EMPOWERING AUTONOMOUS DRIVING WITH LARGE LANGUAGE MODELS: A SAFETY PERSPECTIVE

Yixuan Wang[1*] **Ruochen Jiao**[1*] **Sinong Simon Zhan**[1] **Chengtian Lang**[1]
Chao Huang[2] **Zhaoran Wang**[1] **Zhuoran Yang**[3] **Qi Zhu**[1]

[1]Northwestern University, USA [2]University of Southampton, UK [3]Yale University, USA

ABSTRACT

Autonomous Driving (AD) encounters significant safety hurdles in long-tail unforeseen driving scenarios, largely stemming from the non-interpretability and poor generalization of the deep neural networks within the AD system, particularly in out-of-distribution and uncertain data. To this end, this paper explores the integration of Large Language Models (LLMs) into AD systems, leveraging their robust common-sense knowledge and reasoning abilities. The proposed methodologies employ LLMs as intelligent decision-makers in behavioral planning, augmented with a safety verifier shield for contextual safety learning, for enhancing driving performance and safety. We present two key studies in a simulated environment: an adaptive LLM-conditioned Model Predictive Control (MPC) and an LLM-enabled interactive behavior planning scheme with a state machine. Demonstrating superior performance and safety metrics compared to state-of-the-art approaches, our approach shows the promising potential for using LLMs for autonomous vehicles.

1 INTRODUCTION

The current mainstream of autonomous vehicle (AV) software pipeline consists of key modules: perception (Feng et al., 2020; Man et al., 2023), prediction (Nayakanti et al., 2023; Jiao et al., 2022), planning(Liu et al., 2023b), and control. Deep neural networks (DNNs) have become integral to perception and prediction, with a growing interest in planning and control. However, the black-box nature of DNNs, along with their inherent uncertainties from learning algorithms, presents challenges in ensuring the safety of closed-loop AV systems. These challenges are exacerbated by the generalizability issue of DNNs and the prevalence of long-tail driving scenarios not covered during training and design time (Jiao et al., 2023b; Fu et al., 2024; Ding et al., 2023a; Jiao et al., 2023a).

To this end, researchers and engineers in the AV industry are exploring the potential of Large Language Models (LLMs) (Touvron et al., 2023; OpenAI, 2020; Devlin et al., 2018) for their ability for human interaction, adept reasoning capabilities, and comprehensive knowledge, particularly in handling long-tail driving scenarios (Yang et al., 2023; Fu et al., 2024). Nevertheless, the practical integration of LLMs into the AV software pipeline for safety purposes remains an open question. Therefore, this paper delves into the application of LLMs in autonomous driving from a safety perspective, highlighting its implementation through a couple of illuminating case studies.

From a safety perspective, figure 1 shows the possible integration of LLMs for different modules in the AV software pipeline. As a safety-critical system, we equip the AV with a safety verifier for the proposed control input generated from the software stack with assistance from LLMs. The verifier returns safety-checking results to LLM for in-context safety learning which could affect the outputs from different components in various ways. In this paper, we conduct two case studies to leverage LLM as a behavior-level decision-maker which interacts with a high-level predictor for evaluating the intention and aggressiveness of other agents, and with the low-level trajectory planner and safety verifier. These case studies show that LLM can improve system performance while achieving safety assurance. We hope this paper can provide the AV community with a comprehensive safety standpoint to explore and evaluate the usage of LLM in their AV software stack.

[*]Equal Contribution. Emails: {yixuanwang2024, ruochen.jiao}@u.northwestern.edu

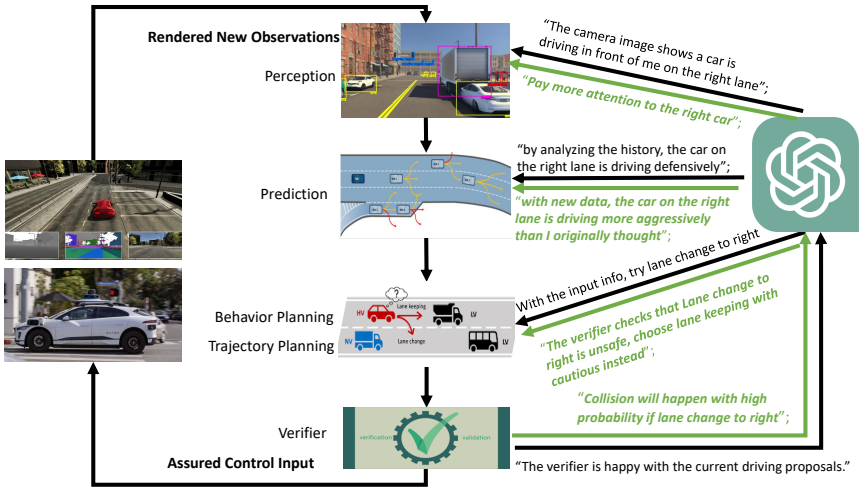


Figure 1: Overview of possible LLM integration for AV with a safety verifier as a shield. Most directly, LLM can make behavior-level decisions such as lane changing by scene understanding via text, which affects the trajectory planning with different safety constraints, as shown in our case studies. The safety verifier checks the safety of the proposed control input from the decision-making and conducts in-context learning if the action is verified to be unsafe, as shown in green arrows. The unsafe feedback can be traced back to the behavior maker, predictor, and perception module as shown. Besides, LLM can assist the perception module in understanding the scene for decision-making better. LLM can also help intention prediction by reading the recent history of the surroundings to better guess their driving habit and intentions (e.g., whether lane changing) for safer decision-making.

This paper is organized as follows. We first introduce related works in Section 2. Section 3 and Section 4 show our proposed designs integrating LLM as an intelligent safety-aware behavioral decision-maker with a safety verifier and an interactive state machine. Section 5 discusses the possible integration of LLM for other components including perception, prediction, and simulation in the AV system for safety purposes. Section 6 concludes the paper.

