63**
RAB	RL	5	0	29.3	**30.3**	1.64	0.71	50.8
	MPC	1	3	29.3	30.4	1.61	0.68	**50.6**
	Ours	**0**	**0**	**26.8**	31.9	**1.51**	**0.65**	50.8
EOA	RL	11	**0**	32.3	16.9	2.99	1.96	51.3
	MPC	8	4	33.4	17.3	3.34	2.06	54.3
	Ours	**3**	2	**28.8**	**16.7**	**2.60**	**1.79**	**47.3**

data to zero in the observation matrix. The final dynamic observation matrix for the MPC is computed as:

$$O_t^{dy} = M_t J^T(H_t, P_t), \quad (7)$$

where $J \in \mathbb{R}^{N \times 1}$ is a vector of ones. This approach ensures that the MPC focuses only on the relevant traffic participants, enabling efficient and effective decision-making.

C. Action Guidance and Weight Adjustment

After the scenario encoding stage, LanguageMPC will further give action guidance. Since most LLMs have limited sensitivity to precise numerical values [51], we discretize the action $A = (a, \theta)^T$ into a set of intervals. The midpoint of the LLM-selected interval is then used as action bias $\hat{A}$, which guides the MPC to generate control actions that align with the high-level decisions of the LLM.

One challenge with MPC is its limited ability to generalize across diverse scenarios, as it relies on predefined weights $\mathbf{W} = W_i$ [50]. In our approach, as detailed in Section III-A, the weight matrices $\mathbf{W} = \{W_x, W_\theta, W_s, W_g, W_l, w_{safe}\}$ are categorized into four components: tracking $\{W_x, W_\theta\}$, action panelty W_s, smoothness $\{W_g, W_l\}$, and safety w_{saf}. Unlike predefined weight matrices that must accommodate all possible scenarios, LanguageMPC dynamically selects the appropriate weights from a weight pool for each specific scenario to balance these four components, as shown in Fig. 4. This approach allows for tuning of weight matrices tailored to each scenario and ensures that the MPC maintains robust control without relying on direct action commands from the LLM.

D. Dual-frequency System

The decision-making speed of the LLM in LanguageMPC is relatively slow due to its large number of parameters. To address this, we implement a dual-frequency system inspired by DriveVLM [52]. In this system, we separate decision-making into two levels: low-frequency for high-level decisions and high-frequency for trajectory planning and control.

RL

Overtake at intersection!

MPC

Wait too long!

Ours

agent

vehicle_32

vehicle_17

Follow and overtake in lane!

LLM: **vehicle_32** is **turning left** and **stopping to wait** for the straight-ahead convoy to pass **due to traffic rules**, so it's **unlikely to affect** the movement of the ego vehicle.
LLM: **vehicle_17** is **traveling straight** ahead of me and **slow**, so it is **very likely to affect** the movement of the ego vehicle.

