Applied Soft Computing Journal 183 (2025) 113625

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

DAPlanner: Dual-agent framework with multi-modal large language model for autonomous driving motion planning

Pin Zhang [a,f], Ke Lin [b], Duantengchuan Li [c,*], Zixun Fu [d], Yuefeng Cai [c], Bing Li [a,*], Huan Yu [e,f], Ming Li [a]

[a] School of Computer Science, Wuhan University, Wuhan 430072, China
[b] Department of Control Science and Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China
[c] School of Information Management, Wuhan University, Wuhan 430072, China
[d] Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China
[e] School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China
[f] Dongfeng Yuexiang Technology Co., Ltd., China

ARTICLE INFO

Keywords:
Autonomous driving
Motion planning
Large language model
BEV

ABSTRACT

In autonomous driving, motion planning is a key task for enabling vehicles to make autonomous decisions and take action. The goal of Motion Planning is to design an optimal trajectory from the current to the target state, considering other traffic participants, road conditions, and driving tasks, ensuring both safety and comfort. Current planning methods are ineffective in integrating multiple features from the perception layer and increase the risk of collisions due to the lack of trajectory constraints. Therefore, we propose a Dual-Agent Framework with a Multi-modal Large Language Model for Autonomous Driving Motion Planning. Specifically, we convert the surrounding traffic environment into Frenet Space, establishing spatial relationships between traffic participants and the map. The transformed information is represented in a bird's-eye view (BEV), supplemented by textual prompts as inputs. Finally, we use trajectory generation agents and trajectory discrimination agents based on large language models to impose constraints on the generated trajectories. Additionally, we propose a data augmentation method for motion planning to further enhance system performance. Through evaluation on the large-scale nuScenes dataset, the proposed dual-agent framework based on large language models outperforms existing methods on various metrics. Our code is publicly available at https://github.com/dacilab/DAPlanner.

1. Introduction

The development of autonomous driving technology has brought revolutionary changes to the transportation industry [1], profoundly altering the way people travel. In autonomous driving systems, motion planning, as a core component, plays a critical role in comprehending complex traffic environments and providing rational behavioral decisions and trajectory plans for autonomous vehicles. This process not only ensures the safety of vehicle operation but also improves the comfort of the ride [2–4].

Traditional rule-based motion planning methods rely on predefined rule sets and logical conditions, ensuring the controllability and consistency of system behavior through explicit norms [5–7]. Although this method offers a certain level of interpretability, its effectiveness is largely limited by the completeness of the rule set and the complexity of the environment. Moreover, relying on manually crafted rules often fails to cover all possible traffic scenarios. With the application of deep learning technologies in fields such as computer vision [8,9], natural language processing [10,11], graph representation learning [12,13], and reinforcement learning [14–16], new solutions for motion planning have emerged. Learning-based motion planning methods, through the analysis and learning of large volumes of traffic data, enable more flexible decision-making and planning in complex traffic environments [17–20][21]. However, these methods are typically considered "black-box" approaches in motion planning, meaning their decision-making processes lack transparency and interpretability, and they still face significant challenges when encountering rare or unseen scenarios.

In recent years, with large language models (LLMs) demonstrating human-level abilities in analysis and reasoning, their application in various fields has attracted widespread attention [22–25]. In autonomous driving, several studies have explored the integration of large language

* Corresponding authors.
E-mail addresses: pzhang1992@whu.edu.cn (P. Zhang), dtclee1222@whu.edu.cn (D. Li), bingli@whu.edu.cn (B. Li).

https://doi.org/10.1016/j.asoc.2025.113625
Received 27 January 2025; Received in revised form 8 June 2025; Accepted 12 July 2025
Available online 28 July 2025

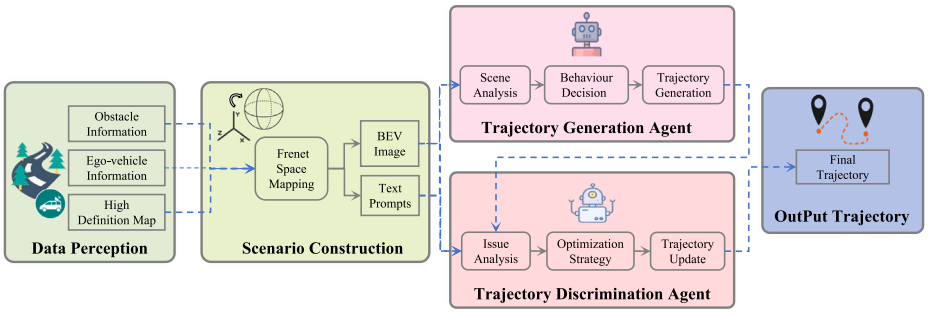

Fig. 1. The paradigm of Dual-Agent Planner, including: a Scenario Construction that constructs BEV image-text pairs through Frenet Space. A Trajectory Generation Agent with Scene Analysis, Behavior Decision and Trajectory Generation. A Trajectory Discrimination Agent analyzes the issue, proposes optimization strategies and updates a better trajectory.

models into motion planning to improve the system's decision-making and planning capabilities [26–29]. Nevertheless, these methods still face some challenges and limitations:

(1) Current models struggle to effectively learn the relationship between the vehicles and the road topology, especially in complex scenarios such as intersections or curves, where the generated trajectories often violate road boundaries, posing significant safety risks.
(2) The planning module of autonomous driving systems receives diverse and complex input information, including map details, vehicle status, and various attributes of other traffic participants. However, many critical scene details are not adequately described or effectively integrated into the LLM, reducing the efficiency of information utilization and negatively impacting the accuracy and rationality of the planning outcomes [27].
(3) Trajectory points generated directly by the LLM are inherently uncertain, which may lead to paths that conflict with human driving patterns or create creating collision risks with other traffic participants, undermining the rationality and safety of autonomous driving behavior.