2 Related Works

The integration of LLMs such as GPT-3 (OpenAI, 2020) into AD has garnered significant attention in recent years, revolutionizing natural language understanding and enhancing the capabilities of self-driving vehicles (Wayve, 2023). The related literature from different perspectives is as follows.

Human-Oriented: One direct application is enabling human-vehicle interaction through natural language. LLMs have been leveraged to interpret, respond, and provide suggestions in natural language to human riders and drivers (Zhang et al., 2023; Wayve, 2023; Xu et al., 2023). These models generate natural language narrations that assist human driving for decision-making and improve the interpretability of AD systems by explaining driving behaviors. Recent works have gone beyond interaction and employed LLMs to learn human driving behaviors and trajectory data through chain-of-thoughts (Wei et al., 2022; Jin et al., 2023b). This approach enables the LLM driver to behave like humans to solve complex driving scenarios and even allows LLMs to function directly as motion planners (Mao et al., 2023).

Perception, Prediction, and Planning (Decision-making): The reasoning, interpretation, memorization, and decision-making abilities of LLMs contribute to solving long-tail corner cases, improving generalizability, and increasing the interpretability of AD systems. Specifically, there is a growing interest in integrating LLMs into the planning (decision-making) module, which significantly improves user trust and generalizes to various driving cases (Jin et al., 2023a). This integration is achieved through fine-tuning pre-trained LLMs (Liu et al., 2023a) or by prompt engineering with chain-of-thought, which usually enable the AD motion planner to process multilabel inputs, e.g., ego-vehicle information, maps, and perception results (Wen et al., 2023; Cui et al., 2023; Fu et al., 2024; Mao et al., 2023). Additionally, researchers are exploring LLMs in the perception module to

enable self-aware perception, and fast and efficient adaptation to changing driving environments, including tracking, detection, and prediction (Malla et al., 2023; Radford et al., 2021; Wu et al., 2023; Ding et al., 2023b). Zhou et al summarize the state-of-the-art works in this field (Zhou et al., 2023).

Nevertheless, the aforementioned references fail to address safety concerns associated with LLM in AD. We prioritize safety under the context of LLM, a perspective evident in our case studies. We allow LLM decisions to directly formulate safety constraints for low-level Model Predictive Control (MPC) under prediction uncertainties. Our case studies align closely with the LanguageMPC (Sha et al., 2023), where the authors also employ LLMs as a decision-maker for AD. They convert LLM decisions into the mathematical representations needed for the low-level controllers, MPC, through guided parameter matrix adaptation. However, LanguageMPC has not been extensively validated in complex driving environments. Additionally, it does not consider uncertainty from predictions nor include safety analysis or optimization in its methodology.

Generation and Simulation: LLMs' generative capabilities have facilitated the acquisition of complex driving data samples, which were previously difficult to gather due to certain environmental constraints. The diffusion model, a method that has recently reached significant success in the text-to-image domain, has become increasingly popular (Sohl-Dickstein et al., 2015; Ho et al., 2020). Some efforts have been put into the area of generating the driving scenarios using diffusion models (Li et al., 2023; Gao et al., 2023; Wang et al., 2023a; Hu et al., 2023; Zhong et al., 2023).

Our work is related to the safety verification for ML-based autonomous systems where AD systems are representative. Safety verification, in general, can be categorized into two groups: 1) explicit reachable set computation (Wang et al., 2023d; Huang et al., 2022; Ivanov et al., 2021; Kochdumper et al., 2023; Goubault & Putot, 2022; Schilling et al., 2022; Huang et al., 2019) and 2) inexplicit reachable set evaluation, such as barrier certificate (Prajna, 2006; Wang et al., 2023b), control barrier function (Ames et al., 2019; Yang et al., 2022), forward invariance (Wang et al., 2020; Chen et al., 2018), etc. There have been emerging works for integrating verification modules into the control learning or reinforcement learning for safety-assured autonomy (Dawson et al., 2022; Wang et al., 2023c;b; Zhan et al., 2023; Jin et al., 2020). Our paper follows a similar idea where we develop the safety verifier as a shield for the LLM decision-maker to generate safe actions.

3 LLM Conditioned Adaptive MPC for Trajectory Planning with Safety Assurance

Here we conduct a case study for LLM as a behavior planner via prompt engineering, as shown in Figure 2. Next, we introduce the components of this case study as follows.

Environment and System: Given the safety cost of driving, we primarily focus on a simulated highway-driving environment by using HighwayEnv Leurent (2018). As shown in Figure 4, we consider a one-way three-lane driving scenario. We assume that the vehicle dynamics is known and available to MPC, which can be expressed as $s_{t+1} = f(s_t, u_t)$ where $s = (x, y, v_x, v_y) \in \mathcal{S} \subset \mathbb{R}^4$ with x, y, v_x, v_y denote longitudinal position, lateral position, longitudinal speed, and lateral speed, respectively. The continuous control input to the ego vehicle $u_t \in \mathcal{U} \subset \mathbb{R}^2$ includes acceleration and steering signal. $f : \mathcal{S} \times \mathcal{U} \to \mathcal{S}$ denotes the bicycle model dynamics (Jiao et al., 2023c).

Input and Output of LLM: We call OpenAI GPT-4 API as our LLM driver agent. We input a template-generated text description of the surroundings within a specific perception range including their relative position (such as "the car i is driving in front of the ego on the right lane" or "the car i is driving behind the ego in the middle lane"), their relative speed (such as "the car i is driving faster/slower than the ego"), the estimation of time to the collision to other agents (relative distance / relative speed), along with other vehicle's intention predictions. The output of the LLM decision maker is constrained to select a target lane for lower level MPC (such as "Middle Lane, Left Lane, Right Lane") with the reasoning. Every decision made by LLM will have 5 consecutive control steps.