LLM: Traffic rules **prohibit overtaking in intersections**, so I should **follow vehicle_17 through** the intersection and **then overtake** it.
LLM: This decision is **based on traffic rules** that I must comply with, so the **MPC** should **rely heavily** on my decisions.
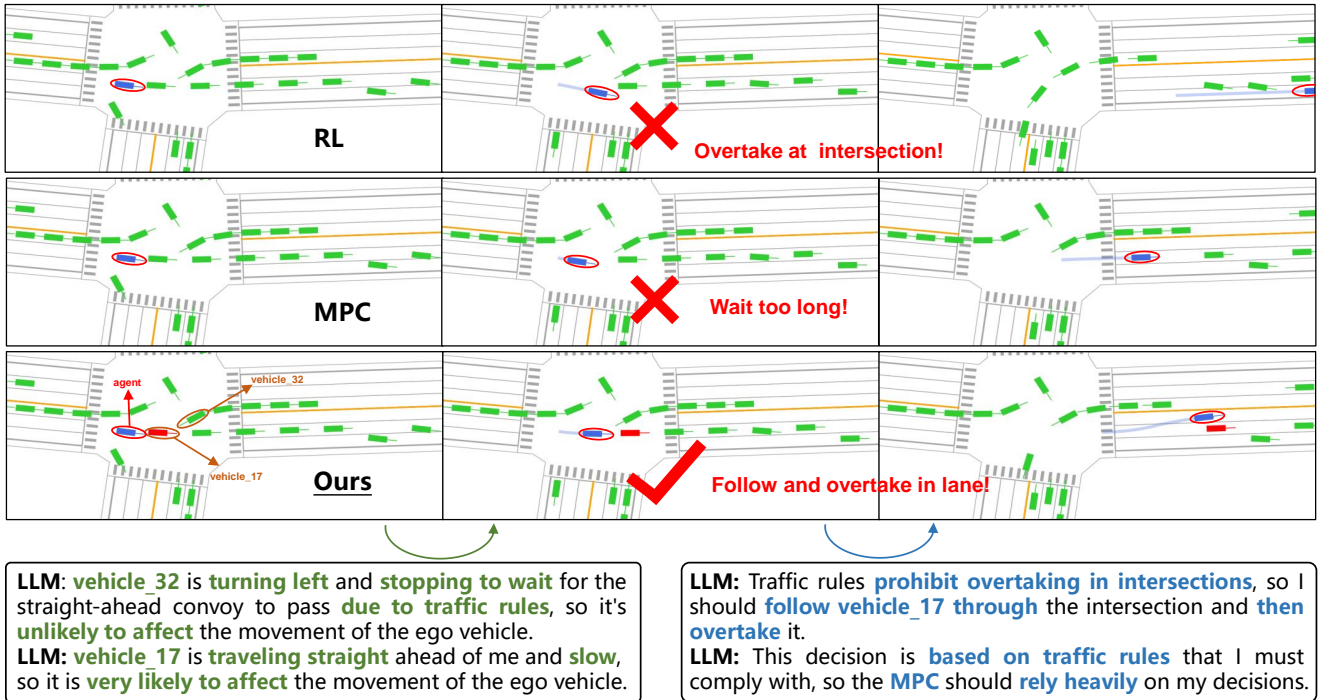
Fig. 5: The ego vehicle is traveling straight through an unsignalized intersection. The red vehicle(s) in the last row represent those to which the LLM assigns attention, meaning information about these vehicles appears in the MPC dynamic observation matrix O_t^{dy}. This example demonstrates the LLM's ability to interpret high-level information, recognizing that "vehicle_32" is yielding and focusing on the vehicle ahead to safely and compliantly navigate the intersection.

At the low-frequency level, the LLM makes strategic decisions every few seconds. These strategic decisions are sufficient to adapt to changes in driving conditions that occur over a longer timescale. In contrast, the MPC operates at the high-frequency level, performing fine-grained trajectory planning and control between LLM decisions. It generates real-time control actions based on the high-level guidance provided by the LLM, as described in the previous sections.

This dual-frequency approach balances the slower, computationally intensive inference of the LLM with the real-time demands of vehicle control, allowing LanguageMPC to achieve robust performance in dynamic driving scenarios.

IV. EXPERIMENTS

In this section, we evaluate the performance of LanguageMPC using GPT-3.5 as the base LLM model. We assess its reasoning and decision-making capabilities through a series of experiments designed to test its effectiveness in single-vehicle decision-making and its adaptability to complex scenarios, such as driving behavior modulation and multi-vehicle coordination. We conduct quantitative experiments to measure its performance on various driving tasks and qualitative analyses to showcase its ability to handle challenging situations, including rare corner cases and coordinated actions between multiple vehicles.