To address the above challenges, we propose a dual-agent framework for autonomous driving motion planning, based on a multi-modal large language model, as illustrated in Fig. 1. The framework converts information of the map, the vehicle, and other traffic participants into Frenet Space, removing the interference of road shapes on model learning and enabling the model to better capture lane topology features. Furthermore, we combine these diverse inputs into a multi-layered BEV, retaining rich scene information, especially those details that are difficult to convey through text. To further address the uncertainty in trajectories generated directly by large models, we design two agents: the trajectory generation agent, which analyzes the scene and identifies key obstacles that significantly affect the vehicle, enabling behavioral decisions and trajectory planning; and the trajectory discrimination agent, which conducts anomaly analysis on the generated trajectory based on the output of the trajectory generation agent and the scene information. In addition, we employ the reflective thinking chain designed in this paper to identify potential issues, analyze their causes, and propose improvement solutions. Through the collaboration of these two agents in a shared environment, the framework ensures that the final trajectory generated by the model is fully guaranteed in terms of rationality and safety.

In summary, our contributions can be encapsulated as follows:

(1) We propose DAPlanner, a dual-agent framework for autonomous driving motion planning based on a multi-modal large language model. The framework employs two decision agents that collaborate: one generates planning trajectories through reasoning about traffic scenarios, while the other analyzes and optimizes the generated trajectories, ensuring the safety and reliability of the final path.
(2) We use Frenet Space to eliminate the interference of road shapes on model learning, and then provide the model with richer scene information through a multi-layered BEV and textual descriptions of key information, achieving a more refined representation of the driving scene.
(3) We build new datasets using the nuScenes data to train and evaluate DAPlanner, specifically for the trajectory generation and discrimination agents, and augmented these datasets through data augmentation techniques. Extensive experiments on the nuScenes validation set further validate the performance advantages of DAPlanner across multiple metrics.

2. Related work

2.1. Rule-based motion planning methods

Rule-based methods primarily rely on sampling and optimization techniques to generate the optimal driving path by analyzing factors like road conditions and traffic signals. McNaughton et al. [30] introduced an adaptive conformal spatiotemporal lattice method, using discrete sampling of spacetime for motion planning. However, this sampling approach limits the flexibility of trajectory planning, and when selecting the optimal trajectory, conflicts between manually defined cost functions can occur, impacting the rationality of the final result. Haoyang Fan et al. [7] adopted a hierarchical approach that combines multi-lane strategies, iterative optimization algorithms for path and speed, and integrates decision-making with traffic rules, striving to provide a safe, comfortable, and scalable autonomous driving experience. However, as environmental complexity increases, particularly with multi-lane strategies, the expansion of the search space brings higher computational costs, and the decomposition of path and speed may lead to local optimal solutions. Yajia Zhang et al. [31] proposed a motion planning method based on a heuristic search strategy, solving the optimal trajectory with a quadratic programming algorithm, achieving real-time performance. Nevertheless, the adaptability and robustness of the algorithm still need further improvement in highly dynamic and complex environments. Overall, traditional rule-based methods, which heavily rely on manually predefined rule logic or constraints, face challenges in handling complex and dynamic scenarios.

2.2. Learning-based motion planning methods

Learning-based methods use deep neural networks to learn human driving behavior from extensive real-world driving data. Compared to rule-based methods, they significantly reduce the workload of manually defining rules and have stronger adaptability, enabling better handling of complex driving scenarios. Bansal et al. [32] introduced a method that uses multiple grid layers to represent input information, generating human-like trajectories through imitation learning and safety detection. However, grid maps have relatively low spatial accuracy, making it challenging to effectively capture the topological relationships of

roads. To tackle this issue, some studies [33–35] employed vectorized representations and utilized graph neural networks to model the spatial relationships between different elements, achieving better results. Moreover, end-to-end frameworks have gained significant attention in recent years. For example, NEAT [36] and ST-P3 [37] enhance performance in autonomous driving tasks such as perception, prediction, and planning by learning spatiotemporal features. Methods like UniAD [38] and VAD [39] use the Transformer [40] architecture to link perception, prediction, and planning modules, further enhancing the overall performance of end-to-end autonomous driving systems. Wang et al. [41] introduced a driving world model compatible with existing end-to-end planning models, greatly improving the robustness of the planning process. Although learning-based autonomous-driving motion-planning methods, which use deep-learning techniques, improve adaptability and reduce the need for manual rule-setting, they generally perform well only in scenarios covered by the training data. However, their performance may significantly deteriorate in rare or unseen environments, limiting the model's generalization ability. Additionally, these methods do not effectively integrate information from various input scenes and modalities, which makes it challenging to ensure that the model has a broader receptive field to achieve more reliable motion planning.