Prediction Module: The prediction module on the AV predicts the future state $\hat{s}_t^j$ of surrounding car j at time step t. To be realistic and considering uncertainties, we assume the predicted position results are intervals on a specific time step, i.e, instead of $\hat{x}_t^j, \hat{y}_t^j$, we now have $[\underline{\hat{x}}_t^j, \bar{\hat{x}}_t^j]$ and $[\underline{\hat{y}}_t^j, \bar{\hat{y}}_t^j]$. We assume the position intervals contain the ground truth x_t^j, y_t^j of the surroundings in the future, i.e., $x_t^j \in [\underline{\hat{x}}_t^j, \bar{\hat{x}}_t^j]$, $y_t^j \in [\underline{\hat{y}}_t^j, \bar{\hat{y}}_t^j]$. Because of the receding horizon nature of MPC, we need to call

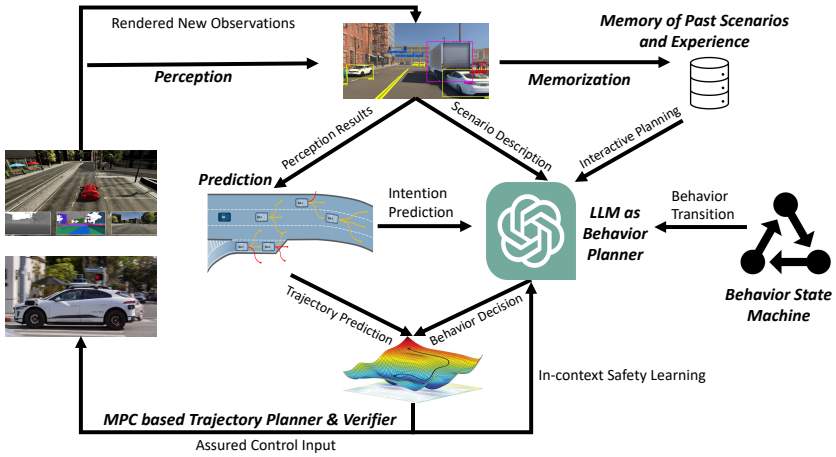


Figure 2: This framework shows LLM as a behavior planner that provides safety constraints for a low-level MPC trajectory planner. The LLM driver takes high-level intention prediction, scenario description, behavior state machine, and its memory via text generated by a template and makes a behavior decision based on its understanding of the driving scene. LLM decisions will formulate safety constraints for low-level MPC-based trajectory planning. Serving as a verifier, the feasibility of the MPC problem will be sent back to LLM to (re)-evaluate its decision for in-context safety learning.

the prediction module to get the prediction results for safety constraint formulation in MPC. Before introducing our LLM-conditioned MPC, we first show that a **naive MPC formulation** of trajectory planning as

$$\begin{aligned} &\min_{u_t,u_{t+1},\cdots,u_{t+k}} -x_{t+k} + \sum_{i=t}^{t+k-1} ||u_{i+1} - u_i||_2, \\ \text{s.t.,}\ \ &s_{i+1} = f(s_i, u_i), \forall i \in [t, t+k], \quad y_{inf} \leq y_i \leq y_{sup}, \forall i \in [t, t+k] \quad \text{(Road boundary)}, \\ &|x_i - \underline{\hat{x}}_i^j| - L \geq 0, |x_i - \bar{\hat{x}}_i^j| - L \geq 0, \text{where Lane}([\underline{\hat{y}}_i^j, \bar{\hat{y}}_i^j]) == \text{Lane}(y_i) \quad \text{(Safety)} \end{aligned} \tag{1}$$

where $\text{Lane}(y) \in 0, 1, 2$ is an indicator function that determines which lane the car is driving on by its lateral position y, specifically $0, 1, 2$ denotes "Left", "Middle", and "Right". The objective function aims to maximize the performance (longitudinal position or speed) with minimal control jerks.

LLM Conditioned Adaptive MPC for Trajectory Planning: To reduce the complexity, we leverage the reasoning ability and common sense knowledge of LLM to decide which lane to drive for the MPC, by providing the scene text description to LLM and ask for a decision that relaxes the constraints in MPC. Specifically, at time step t, our **LLM conditioned MPC** tries to solve the following optimization problem

$$\begin{aligned} &\min_{u_t,u_{t+1},\cdots,u_{t+k}} -x_{t+k} + \sum_{i=t}^{t+k-1} ||u_{i+1} - u_i||_2, \\ \text{s.t.,}\ \ &s_{i+1} = f(s_i, u_i), \forall i \in [t, t+k], \quad y_{inf} \leq y_i \leq y_{sup}, \forall i \in [t, t+k] \quad \text{(Road boundary)} \\ &\textbf{Lane}(y_i) = \textbf{Lane(LLM)} \quad \textbf{(Behavior provided by LLM)} \\ &|x_i - \underline{\hat{x}}_i^j| - L \geq 0, |x_i - \bar{\hat{x}}_i^j| - L \geq 0, \text{where Lane}([\underline{\hat{y}}_i^j, \bar{\hat{y}}_i^j]) == \textbf{Lane(LLM)} \quad \text{(Safety)} \end{aligned} \tag{2}$$

The problem 1 is harder to solve than problem 2. The increased complexity originates from the constraint $\text{Lane}([\underline{\hat{y}}_i^j, \bar{\hat{y}}_i^j]) == \text{Lane}(y_i)$, where $\text{Lane}(y_i)$ is undetermined and can choose from $\{0, 1, 2\}$. Therefore problem 1 is a mixed integer nonlinear programming problem. *In practice, this problem is often infeasible, which is also observed in our case studies.* With the decision from LLM by its knowledge, we remove the integer decision variable in problem 2 and thus it is easier to solve. Our approach shares a similar philosophy of hierarchical MPC as introduced in (Huang et al., 2016) where we decompose a hard trajectory planning into a two-phase problem that is easier to solve.