A. Settings

Simulator. We conduct simulation experiments using the SUMO simulator [53] with IDSim [54] to generate dynamic traffic scenarios, including vehicles, bicycles, and pedestrians. Four distinct driving scenarios are evaluated: sig-

nalized intersections, unsignalized intersections, driveways, and roundabouts. Each scenario is initialized with varying road structures, traffic densities, and participant behaviors to simulate challenging conditions. We exclude straightforward situations, such as driving straight within a lane, and instead focus on complex corner cases, with 25 different scenes reserved for each scenario.

Baselines. We compare LanguageMPC against two baselines: **1) Model Predictive Control (MPC)** [55]: We use a traditional MPC system with a fixed weight matrix, consistent with the control module in LanguageMPC. The dynamic observation matrix O_t^{dy} incorporates all traffic participants within the sensing range. To predict the movement of these participants, all waypoints are used as control points for generating reference trajectories. The trajectory closest to each participant's current path is chosen as the predicted trajectory. This baseline has been finely tuned and validated using real-world vehicle experiments conducted at iDLab, Tsinghua University. **2) Reinforcement Learning-Based Planning (RL)** [55], [56]: We employ a finite-horizon approximate dynamic programming (ADP) approach. The MPC observation matrix O_t is used as input to output the control action $A = (a, \theta)^T$. The RL agent is trained using IDSim in the four scenarios mentioned above, with randomly generated scenarios. Training is performed over 2×10^5 epochs.

B. Metrics

Failure and Collision Cases (Fail/Col.): We track failure cases where the ego vehicle fails to reach its target area within a 200-second time frame. Collision cases are recorded separately to monitor safety incidents.

TABLE II: Ablation experiments. O^{dy} indicates the use of the scenario encoder to modify the MPC observation matrix, while $\hat{A}\&W$ indicates the use of the LLM for action bias $\hat{A}$ and weight matrix $\mathbf{W}$ generation.

Scenario	O^{dy}	$\hat{A}\&W$	Coll.	Fail	Ineff	Time	Penalty		O.C.
							Acc	Dist	
USI	×	×	2	4	74.0	30.7	4.30	2.55	87.2
	✓	×	2	3	69.8	**28.7**	3.95	2.32	81.0
	×	✓	**0**	2	42.0	44.0	2.37	1.21	74.5
	✓	✓	**0**	1	**33.7**	42.2	**1.94**	**0.98**	**67.0**
RAB	×	×	1	3	29.3	30.4	1.61	0.68	50.6
	✓	×	1	3	29.4	**30.3**	1.61	0.69	50.6
	×	✓	0	0	27.8	30.9	1.53	0.67	**50.2**
	✓	✓	**0**	**0**	**26.8**	31.9	**1.51**	**0.65**	50.8

Inefficiency (Ineff): We measure traffic flow efficiency by calculating the average speed difference between the ego vehicle's maximum speed and its current speed:

$$P_{eff} = \frac{1}{N}\sum_{i=1}^{N}(v_i^{max} - v_i) \qquad (8)$$

This metric focuses on the lead vehicle within each convoy, as it is most affected by the ego vehicle's driving behavior. Vehicles stopped at red lights are excluded.

Time Efficiency (Time): This metric records the time taken by the ego vehicle to reach its target area, reflecting the overall driving efficiency.

Safety Penalty (Penalty): The penalty metric assesses the safety of the ego vehicle's driving behavior.It considers both the distance between the ego vehicle and surrounding vehicles, and the deceleration of those vehicles. Smaller distances and higher deceleration rates indicate more unsafe driving and contribute to a higher penalty score:

$$P_{acc} = \sum_{i=1}^{N} ReLU(d_i - d_0), P_{dist} = \sum_{i=1}^{N} d_i f(a_0 - a_i) \quad (9)$$

where f is a step function from 0 to 1 with $a_0 - a_i = 0$ as the cutoff.

Overall Cost (O.C.): To provide a comprehensive evaluation, we combine the above metrics into an overall cost function using weighted values:

$$\text{O.C.} = 30P_{eff} + t + 500P_{acc} + 5P_{dist} \qquad (10)$$

These weights balance the influence of each metric, ensuring proportional contribution to the overall cost.