2.3. Large language model-based motion planning methods

The emergence of LLMs [22,23] has brought profound changes across various industries. As LLMs have evolved, they have gradually gained the ability to process multi-modal inputs such as text, images, and video, with their generalization ability fully validated [42–45]. An increasing number of studies have applied LLMs to autonomous driving planning tasks. Xu et al. [46] introduced an end-to-end multi-modal large language model (MLLM) for autonomous driving, which interprets vehicle behavior and predicts low-level control signals by processing monocular camera video input and textual queries. Shao et al. [29] proposed a method that processes multi-camera data and LiDAR point cloud data, extracting scene information using a specially designed encoder to optimize driving decisions and control. Furthermore, some studies have used LLMs for mid-to-mid autonomous driving planning. Mao et al. [27] framed motion planning as a language modeling problem, transforming the vehicle state and obstacle information into language tokens, and leveraged the reasoning capabilities of OpenAI GPT-3.5 for planning tasks. Their proposed chain-of-thought enhances the interpretability of the planning process. Wen et al. [47] built upon this by adding reflection and memory modules, improving decision quality through experience accumulation and self-improvement. However, these methods mostly overlook map information, which is crucial for planning. Zhang et al. [48] transformed scene data into richer textual descriptions, including information on the vehicle, obstacles, lanes, traffic lights, and speed limits, and utilized a fine-tuned LLM to generate trajectories aligned with human driving behavior. Although LLM-based methods have improved motion planning performance, planning for complex inputs, especially map information, may exceed the model's maximum token limit, making it difficult to ensure that all input information can be effectively learned by the model. Additionally, the trajectory points generated directly by LLMs may have uncertainty, leading to trajectories that may not align with human driving habits or pose the risk of collisions with other road participants, thereby failing to ensure the rationality and safety of autonomous driving behavior. Therefore, this paper proposes a motion planning method based on a dual-agent framework in Frenet Space to effectively reduce the safety risks associated with LLM uncertainty.

3. Proposed method

This section provides a detailed overview of the DAPlanner framework. First, we explain the problems that this method is designed to solve (see Section 3.1). Next, we describe the main components of the method in detail, including the scene construction module, the trajectory generation agent, and the trajectory discrimination agent (see Section 3.2). Finally, we discuss the dataset generation process for the various agents (see Section 3.3).

3.1. Problem definition

The perception module provides obstacle information $\mathcal{O}$, including the type, position, speed, and size of obstacles, combined with high definition map data $\mathcal{M}$ (such as pedestrian crossings, lane dividers, road boundaries, and their topological relationships) and ego-vehicle information $\mathcal{E}$ (such as location and speed), to generate a safe and comfortable trajectory $\mathcal{T}$. The motion planning process of DAPlanner is represented by the following formula:

$$\mathcal{T} = \mathrm{DAP}(\mathcal{O}, \mathcal{M}, \mathcal{E}). \tag{1}$$

The planned trajectory $\mathcal{T}$ can be represented as a series of 2D waypoints sampled at time intervals t:

$$\mathcal{T} = [(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)], \tag{2}$$

where (x_i, y_i) denotes the waypoints coordinates the ego-vehicle will reach at the future time step i.

3.2. Dual-agent planner

The overall framework of DAPlanner is illustrated in Fig. 2, consisting of three core modules:

(1) Scene construction module: This module is responsible for converting both dynamic and static elements of the scene into Frenet Space and building a BEV map. Additionally, it generates key text prompts related to the scene to provide necessary context for subsequent decision-making and planning.
(2) Trajectory Generation Agent: This module processes the input data using an MLLM, we choose Vicuna which is a conversational large language model based on Meta's LLaMA 2. The agent can generate the driving trajectory step by step through scene analysis, behavioral decision-making, and trajectory planning.
(3) Trajectory Discrimination Agent: This module, based on another multimodal large language model, named LLaVA, evaluates the generated trajectory, employing the reflective reasoning chain to analyze potential issues within the trajectory and provide targeted optimization recommendations.

Through this series of steps, a safe and reasonable driving trajectory is ultimately generated.

3.2.1. Scenario construction

Frenet Space Mapping: The advantage of using Frenet Space for decision-making and planning tasks lies in its ability to effectively utilize the structural features of the road, thereby enabling clearer scene comprehension [31,49–51]. This method selects a guide line based on navigation routing information, decomposing the vehicle's motion in Cartesian Space into two independent one-dimensional movements: longitudinal motion along the guide line and lateral motion perpendicular to it. The coordinates of any point in the Cartesian Coordinate System are denoted as (x, y), and the guide line is represented as $\mathcal{G}[x(s), y(s), \theta(s)]$, where $x(s), y(s), \theta(s)$ denote the (x, y)-coordinates and tangent direction of the point corresponding to the arc length s along the guide line. The state of the nearest matching point $\mathcal{P}_r$ on the guide line $\mathcal{G}$ for point $\mathcal{P}$ is denoted as $(x(s_r), y(s_r), \theta(s_r))$. s_r is the arc length of $\mathcal{P}_r$ on the guide line, then it can be converted to Frenet Space by the following formulas:

$$s = s_r, \tag{3}$$

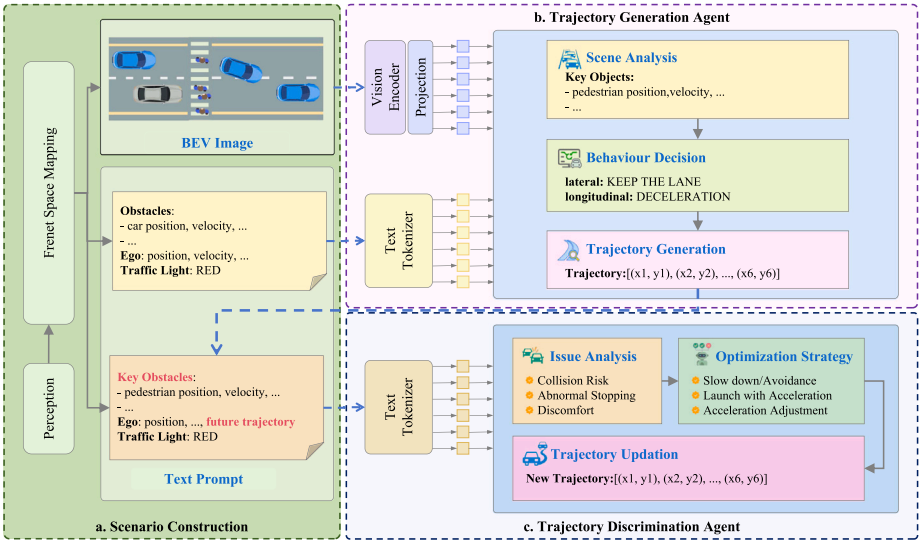

Fig. 2. The framework of Dual-Agent Planner, comprising three modules: Scenario Construction, Trajectory Generation Agent and Trajectory Discrimination Agent. The Scenario Construction can map the scenario from Cartesian Space to Frenet Space, and construct the BEV image along with their corresponding text prompts. In the Trajectory Generation Agent, multi-modal inputs are encoded and subsequently analyzed by an inference engine to identify key objects with significant impact on the ego vehicle, determine the lateral and longitudinal behavioral decisions the ego vehicle should take, and generate specific trajectory point coordinates. Ultimately, the Trajectory Discrimination Agent is responsible for identifying whether the generated trajectory contains any irrational situations such as collisions, and provides targeted optimization strategies accordingly to update the trajectory and ensure safety.

$$l = \text{sign}\left(\left(y - y\left(s_r\right)\right) \cdot \cos\left(\theta\left(s_r\right)\right) - \left(x - x\left(s_r\right)\right) \cdot \sin\left(\theta\left(s_r\right)\right)\right) \cdot \sqrt{\left(x - x\left(s_r\right)\right)^2 + \left(y - y\left(s_r\right)\right)^2}, \tag{4}$$

where s and l are the longitudinal and lateral positions in Frenet Space, respectively. This framework draws heavily on traditional driving tasks and greatly simplifies the complexity of the problem by reducing the dimensionality of the planning task. Additionally, it converts the positions of the ego vehicle and other traffic participants into road mileage and lateral offsets relative to the road, thus enabling a more effective representation of the relative positions between the ego vehicle and other participants, as well as the spatial arrangement between different traffic participants and the lanes.

BEV image: We convert various map elements in Frenet Space, including lane dividers, road boundaries, pedestrian crossings, as well as obstacles (e.g., vehicles, motor vehicles, pedestrians, traffic cones), along with the ego vehicle and its historical trajectory, onto a BEV image using distinct colors and shapes.

Text prompts: As the BEV image clearly presents the rich and detailed input information in visual form, we only need to provide brief descriptions of the objects in the image in the text prompts. Specifically, the text prompts emphasize the pixel coordinates of each object in the image for localization, while also quantifying the velocity information of each object and providing the status of relevant traffic infrastructure, such as traffic lights. This text prompting approach not only simplifies the input effectively but also ensures the precise transmission of information, further improving the accuracy and robustness of motion planning.

3.2.2. Trajectory generation agent

Feature Encoding: The visual encoder utilizes the CLIP-ViT-L-336px [52,53] to interpret and extract features from the image. By extracting intermediate-layer features, the encoder captures both local and global information more effectively. This encoding process allows the details and overall structure of the image to be fully expressed, enhancing the comprehensive understanding of the image content. The output features of the visual encoder are then processed through a projection layer to align the dimensions of the image and text features. This alignment process ensures that the image and text features can be effectively fused in the same representation space, providing unified input to the subsequent large language models. This allows the image and text information to work together, enhancing the model's overall performance.

Reasoning Engine: Based on human driving habits and the functional decomposition of autonomous driving planning methods [27,31], we have designed a thought chain for trajectory generation, which consists of the following steps:

(1) Scene Analysis: Analyze the current driving scenario, considering road conditions and interactions with surrounding traffic participants, to identify the key obstacles that significantly affect the vehicle's future driving strategy.

(2) Behavior Decision: Based on the understanding of the scene and the prediction of key obstacles, along with the vehicle's real-time information, high-level behavior decisions are made. These decisions are further broken down into lateral and longitudinal decisions. Lateral decisions include, but are not limited to, KEEP THE LANE, CHANGE TO LEFT LANE, CHANGE TO RIGHT LANE; longitudinal decisions include, but are not limited to, STOP, CONSTANT SPEED, ACCELERATION, DECELERATION, and DECELERATION TO ZERO. By combining lateral and longitudinal decisions, a wide variety of flexible behavioral strategies can be generated. It is important to note that STOP refers to the ego vehicle remaining stationary with a speed of 0 throughout the planning horizon of 3 s. DECELERATION TO ZERO refers to the scenario where the ego vehicle starts with an initial speed and gradually decelerates to 0 within the planning horizon. This is a deceleration process with the goal of "reducing speed to 0", emphasizing the full transition "from motion to rest", but it does not require the vehicle to remain stationary after coming to a stop all the time.