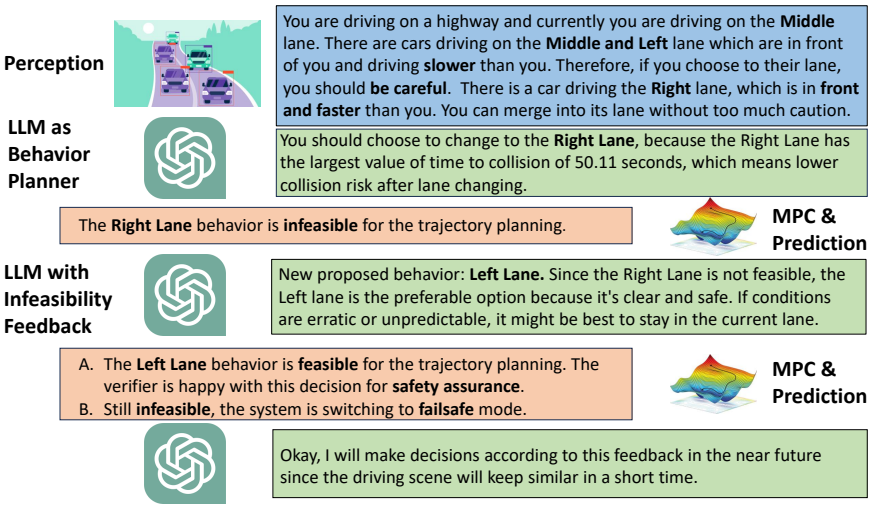
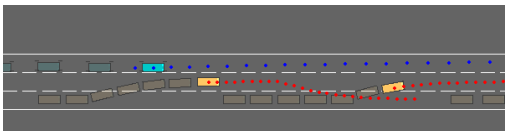
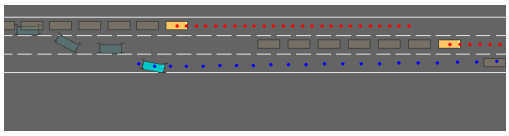


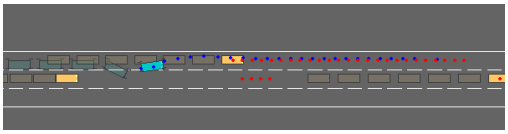
Figure 3: In-context safety learning for LLM with the feedback from MPC for trajectory planning.



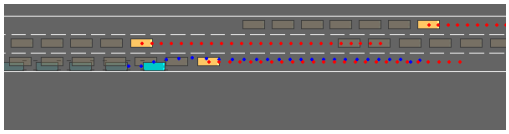
(a) **Lane Keeping:** The LLM decides to keep the current lane because it is clear and safe. The MPC maintains the highest speed for trajectory planning.



(b) **Lane Change:** LLM decides to change to the rightmost lane from the leftmost lane because the target lane has more space with minimal safety risk.



(c) **Aborted Lane Change:** We discover that LLM can abort its lane-changing if the MPC is infeasible during lane changing to reduce the collision risk.



(d) **Failsafe:** The failsafe mode keeps the current lane and maintains a minimal distance for safety. It is only used if LLM's decisions are infeasible in MPC.

Figure 4: The ego car is in blue and other agents are in yellow. The blue dots are the planned trajectory waypoints of the ego. The red dots are the sampled waypoints of other agents from the interval-based prediction. The grey rectangles are the recent trajectory histories of the ego and other agents. The LLM exhibits safe lane keeping, optimistic lane changing, cautious lane changing abort, and conservative failsafe in the simulations.

In-context Safety Learning with Verifier: For safety purposes, control input to the ego vehicle has to go through a verifier for safety checking and provide the verification result back to the LLM to reevaluate the behavior decision. In general, the verifier could be in the form of reachability analysis (Wang et al., 2023d), barrier theory, etc (Wang et al., 2023c), as we detailed in the related work. In this case study, we use the *feasibility of the LLM-conditioned MPC 2 as the safety verifier*. If the MPC is feasible which means there exists a safe control signal, we then feedback "the verifier is happy with the proposed **Lane**" to LLM. Otherwise, infeasible MPC indicates potential collisions which we feedback to LLM to reevaluate and regenerate another behavior, as shown in Figure 3.

Failsafe Mode: It is possible that regenerated behavior or all behaviors are still infeasible for the low-level MPC and thus safety cannot be assured. In this case, we design the AV system switch to a failsafe mode, to keep the current lane and apply a (possibly hard) brake to keep a minimal distance from the front leading car as $-\frac{v_e^2 - v_s^2}{2(\underline{x_l} - x_e - \epsilon)}$ where v_e, v_s are the ego and leading velocity, $\underline{x_l}$ is the lower bound of the estimation for the leading car's location and x_e is the ego position, $\epsilon > 0$. This failsafe optimistically disregards collision with the following car as the ego is optimized to driving faster than the rest IDM-based cars. To be more conservative, one can consider the following car.

Experiments Analysis: We compare our approach with the state-of-the-art open-source DriveLikeAHuman (Fu et al., 2024) because it also testfies in the HighwayEnv simulator. We add the same interval-based prediction uncertainty to the DriveLikeAHuman framework and adapt its heuristic safety rule considering the interval uncertainty for a fair comparison. We simulate 300 control steps in one test episode. The maximum velocity is set to $40\ m/s$. We run 5 trials/episodes for each method and record their results as in Table 1.

Table 1: Comparison results of the case study with 5 episodes.