C. Single-vehicle Decision-making

The quantitative results of single-vehicle decision-making are presented in Table I. Our system shows significant reductions in overall cost across all four scenario types, indicating improved driving performance. We report minimal failures and no collisions, demonstrating the safety and reliability of our approach.

In intersection and roundabout scenarios, our method slightly increases travel time compared to the baselines. However, this trade-off is offset by notable improvements in traffic flow efficiency and reduced safety penalties, reflecting a more cautious and regulation-compliant driving style. In lane-changing scenarios, our system outperforms both RL and MPC across all metrics, demonstrating superior decision-making in complex overtaking and lane-changing situations.
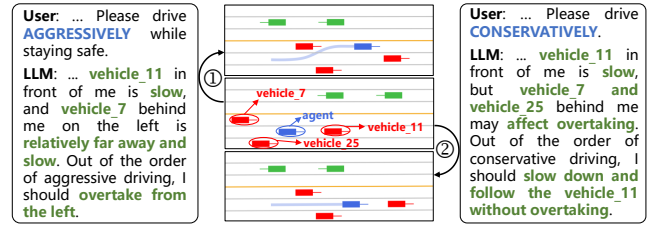

Fig. 6: The ego vehicle adjusts its driving style based on user instructions. When directed to drive conservatively, it slows down and follows the vehicle ahead. Under aggressive driving commands, it safely overtakes the slower vehicle.

Ablation Study. To gain deeper insights into the system's capabilities, we performed ablation experiments in two challenging driving scenarios: unsignalized intersections and roundabouts, as shown in Table II. These ablations highlight the contributions of each component of LanguageMPC. When using only the scenario encoder, significant improvements are observed in intention prediction and attention allocation compared to baseline MPC. In contrast, using only action bias and weight matrix adjustments leads to substantial improvements in all metrics except time. The full configuration, integrating both components, achieves the best performance, showcasing the synergistic effect of scenario understanding and high-level decision-making.

Fig. 5 provides a qualitative example of how these components work together. The ego vehicle navigates an unsignalized intersection, demonstrating LanguageMPC's ability to predict the intentions of surrounding vehicles, allocate attention effectively, and make decisions aligned with traffic regulations.

1) Intention Prediction and Attention Allocation. The two left side of Fig. 5 demonstrate the effectiveness of scenario encoding. Unlike the MPC, which fails to anticipate that "vehicle_32" intends to stop and wait, LanguageMPC correctly interprets that the vehicle has a lower right-of-way and will yield. Consequently, LanguageMPC focuses on the vehicle ahead that may affect the ego vehicle's path and follows it through the intersection. This demonstrates how the LLM enhances the system's ability to understand complex traffic dynamics and allocate attention effectively, leading to more rational driving behavior.

2) High-level Reasoning. In the scenario shown in Fig. 5, the ego vehicle, despite being slowed down by a vehicle ahead, must refrain from overtaking within the intersection due to traffic regulations. The RL method incorrectly decides to overtake the vehicle within the intersection, violating traffic rules. In contrast, LanguageMPC understands the rules and chooses to follow the vehicle until it exits the intersection before overtaking, as shown in the dialogue box on the right. This highlights the LLM's capability to reason about high-level constraints and make decisions that comply with traffic regulations, demonstrating a more human-like understanding of complex scenarios.

D. Text-modulated Driving

Driving Style Adjustment. Users often wish to customize the driving behavior of AD systems to match their pref-
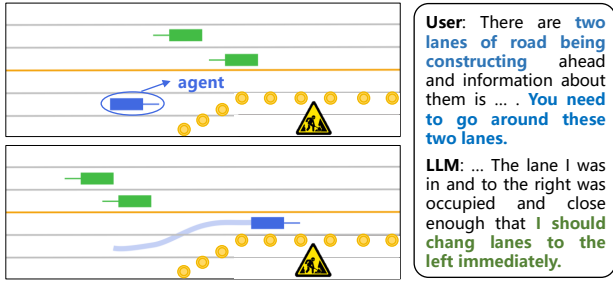
Fig. 7: The LanguageMPC changes lanes under textual guidance to safely navigate a road construction zone, showcasing the system's ability to follow user instructions and handle complex driving scenarios effectively.