(3) Trajectory Generation: Based on the behavioral decisions, a trajectory $\mathcal{T} = [(s_1, l_1), (s_2, l_2), \ldots, (s_t, l_t)]$ which is represented by a series of sampled points along the timeline is generated in Frenet Space. These points are subsequently projected back into Cartesian Space to facilitate execution in the real environment, as outlined below:

$$\left(x_i, y_i\right) = \left(x\left(s_i\right) - l_i \cdot \sin\left(\theta\left(s_i\right)\right), y\left(s_i\right) + l_i \cdot \cos\left(\theta\left(s_i\right)\right)\right), \tag{5}$$

where $i = 1, 2, \ldots, t$ and the guide line is denoted as $G[x(s), y(s), \theta(s)]$. This chain of reasoning enables efficient decision-making and trajectory planning for autonomous driving systems.

3.2.3. Trajectory discrimination agent

The trajectory points generated by LLM exhibit inherent uncertainty, which prevents a full assurance of the safety and rationality of autonomous driving systems. Unlike the Reflection module in PlanAgent [54], which only provides a score for the trajectory, or the post-processing methods like collision checks and Newton iterations used in AgentDriver [55] to correct the trajectory, we have designed and trained an independent intelligent decision-making agent, which uses a reflective reasoning chain to progressively analyze the rationality of the generated trajectory points within the current driving context, identify any anomalies, and propose appropriate improvement strategies based on the analysis results, thereby ensuring the safety and rationality of the final trajectory.

Issue Analysis: The ego-vehicle trajectory generated by the Trajectory Generation Agent is denoted as $\mathcal{T} : \left[(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)\right]$, while the future trajectories of other obstacles are represented as $\mathcal{T}^j : \left[\left(x_1^j, y_1^j\right), \left(x_2^j, y_2^j\right), \ldots, \left(x_t^j, y_t^j\right)\right]$, and $\left(x_i^j, y_i^j\right)$ indicates the position of the jth obstacle at time t_i. In this module, we perform a comprehensive analysis of the generated trajectory information, key obstacle data, ego-vehicle information, and traffic light signals to identify potential anomalies and classify them. Specifically, it includes:

(1) **Collision Risk:** By calculating $\theta = \arctan(\Delta y/\Delta x)$, the directions of the ego-vehicle and obstacle trajectory points can be calculated. Using their respective lengths and widths, the ego-vehicle's polygon $Polygon_i$ and the obstacle's polygon $Polygon_i^j$ at time i can be constructed, and it is then determined whether polygon $Polygon_i$ and $Polygon_i^j$ overlap. If the ego-vehicle's polygon overlaps with any obstacle's polygon at any time, it indicates a collision risk in the generated trajectory.

(2) **Abnormal Stopping:** The average speed of the ego-vehicle's planned trajectory is calculated from the ego-vehicle's trajectory point coordinates as follows:

$$v_{\text{avg}} = \frac{\sum_{i=1}^{t-1} \sqrt{\left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2}}{\sum_{i=1}^{t-1} \left(t_{i+1} - t_i\right)}. \tag{6}$$

If $v_{avg} < v_{\min}$, where $v_{\min}$ represents a low-speed threshold, it indicates that the planned trajectory expects the ego-vehicle to remain stationary or crawl at an extremely low speed. This occurs when there is no traffic light, or the traffic light is green, and no other obstacles are influencing the vehicle, which is considered an abnormal stopping state.

(3) **Discomfort:** A cubic polynomial is used to fit the generated trajectory, as follows:

$$x = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \tag{7}$$

$$y = b_0 + b_1 t + b_2 t^2 + b_3 t^3. \tag{8}$$

This allows the calculation of the acceleration $a = \sqrt{(x'')^2 + (y'')^2}$ and curvature $k = \left|x'y'' - y'x''\right| / \left(\left(x'\right)^2 + \left(y'\right)^2\right)^{1.5}$ at any point on the trajectory. If $a > acc_{\max}$ or $a < -dec_{\max}$, where $acc_{\max}$ and $dec_{\max}$ represent the maximum acceleration and maximum deceleration, respectively, this indicates an abnormal situation of rapid acceleration or deceleration in the planned trajectory. If $k > k_{\max}$, with $k_{\max}$ representing the threshold for maximum curvature, this indicates an abnormal situation of sharp turns in the planned trajectory.

Optimization Strategy: We propose specific optimization strategies for different types of abnormal situations:

(1) **Collision Risk:** When potential collision risks are detected, the system automatically evaluates the current environment and adjusts the ego-vehicle's longitudinal speed or performs a lateral shift to avoid collisions with obstacles.

(2) **Abnormal Stopping:** If the system detects an inappropriate stopping situation, it automatically triggers an acceleration strategy to return the vehicle to a normal driving speed. At the same time, the acceleration will not create new collision risks.