	Safety	Velocity(m/s)	Latency(s)
Ours	✓	**34.3**($\pm$7.7)	**1.7**($\pm$2.7)
DriveLikeAHuman	×	31.9($\pm$5.1)	55.5($\pm$15.2)

- *Safety*: No collision happened in our simulations with 1500 total control steps and more than 300 LLM decision-makings (each decision made by LLM is followed by 5 consecutive control steps). Except for an LLM calling error in one trial, DriveLikeAHuman has collisions in 4 trials around 30th $\sim$ 50th steps. This is because it uses a low-level PID control with a naive high-level heuristic safety rule that does not consider vehicle dynamics and constraints for safety checking.
- *Average Velocity*: We measure the longitudinal speed average and standard deviation as performance metrics. The ego drives faster with our approach. This is because we maximize the longitudinal location (speed) in the objective function of our LLM-conditioned MPC.
- *Latency*: The latency of our approach includes the OpenAI API call every 5 control steps and the timing of solving MPC every step while the baseline spends most of the time on the chain-of-thought process with the API per control step. Although both latency are not realistic for real-world driving, ours is significantly shorter than the baseline's.

4 LLM AS INTERACTIVE DECISION MAKER: INTERACTIVE PLANNING BY BEHAVIOR PREDICTION AND STATE MACHINE

As with most existing works on LLM for AD, our previous case study focuses on one-step planning or single-frame decision-making. We can further improve the performance and safety of LLM for driving tasks by explicitly considering the ego vehicle's high-level behavior transitions and the interaction with surrounding agents in multiple consecutive steps. In Figure 5, besides the MPC verifier we proposed previously, we further design the state machine framework as behavior transition guidance, the memory module for intention prediction, and the reflection module for behavior-level safety checks and in-context learning. We will explain them in detail in the following.

State Machine as Behavior Guidance for LLM Decision Maker: In practical driving situations, executing maneuvers like lane changes or unprotected left turns involves several steps. Vehicles must adhere to established rules or patterns to interact effectively with nearby traffic and accomplish these tasks. This necessitates that LLMs conform to specific *behavioral constraints*. We propose to integrate a state machine into the LLM through in-context learning. This approach aims to ensure adherence to these essential regulations and established patterns while maintaining sufficient flexibility in decision-making.

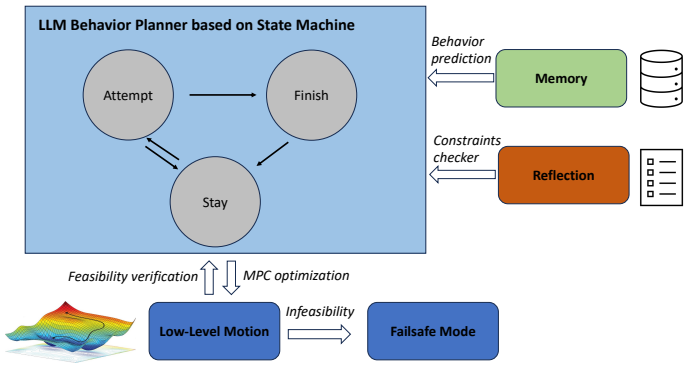


Figure 5: Interactive multi-step decision-making based on the behavior state machine and intention prediction. The memory will store the previous observations of other agents and the reflection module will check hard constraints including state transition rules and minimum safety requirements.

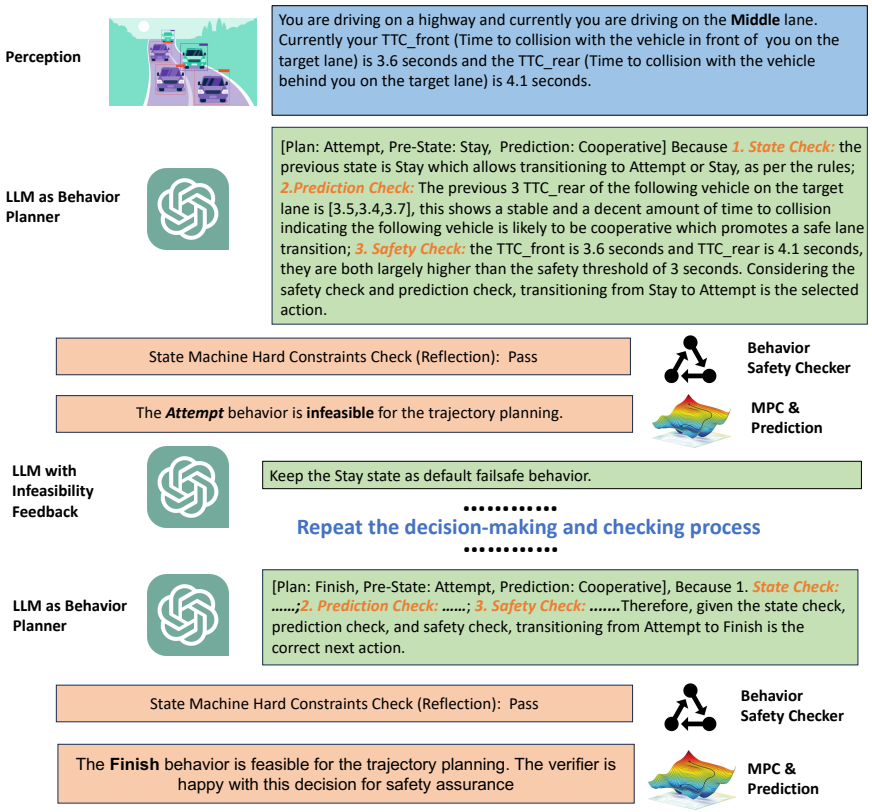


Figure 6: An example of our proposed safe interactive decision-making pipeline for lane changing. With the state machine design and behavior level prediction, the LLM-powered agent can make explainable and safe decisions continually and interactively in complex scenarios. In each cycle, the LLM will reason its decision by three behavior-level checks (state, prediction, and safety). The reflection module will provide feedback for failsafe plans and in-context learning if LLM makes severe and obvious mistakes. The low-level MPC is in charge of the safety verification and execution.