erences for efficiency and comfort. Traditional learning-based or optimization-based systems require complex rule or reward function designs to achieve this [57]. Our approach simplifies this process by allowing users to provide simple textual descriptions through an intuitive interface. As shown in Fig. 6, the LLM can easily interpret commands to drive more aggressively or conservatively. For example, when instructed to drive conservatively, the vehicle opts to slow down and follow the car ahead, whereas an aggressive driving style results in strategic overtaking when safe.

Textual Guidance for Complex Scenarios. Complex driving situations, such as road construction, can be challenging for many AD systems [1]. Our approach addresses this by enabling users to provide textual instructions that guide decision-making. In the road construction scenario depicted in Fig. 7, the LLM received textual guidance and correctly changed lanes to avoid the blocked area. This capability allows our system to adapt seamlessly to unusual or dynamic situations that are typically difficult for AD systems to handle.

E. Multi-vehicle Joint Control

Multi-vehicle joint control is a promising approach to improving transportation efficiency and safety. Traditional centralized and distributed methods often perform well under known traffic conditions but may face challenges when adapting to dynamic and less predictable environments [58]. To address these challenges, we propose a hybrid solution where each vehicle is controlled by a distributed LLM, while a central LLM serves as the convoy's coordinator. Each distributed LLM communicates its local status to the central LLM, which then issues coordinated commands based on the collective information, enabling real-time, adaptive decision-making even in complex and unforeseen scenarios.

Fig. 8 illustrates a classic corner case: a narrow road encounter between two vehicles. For a single-vehicle AD system, handling this situation requires complex gaming. However, our multi-vehicle joint control approach simplifies this problem by leveraging both centralized and distributed LLMs. In this scenario, two distributed LLMs recognize they are in a "meeting" situation and report their status to a central LLM. Based on this information and a self-assessment of each vehicle's position and speed, the central
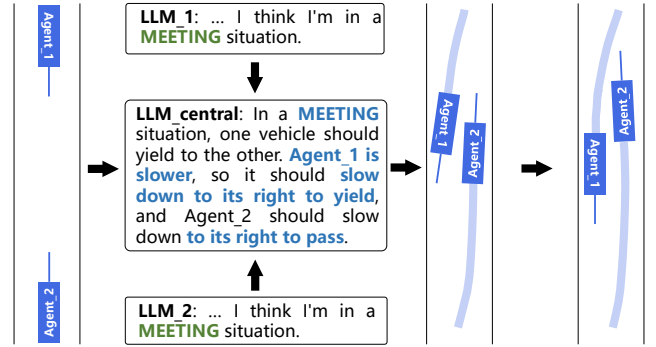

Fig. 8: The two vehicles meet on a narrow road. Distributed LLMs detect the situation and report to the central LLM, which coordinates their actions, one vehicle yields while the other proceeds. This demonstrate effective multi-vehicle coordination.

LLM coordinates the actions of both vehicles. It instructs one vehicle to decelerate and yield while allowing the other to proceed slowly, ensuring safe and efficient passage. Each distributed LLM then adjusts its vehicle's control according to these high-level instructions. This example demonstrates the effective collaboration between centralized decision-making and distributed vehicle control, optimizing multi-vehicle coordination in complex scenarios.

V. CONCLUSION

This paper demonstrates the effectiveness of LLMs as a core component for high-level decision-making in AD systems. By integrating LLMs with MPC, our approach significantly improves performance in key areas such as safety, efficiency, and adaptability, particularly in complex driving scenarios. The reasoning capabilities and interpretability of LLMs help address limitations found in traditional learning-based methods, enhancing both transparency and flexibility. We hope this work inspires continued exploration of LLMs in autonomous driving, paving the way for safer and more efficient transportation solutions.