(3) **Discomfort:** When the trajectory exhibits sharp turns, abrupt acceleration or braking that fail to meet comfort requirements, the longitudinal or lateral speed is adjusted to ensure both comfort and safety during driving. At the same time, the acceleration will not create new collision risks.

Trajectory Update: Based on the optimization strategies provided by issue analysis, the original trajectory is adjusted to generate a new, safer, and more rational trajectory, effectively avoiding the aforementioned anomalies. Through this feedback mechanism, the stability and driving experience of the autonomous driving system in complex scenarios are effectively enhanced.

3.3. Dataset generation

To create a vision-language-based foundational driving dataset, we generated BEV data and corresponding textual descriptions based on the nuScenes dataset. Additionally, we produced driving data containing various types of anomalies and further expanded the dataset's scale using data augmentation techniques. We separately generated data for the Trajectory Generation Agent and the Trajectory Discrimination Agent. An example is shown in Fig. 3.

Trajectory Generation Agent Data: We extracted map information from the nuScenes dataset within a specified range around the ego-vehicle and obtained continuous frame data for the ego-vehicle and obstacles within the past 2 s and the next 3 s. Using this information, we generated a reference guide line by combining it with navigation data, performed spline curve fitting, and transformed all information into Frenet Space with respect to the guide line. The final result was visualized as a BEV map. As shown in the top-right corner of Fig. 3 left part, the white box represents the ego-vehicle's position, orientation, and shape, while the blue, magenta, red, and yellow boxes correspond to the information of vehicles, motor vehicles, pedestrians, and other obstacles, respectively. The dashed lines in matching colors represent the historical trajectory points of these objects. The orange lines represent lane dividers, the green lines indicate road boundaries, and the cyan lines represent pedestrian crossings.

For each BEV image, corresponding textual data is generated, which includes:

(1) **Captions:** Descriptions of the pixel coordinates, motion states, and speed information of dynamic obstacles (such as vehicles, pedestrians, etc.) in the image, as well as the ego-vehicle's position, speed, historical trajectory points, and traffic light information.

(2) **Reasoning:** Identifying targets from the perception results that significantly impact the ego-vehicle's movement, inferring the necessary behavioral strategy for the ego-vehicle, and generating the corresponding trajectory point data.

Trajectory Discrimination Agent Data: Most existing datasets are primarily based on normal driving data, with relatively few abnormal situations, such as collisions. To address this, we developed a method for generating abnormal datasets, simulating various types of anomalies, including collisions and abnormal stopping, and performing cause analysis. These abnormal data will be used for training and validating the trajectory discrimination agent. Specifically, we selected various abnormal situations mentioned in Section 3.3 from the reasoning results in the training set and generated corresponding textual data:

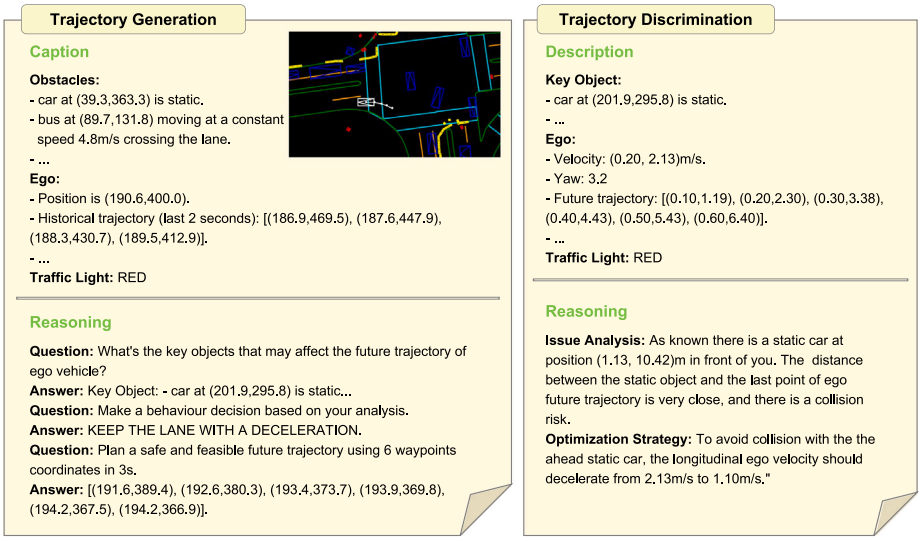

Fig. 3. Example of generated data. The left part of this figure demonstrates the data for Trajectory Generation, which includes a BEV image depicting the ego vehicle, other traffic participants and various map elements, as well as captions and detailed reasoning regarding the image. The right part shows the data for Trajectory Discrimination, primarily consisting of a description of the scene and an anomaly analysis of the ego vehicle's future trajectory.

(1) **Description:** Descriptions of the pixel coordinates, motion states, and speed information of key dynamic obstacles (such as vehicles, pedestrians, etc.) in the image, as well as the ego-vehicle's position, speed, planned future trajectory points, and traffic light information.
(2) **Reasoning:** A detailed characterization of various anomaly types and their corresponding optimization strategies.

Data Augmentation: Due to the scarcity of abnormal data, we applied data augmentation to improve the performance of the trajectory discrimination agent. The specific augmentation methods include mirror clipping of the scene and adding lateral or longitudinal noise to the future trajectory. Experimental results indicate that these data augmentation methods significantly improved the accuracy and robustness of the trajectory discrimination agent.