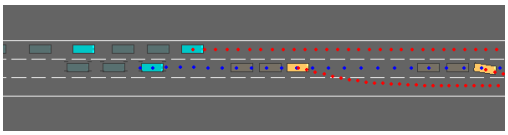
The **state-machine-conditioned LLM** can enhance the safety and interpretability of the decision-making process in several aspects. First, the state machine can constrain the decision space and simplify the dependency among time-series decisions. Humans can trust the decision-making pipeline if the LLM ensures the transition between states is safe. Second, we add some intermediate/interactive states into the state machine design to help the LLM better understand other vehicles' behavior. During these states, the LLM can proactively interact with the surrounding vehicle (e.g. the following vehicle on the target lane when changing lanes) but still ensure safety.

In Figure 5, we present our pipeline for interactive lane changing using LLM as the decision-maker. The framework is centered around the state machine which defines the basic behavior pattern of our LLM. The memory stores important past information about surrounding vehicles, helping the LLM make predictions of their intentions. The reflection module is to monitor the LLM and make sure the transition is valid from state to state and to give feedback to the LLM for in-context learning when the LLM violates hard transition constraints. The LLM determines transitions based on predefined rules and inferred information. The transition involves several checks:

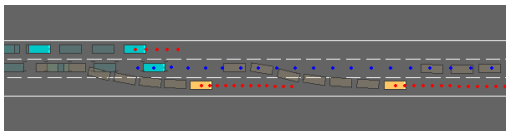
State Check: The selected state must be valid as per a predefined state machine graph.

Safety Check: The LLM evaluates the possibility of collision if it takes certain actions transiting to the next state. In this particular lane-changing example, the time-to-collision (TTC) is applied to ensure the proposed state won't lead to a collision. The LLM will compare the TTC against a set threshold.

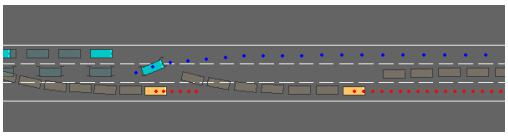
Prediction Check: The LLM predicts the intentions of nearby vehicles based on their historical behaviors in past multiple frames in the memory modules. If the LLM deems a surrounding vehicle



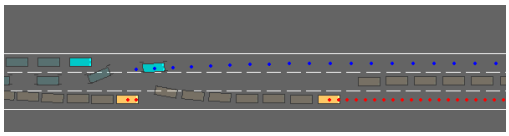
(a) **Phase 1.** The ego (blue, middle) by LLM aims to cut into the left lane. The LLM notices there isn't enough space for a safe lane change, picks the "Stay", and accelerates to pass the blue car in front.



(b) **Phase 2.** Ego vehicle (LLM) is passing the blue car in front and now it only needs to consider and interact with the leading vehicle on the target left lane. LLM decides to continue in the "Stay" in this cycle.



(c) **Phase 3.** Ego vehicle (LLM) passed the blue car. LLM decides to transit the behavior state to "Attempt" given the comprehensive reasoning including prediction, state transition check and safety analysis. In the state "Attempt", the ego vehicle moves to the middle of two lanes and further observes the reaction of the following vehicle on the target lane.



(d) **Phase 4.** During the state "Attempt", the LLM predicts the following vehicle as a cooperative agent and updates the collision time estimation for safety analysis. LLM decides to transit to state "Finish" given all the analysis and the feedback from the reflector and MPC. Finally, it is moving to the target lane safely.

Figure 7: The ego car is in blue on the middle lane, aiming to cut into the left lane. It interacts with two other blue vehicles in the left lane. The blue dots are the planned trajectory waypoints of the ego. The red dots are the sampled waypoints of other agents from the interval-based prediction. The grey rectangles are the recent trajectory histories of the ego and other agents. The LLM exhibits safe interactive lane-changing behaviors in the multiple-step decision-making process.

too aggressive or uncooperative, it's unsafe to proceed with the maneuver. The LLM can interact with the surrounding vehicles in different manners given their different predicted behavior patterns.

Reflection Module: State and safety checks are stringent requirements in the decision-making process. To ensure compliance with these requirements, a reflection module monitors state transitions. This module corrects and provides feedback to the LLM, facilitating in-context learning, especially when decisions breach these strict constraints. For behavior prediction, the reflection module enforces no constraints to the intention estimation - the LLM independently and flexibly assesses the intentions of surrounding vehicles, categorizing them as either aggressive or cooperative.

Intention Prediction Module: Unlike the prediction for MPC, the intention prediction is to estimate the high-level behavior patterns of the surrounding vehicle, which doesn't need to be very detailed but is important for interaction. We define the potential intention of surrounding agents as cooperative and aggressive. We use the time-to-collision (TTC) of surrounding vehicles as input to the LLM for prediction. At every planning step, the LLM decision-maker will extract the surrounding vehicles' TTCs with the past 3 steps and predict their corresponding intentions. We give several human-labeled demonstrations when setting up the LLM.

Experimental Analysis: In this study, we evaluate our proposed framework using the HighwayEnv simulation platform. As depicted in Figure 6, our framework successfully guides the LLM to perform safe motion planning in sequential steps, relying on a state machine, along with prediction and reflection modules. Figure 7 visualizes the lane-changing scenario, showing the LLM's continuous reasoning and interaction with nearby vehicles under complex conditions. This figure also details the state transitions within the decision-making process. we compared our approach with the open-source DriveLikeAHu-

Table 2: Experimental results for lane changing collision rate and success rate with 17 episodes.

	Collision Rate	Success Rate
DriveLikeAHuman	47.1%	41.2%
Ours	**0**	**100%**
Ours w/o failsafe	23.1%	76.9%
Ours w/o reflection	**0**	92.3%

man Fu et al. (2024) framework in terms of safety (collision rate) and the success rate of lane changes. The findings, presented in Table 2, indicate a significantly higher rate of collisions and aborts with the DriveLikeAHuman's naive chain-of-thoughts strategy. In contrast, our method not only ensures safety but also exhibits a remarkable success rate in a variety of generated scenarios, highlighting the efficacy and generalizability of our bi-level interactive planning framework. The final two columns of Table 2 showcase the significance of our framework's components through an ablation study.