REFERENCES

[1] L. Chen, Y. Li, C. Huang, Y. Xing, D. Tian, L. Li, Z. Hu, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, and F.-Y. Wang, "Milestones in autonomous driving and intelligent vehicles—part i: Control, computing system design, communication, hd map, testing, and human behaviors," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 9, pp. 5831–5847, 2023.

[2] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE T-ITS*, vol. 23, no. 6, pp. 4909–4926, 2022.

[3] T. Buhet, E. Wirbel, and X. Perrotton, "Conditional vehicle trajectories prediction in carla urban environment," in *2019 IEEE/CVF ICCVW*, 2019, pp. 2310–2319.

[4] S. Atakishiyev, M. Salameh, H. Yao, and R. Goebel, "Explainable artificial intelligence for autonomous driving: A comprehensive overview and field guide for future research directions," 2023.

[5] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *Advances in neural information processing systems*, vol. 31, 2018.

[6] P. Gohel, P. Singh, and M. Mohanty, "Explainable ai: current status and future directions," 2021.

[7] OpenAI, "Introducing chatgpt," https://openai.com/blog/chatgpt/, 2023.

[8] J. Wei, Y. Tay, R. Bommasani, *et al.*, "Emergent abilities of large language models," 2022.

[9] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, "A survey on in-context learning," 2023.

[10] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao, "Drive like a human: Rethinking autonomous driving with large language models," 2023.

[11] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[12] J. Daudelin, G. Jing, T. Tosun, M. Yim, H. Kress-Gazit, and M. Campbell, "An integrated system for perception-driven autonomy with modular robots," *Science Robotics*, vol. 3, no. 23, p. eaat4983, 2018.

[13] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," 2015.

[14] N. Akai, L. Y. Morales, T. Yamaguchi, E. Takeuchi, Y. Yoshihara, H. Okuda, T. Suzuki, and Y. Ninomiya, "Autonomous driving based on accurate localization using multilayer lidar and dead reckoning," in *2017 IEEE 20th ITSC*, 2017, pp. 1–6.

[15] X. Li, T. Ma, Y. Hou, B. Shi, Y. Yang, Y. Liu, X. Wu, Q. Chen, Y. Li, Y. Qiao, *et al.*, "Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion," in *CVPR*, 2023, pp. 17 524–17 534.

[16] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *ECCV*. Springer, 2022, pp. 1–18.

[17] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *CVPR*, 2021, pp. 11 784–11 793.

[18] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 583–601, 2003.

[19] C. Zhang, R. Guo, W. Zeng, Y. Xiong, B. Dai, R. Hu, M. Ren, and R. Urtasun, "Rethinking closed-loop training for autonomous driving," in *ECCV*. Springer, 2022, pp. 264–282.

[20] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE ICRA*. IEEE, 2018, pp. 3803–3810.

[21] M. A. Johnson and M. H. Moradi, *PID control*. Springer, 2005.

[22] P. S. Chib and P. Singh, "Recent advancements in end-to-end autonomous driving using deep learning: A survey," 2023.

[23] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," in *CVPR*, 2021, pp. 14 403–14 412.

[24] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *CVPR*, 2023, pp. 17 853–17 862.

[25] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," *ECCV*, pp. 414–430, 2020.

[26] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *TNNLS*, vol. 33, no. 4, pp. 1364–1384, 2020.

[27] A. B. Arrieta, N. Díaz-Rodríguez, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," 2019.

[28] W. Xu, "From automation to autonomy and autonomous vehicles: Challenges and opportunities for human-computer interaction," *interactions*, vol. 28, 01 2021.

[29] L. Kong, Y. Liu, X. Li, R. Chen, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu, "Robo3d: Towards robust and reliable 3d perception against corruptions," *arXiv preprint arXiv:2303.17597*, 2023.