4. Experiment

4.1. Experimental setup

4.1.1. Implementation details

We applied the data generation method described in Section 3.3 to process the nuScenes dataset and constructed training data for the Trajectory Generation Agent and Trajectory Discrimination Agent. The training was conducted on 4 V100 GPUs, with fine-tuning applied to the pre-trained LLM. During optimization, the AdamW optimizer was employed, without weight decay. For the Trajectory Generation Agent, the input BEV resolution was set to 800 × 400, with a perception range extending 40 meters in front and behind the ego-vehicle, and 20 meters to each side. The input data also included the past 2 s of the ego-vehicle's trajectory. Based on the initialization parameters of the vicuna-v1.5-7b model, the global batch size was set to 8, with training conducted for 1 epoch, at a learning rate of 2e-4, and a cosine annealing schedule. Vicuna-v1.5 is an open-source conversational large language model based on Meta's LLaMA 2, developed by the LMSYS team. The final model output consists of the coordinates of 6 trajectory points for the next 3 s, with a time interval of 0.5 s between each point. For the Trajectory Discrimination Agent, we used llava-v1.5-7b as the model initialization. During training, the global batch size was set to 16, with training conducted over 4 epochs, at a learning rate of 2e-5, and a cosine annealing schedule. LLaVA-v1.5 is also an open-source multimodal large language model focused on integrating visual and language tasks. The model is developed by the LLaVA team and aims to achieve advanced multimodal learning capabilities through efficient training methods and the combination of multimodal data.

4.1.2. Evaluation metrics

Referring to previous studies [37–39], the L2 error and collision rate were used as the primary metrics for evaluating model performance in the experiments based on the nuScenes validation set. However, we noticed that the evaluation protocols used in VAD and ST-P3 differ from the evaluation method in UniAD. To ensure consistency and reliability in the evaluation, this paper adopts the evaluation protocols used in VAD and ST-P3. Additionally, it is also important to note that L2 error and collision rate do not account for collisions between the vehicles and the road boundaries, which is a critical factor for driving safety. Through testing the planned trajectories generated by various methods, we observed that some high-risk trajectories, such as veering off the road, may not be heavily penalized by the existing metrics. To more comprehensively and accurately assess the quality of planned trajectories, we introduced a new evaluation metric — the intersection rate with road boundaries [56], measured as a percentage. The existing L2 Error and Collision Rate metrics reflect the distance deviation between the predicted trajectory and the actual human driving trajectory, as well as the risk of collision with other traffic participants. However, these two metrics fail to account for the risk of collision between the driving trajectory and the lane boundary. In our replication of other baseline experiments, we often found that the predicted trajectory intersects with the road boundary. This suggests that the vehicle will eventually exit the road boundary. This violates lane-keeping rules and may result in a collision with the curb, even though such trajectories may not necessarily increase the Collision Rate metric. Therefore, we introduce the new metric, intersection rate with road boundary. This metric determines whether the ego vehicle's bounding box will overlap with the road boundary based on the predicted trajectory and calculates the proportion of overlapping scenarios among all scenarios. It better reflects the relative positional relationship between the vehicle and the road boundaries during driving, providing a more detailed assessment of driving safety.

4.2. Main results

To more comprehensively evaluate the safety of the planning method, we added a new evaluation metric — the intersection rate with the road

Table 1
Motion planning performance. The top and runner-up performances are respectively indicated by bold and underline formatting.

Method	L2 (m)				Collision (%)				Intersection (%)			
	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.
NMP [57]	–	–	2.31	–	–	–	1.92	–	–	–	–	
SA-NMP [57]	–	–	2.05	–	–	–	1.59	–	–	–	–	
FF [58]	0.55	1.20	2.54	1.43	0.06	0.17	0.07	0.43	–	–	–	–
EO [59]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33	–	–	–	–
ST-P3 [37]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	2.53	8.17	14.4	8.37
UniAD [38]	0.44	0.67	**0.96**	0.69	0.04	0.08	0.23	0.12	**0.21**	1.32	3.63	1.72
VAD-Tiny [39]	0.46	0.76	1.12	0.78	0.21	0.35	0.58	0.38	0.94	3.22	7.65	3.94
VAD-Base [39]	0.41	0.70	1.05	0.72	0.07	0.17	0.41	0.22	0.60	2.38	5.18	2.72
Drive-WM [41]	0.43	0.77	1.20	0.80	0.10	0.21	0.48	0.26	–	–	–	–
Ours	**0.33**	**0.63**	1.03	**0.66**	**0.01**	**0.05**	**0.19**	**0.08**	0.57	**0.88**	**1.07**	**0.84**

boundaries — in addition to the traditional metrics of L2 error and collision rate. As shown in Table 1, the experimental results demonstrate that, compared to previous methods, our approach achieves significant improvements across all metrics. It is worth noting that recent studies have emphasized the impact of ego status on planning results [39,56]. For fairness, none of the methods in this study used ego status for comparison. Furthermore, the collision rate and intersection rate with road boundaries are more critical compared to the L2 error, because there may be multiple choices of driving behaviors in the same scenario, but the collision rate and road boundaries crossing rate directly affect the driving safety of the vehicle. Therefore, optimizing these two metrics is especially important. From the results in Table 1, it can be seen that, compared to existing methods, our proposed model shows further improvement in the L2 error metric, indicating that this method can better learn and simulate human driving behavior. In terms of collision rate, our method shows significant improvement in all time intervals compared to the excellent-performing UniAD and VAD-Base methods. Specifically, the average collision rate decreased by 33.33% and 63.64%, respectively. This improvement is largely due to our Trajectory Discrimination Agent's ability to effectively identify potential collision risks and adjust accordingly through the reflective reasoning mechanism. Additionally, by mapping all relevant information to Frenet Space, we eliminated interference from road structures during the model learning process, which significantly reduced the intersection rate with the road boundaries. Compared to UniAD and VAD-Base, the average intersection rate with the road boundaries decreased by 50.00% and 68.38%, respectively, greatly reducing the risk of the vehicle crossing the road boundaries or colliding with the curb. These results further validate the significant advantages of our method in terms of safety and robustness.