5 Discussion: LLM as Other Roles for Safety

We discuss the possible usage of LLMs for other components in the AD software pipeline, as shown in Figure 1. We directly ask ChatGPT-3.5 (e.g., prompt as "How can a large language model assist the perception module for safer autonomous driving?") and summarize its responses below.

LLMs for Perception. *1) Multimodal Fusion*: It is possible to consider multimodal infusion with language input. By integrating information from both sensor data and language input, the perception module can create a more comprehensive understanding of the environment. This multimodal fusion enables the system to make more informed safer decisions by considering both visual information and contextual cues provided by natural language. *2) Semantic Object Recognition*: LLMs can assist in recognizing and understanding objects in the environment based on their semantic context of safety. For instance, if a passenger says, "Watch out for the cyclist ahead", LLMs can understand this information to prioritize and adapt the behavior accordingly, enhancing safety. *3) Adaptive Object Detection*: LLMs can provide information that helps the perception module adapt its object detection algorithms based on specific scenarios. For example, if LLMs understand that the vehicle is in a construction zone, they can convey this information to the perception module, prompting the system to be more cautious and attentive to potential hazards.

LLMs for Prediction. *1) Natural Language Inputs for Contextual Awareness*: The language model in the prediction module can process natural language inputs (possibly from perception) to understand and infer the potential intentions of other drivers. For example, if the perception model or human user interprets "heavy traffic ahead," the prediction module with LLMs can understand it and adjust its expectations and predictions accordingly for safer operation. *2) Human-Centric Predictions*: Language understanding can help the prediction module make more human-centric predictions by considering factors such as hand gestures, turn signals, or spoken commands from other drivers. This allows the autonomous vehicle to anticipate and respond to human behaviors more effectively, improving AV safety. *3) Behavioral Evaluation*: The language model can assist in evaluating the driving behaviors and aggressiveness of surrounding cars. This helps the prediction module adjust its predictions based on the perceived driving styles of other vehicles.

LLMs for Simulation. *1) User Specific Scenario Generation and Variation*: The language model can generate natural language descriptions of diverse driving scenarios by user input for safety concerns, allowing the simulation module to create a wide range of realistic and challenging situations for testing and training in a safety perspective. This helps in ensuring that the autonomous system is well-prepared for various real-world conditions. *2) Human-Like Interaction*: The language model can simulate human-like interactions by generating realistic communication between simulated drivers, pedestrians, and other entities. This enhances the realism of the simulation, allowing the autonomous system to practice responding to natural language cues and gestures for safety purposes. *3) Simulation Annotation and Analysis*: The language model can assist in annotating simulation data by generating descriptions or labels for different events and entities, which further the AV development.

6 Conclusion

In conclusion, our presented framework explores the integration of an LLM as an intelligent decision-maker for autonomous driving, fortified by a safety verifier feedback for in-context safety learning. Through two case studies, we demonstrate the efficacy of our approach, showcasing notable enhancements in both performance and safety. We further discuss the potential usage of the LLM for other components. This paper intends to broaden the safety perspective within the autonomous driving community concerning the utilization of LLMs. The future directions and remaining challenges include testing this framework in real-world environment and handling ambiguity, biases, and inconsistencies in LLM outputs.

REFERENCES

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pp. 3420–3431. IEEE, 2019.

Yuxiao Chen, Huei Peng, Jessy Grizzle, and Necmiye Ozay. Data-driven computation of minimal robust control invariant set. In *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4052–4058. IEEE, 2018.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles. *arXiv preprint arXiv:2309.10228*, 2023.

Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, pp. 1724–1735. PMLR, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Wenhao Ding, Chejian Xu, Mansur Arief, Haohong Lin, Bo Li, and Ding Zhao. A survey on safety-critical driving scenario generation—a methodological perspective. *IEEE Transactions on Intelligent Transportation Systems*, 2023a.

Xinpeng Ding, Jianhua Han, Hang Xu, Wei Zhang, and Xiaomeng Li. Hilm-d: Towards high-resolution understanding in multimodal large language models for autonomous driving. *arXiv preprint arXiv:2309.05186*, 2023b.

Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 910–919, 2024.

Ruiyuan Gao, Kai Chen, Enze Xie, Lanqing Hong, Zhenguo Li, Dit-Yan Yeung, and Qiang Xu. Magicdrive: Street view generation with diverse 3d geometry control. *arXiv preprint arXiv:2310.02601*, 2023.

Eric Goubault and Sylvie Putot. Rino: robust inner and outer approximated reachability of neural networks controlled systems. In *International Conference on Computer Aided Verification*, pp. 511–523. Springer, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Chao Huang, Xin Chen, Yifan Zhang, Shengchao Qin, Yifeng Zeng, and Xuandong Li. Hierarchical model predictive control for multi-robot navigation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 3140–3146, 2016.

Chao Huang, Jiameng Fan, Wenchao Li, Xin Chen, and Qi Zhu. Reachnn: Reachability analysis of neural-network controlled systems. *ACM Transactions on Embedded Computing Systems (TECS)*, 18(5s):1–22, 2019.

Chao Huang, Jiameng Fan, Xin Chen, Wenchao Li, and Qi Zhu. Polar: A polynomial arithmetic framework for verifying neural-network controlled systems. In *International Symposium on Automated Technology for Verification and Analysis*, pp. 414–430. Springer, 2022.

Radoslav Ivanov, Taylor Carpenter, James Weimer, Rajeev Alur, George Pappas, and Insup Lee. Verisig 2.0: Verification of neural network controllers using taylor model preconditioning. In *International Conference on Computer Aided Verification*, pp. 249–262. Springer, 2021.