[30] N. Bian, X. Han, L. Sun, H. Lin, Y. Lu, and B. He, "Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models," 2023.

[31] J. J. Nay, "Law informs code: A legal informatics approach to aligning artificial intelligence with humans," *Nw. J. Tech. & Intell. Prop.*, vol. 20, p. 309, 2022.

[32] A. Chowdhery, S. Narang, J. Devlin, and et al., "Palm: Scaling language modeling with pathways," 2022.

[33] L. Ouyang, J. Wu, X. Jiang, D. Almeida, and et al., "Training language models to follow instructions with human feedback," 2022.

[34] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," 2021.

[36] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[37] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," 2023.

[38] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," 2024.

[39] X. Ding, J. Han, H. Xu, W. Zhang, and X. Li, "Hilm-d: Towards high-resolution understanding in multimodal large language models for autonomous driving," 2023.

[40] 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," 2023.

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

[42] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," 2023.

[43] 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.

[44] 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," 2024.

[45] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A. J. Willmott, D. Birch, D. Maund, and J. Shotton, "Driving with llms: Fusing object-level vector modality for explainable autonomous driving," 2023.

[46] A. Kamath, P. Anderson, S. Wang, J. Y. Koh, A. Ku, A. Waters, Y. Yang, J. Baldridge, and Z. Parekh, "A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning," in *CVPR*, 2023, pp. 10 813–10 823.

[47] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," 2023.

[48] Y. Jin, X. Shen, H. Peng, X. Liu, J. Qin, J. Li, J. Xie, P. Gao, G. Zhou, and J. Gong, "Surrealdriver: Designing generative driver agent simulation framework in urban contexts based on large language model," 2023.

[49] A. Keysan, A. Look, E. Kosman, G. Gürsun, J. Wagner, Y. Yao, and B. Rakitsch, "Can you text what is happening? integrating pre-trained language encoders into trajectory prediction models for autonomous driving," 2023.

[50] I. Askari, B. Badnava, T. Woodruff, and S. Zeng, "Sampling-based nonlinear mpc of neural network dynamics with application to autonomous vehicle motion planning," 05 2022.

[51] J. Ahn, R. Verma, R. Lou, D. Liu, R. Zhang, and W. Yin, "Large language models for mathematical reasoning: Progresses and challenges," 2024.

[52] X. Tian, X. Gu, B. Li, Y. Liu, C. Hu, Y. Wang, K. Zhan, P. Jia, X. Lang, and H. Zhao, "Drivevlm: The convergence of autonomous driving and large vision-language models," 2024.

[53] SUMO, "Sumo," https://sumo.dlr.de/docs/index.html/, 2000.

[54] Y. Jiang, G. Zhan, Z. Lan, C. Liu, B. Cheng, and S. E. Li, "A reinforcement learning benchmark for autonomous driving in general urban scenarios," *IEEE T-ITS*, pp. 1–11, 2023.

[55] Y. Guan, Y. Ren, Q. Sun, S. E. Li, H. Ma, J. Duan, Y. Dai, and B. Cheng, "Integrated decision and control: Toward interpretable and computationally efficient driving intelligence," *IEEE Transactions on Cybernetics*, vol. 53, no. 2, pp. 859–873, 2023.

[56] Y. Ren, J. Jiang, G. Zhan, S. E. Li, C. Chen, K. Li, and J. Duan, "Self-learned intelligence for integrated decision and control of automated vehicles at signalized intersections," *IEEE T-ITS*, vol. 23, no. 12, pp. 24 145–24 156, 2022.

[57] W.-J. Chang, C. Tang, C. Li, Y. Hu, M. Tomizuka, and W. Zhan, "Editing driver character: Socially-controllable behavior generation for interactive traffic simulation," 2023.

[58] J. Wang, Y. Zheng, K. Li, and Q. Xu, "Deep-lcc: Data-enabled predictive leading cruise control in mixed traffic flow," *IEEE Transactions on Control Systems Technology*, pp. 1–17, 2023.