Ruochen Jiao, Xiangguo Liu, Bowen Zheng, Dave Liang, and Qi Zhu. Tae: A semi-supervised controllable behavior-aware trajectory generator and predictor. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12534–12541. IEEE, 2022.

Ruochen Jiao, Juyang Bai, Xiangguo Liu, Takami Sato, Xiaowei Yuan, Qi Alfred Chen, and Qi Zhu. Learning representation for anomaly detection of vehicle trajectories. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9699–9706. IEEE, 2023a.

Ruochen Jiao, Xiangguo Liu, Takami Sato, Qi Alfred Chen, and Qi Zhu. Semi-supervised semantics-guided adversarial training for robust trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8207–8217, 2023b.

Ruochen Jiao, Yixuan Wang, Xiangguo Liu, Chao Huang, and Qi Zhu. Kinematics-aware trajectory generation and prediction with latent stochastic differential modeling. *arXiv preprint arXiv:2309.09317*, 2023c.

Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. Adapt: Action-aware driving caption transformer. *arXiv preprint arXiv:2302.00673*, 2023a.

Wanxin Jin, Zhaoran Wang, Zhuoran Yang, and Shaoshuai Mou. Neural certificates for safe control policies. *arXiv preprint arXiv:2006.08465*, 2020.

Ye Jin, Xiaoxi Shen, Huiling Peng, Xiaoan Liu, Jingli Qin, Jiayang Li, Jintao Xie, Peizhong Gao, Guyue Zhou, and Jiangtao Gong. Surrealdriver: Designing generative driver agent simulation framework in urban contexts based on large language model. *arXiv preprint arXiv:2309.13193*, 2023b.

Niklas Kochdumper, Hanna Krasowski, Xiao Wang, Stanley Bak, and Matthias Althoff. Provably safe reinforcement learning via action projection using reachability analysis and polynomial zonotopes. *IEEE Open Journal of Control Systems*, 2:79–92, 2023.

Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

Xiaofan Li, Yifu Zhang, and Xiaoqing Ye. Drivingdiffusion: Layout-guided multi-view driving scene video generation with latent diffusion model. *arXiv preprint arXiv:2310.07771*, 2023.

Jiaqi Liu, Peng Hang, Jianqiang Wang, Jian Sun, et al. Mtd-gpt: A multi-task decision-making gpt model for autonomous driving at unsignalized intersections. *arXiv preprint arXiv:2307.16118*, 2023a.

Xiangguo Liu, Ruochen Jiao, Yixuan Wang, Yimin Han, Bowen Zheng, and Qi Zhu. Safety-assured speculative planning with adaptive prediction. *arXiv preprint arXiv:2307.11876*, 2023b.

Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. Drama: Joint risk localization and captioning in driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1043–1052, 2023.

Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Bev-guided multi-modality fusion for driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21960–21969, 2023.

Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2980–2987. IEEE, 2023.

OpenAI. Chatgpt-3: Language model for conversational agents. `https://www.openai.com/`, 2020.

Stephen Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Christian Schilling, Marcelo Forets, and Sebastián Guadalupe. Verification of neural-network control systems by integrating taylor models and zonotopes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8169–8177, 2022.

Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, and Jiwen Lu. Drivedreamer: Towards real-world-driven world models for autonomous driving. *arXiv preprint arXiv:2309.09777*, 2023a.

Yixuan Wang, Chao Huang, and Qi Zhu. Energy-efficient control adaptation with safety guarantees for learning-enabled cyber-physical systems. In *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1–9, 2020.

Yixuan Wang, Simon Zhan, Zhilu Wang, Chao Huang, Zhaoran Wang, Zhuoran Yang, and Qi Zhu. Joint differentiable optimization and verification for certified reinforcement learning. In *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, pp. 132–141, 2023b.

Yixuan Wang, Simon Sinong Zhan, Ruochen Jiao, Zhilu Wang, Wanxin Jin, Zhuoran Yang, Zhaoran Wang, Chao Huang, and Qi Zhu. Enforcing hard constraints with soft barriers: Safe reinforcement learning in unknown stochastic environments. In *International Conference on Machine Learning*, pp. 36593–36604. PMLR, 2023c.

Yixuan Wang, Weichao Zhou, Jiameng Fan, Zhilu Wang, Jiajun Li, Xin Chen, Chao Huang, Wenchao Li, and Qi Zhu. Polar-express: Efficient and precise formal reachability analysis of neural-network controlled systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023d.

Wayve. Lingo: Natural language for autonomous driving, 2023. URL `https://wayve.ai/thinking/lingo-natural-language-autonomous-driving/`. Accessed: [Insert Access Date].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*, 2023.

Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. *arXiv preprint arXiv:2309.04379*, 2023.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023.

Shuo Yang, Shaoru Chen, Victor M Preciado, and Rahul Mangharam. Differentiable safe controller design through control barrier functions. *IEEE Control Systems Letters*, 7:1207–1212, 2022.

Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023.

Simon Sinong Zhan, Yixuan Wang, Qingyuan Wu, Ruochen Jiao, Chao Huang, and Qi Zhu. State-wise safe reinforcement learning with pixel observations. *arXiv preprint arXiv:2311.02227*, 2023.

Siyao Zhang, Daocheng Fu, Zhao Zhang, Bin Yu, and Pinlong Cai. Trafficgpt: Viewing, processing and interacting with traffic foundation models. *arXiv preprint arXiv:2309.06719*, 2023.

Ziyuan Zhong, Davis Rempe, Yuxiao Chen, Boris Ivanovic, Yulong Cao, Danfei Xu, Marco Pavone, and Baishakhi Ray. Language-guided traffic simulation via scene-level diffusion. *arXiv preprint arXiv:2306.06344*, 2023.

Xingcheng Zhou, Mingyu Liu, Bare Luka Zagar, Ekim Yurtsever, and Alois C Knoll. Vision language models in autonomous driving and intelligent transportation systems. *arXiv preprint arXiv:2310.14414*, 2023.