4.3. Ablation study

4.3.1. Ablation of components

To validate the effectiveness of the individual modules and strategies we proposed, we performed ablation experiments to analyze the contribution of each component to the planning performance. As shown in Table 2, the experiments demonstrate the impact of the Trajectory Generation Agent, Frenet Space Mapping, Trajectory Discrimination Agent, and Data Augmentation on the final outcomes. In the ablation study, the Trajectory Generation Agent represents trajectory generation performed only in Cartesian Space. Frenet Space Mapping is responsible for converting Cartesian Space to Frenet Space. Subsequently, the Trajectory Discrimination Agent is added, and the Data Augmentation strategy is applied during data generation. The experimental results indicate that the addition of each component played an important role in improving planning performance, especially in the two metrics closely related to vehicle safety: average collision rate and average intersection rate with road boundaries.

The introduction of Frenet Space Mapping significantly improved the planning results by eliminating the interference of road structure on the model's learning, making the vehicle trajectory better align with the actual road constraints. The application of this module effectively reduced the average collision rate and average intersection rate with the road boundaries by 55.56% and 73.50%, respectively, greatly enhancing vehicle safety and ensuring that the vehicle consistently stayed within the road boundaries. The inclusion of the Trajectory Discrimination Agent, although increasing the average L2 error by 5 cm, significantly reduced the average collision rate, further decreasing it by 37.50%. This result shows that the Trajectory Discrimination Agent performs excellently in detecting and analyzing potential collision risks, as it can identify trajectories that may collide with other traffic participants and adjust them using appropriate optimization strategies to effectively avoid creating collision risks.

Due to the relative scarcity and difficulty in obtaining abnormal driving data, such as traffic accidents, we effectively addressed the issues of data scarcity and imbalance through the Data Augmentation strategy. Specifically, Data Augmentation added more variations, including lateral and longitudinal noise, when generating abnormal data. This not only reduced the risk of overfitting during training but also significantly enhanced the model's generalization ability. Through this data augmentation method, the average collision rate was further reduced by 20%, further enhancing the model's adaptability and safety in real-world driving scenarios.

4.3.2. Ablation of LLM

We fine-tuned different large pre-trained models for the Trajectory Generation Agent and Trajectory Discrimination Agent, and tested them on the validation set to evaluate the performance metrics of different model combinations. The results are presented in Table 3. The experimental results clearly show that agents using different LLM combinations for both the Trajectory Generation Agent and Trajectory Discrimination Agent demonstrated satisfactory performance, validating the high compatibility of our system with various LLMs. Specifically, our proposed motion planning system based on the dual-agent framework is capable of achieving safe and high-quality planning performance without relying on a specific large model, by efficiently expressing and processing environmental data and ensuring effective collaboration between the two agents.

The Trajectory Generation Agent generates an initial driving trajectory based on environmental data, while the Trajectory Discrimination Agent analyzes the generated trajectory, identifies potential safety risks, and optimizes it. The collaboration between the two ensures that the motion planning process can not only handle complex traffic scenarios but also ensure the safety and rationality of the autonomous vehicle's driving in dynamic environments. Furthermore, our framework fully exploits the powerful data comprehension and logical reasoning abilities of large models to perform the motion planning task, ensuring the stability and safety of the autonomous driving system in practical applications.

Among all the test combinations, the system delivered the best overall performance when the Trajectory Generation Agent used vicuna-v1.5-7b and the Trajectory Discrimination Agent used llava-v1.5-7b. This result indicates that, for the current task, combining these two

Table 2
The influence of using different module combination on the metrics of average L2, Collision and Intersection in the 1–3 s period.

Trajectory Generation Agent	Frenet Space Mapping	Trajectory Discrimination Agent	Data Augmentation	L2 (m)	Collision (%)	Intersection (%)
✓				0.88	0.36	3.17
✓	✓			0.62	0.16	0.84
✓	✓	✓		0.67	0.10	0.84
✓	✓	✓	✓	0.66	0.08	0.84

Table 3
The influence of agents using different language models on the metrics of average L2, Collision and Intersection in the 1–3 s period.

Trajectory Generation Agent	Trajectory Discrimination Agent	L2 (m)	Collision (%)	Intersection (%)
vicuna-v1.5-7b	llava-v1.5-7b	0.66	0.08	0.84
vicuna-7b-v1.5	vicuna-v1.5-7b	0.69	0.08	0.86
llava-v1.5-7b	vicuna-7b-v1.5	0.69	0.10	0.89
llava-v1.5-7b	llava-v1.5-7b	0.67	0.10	0.87

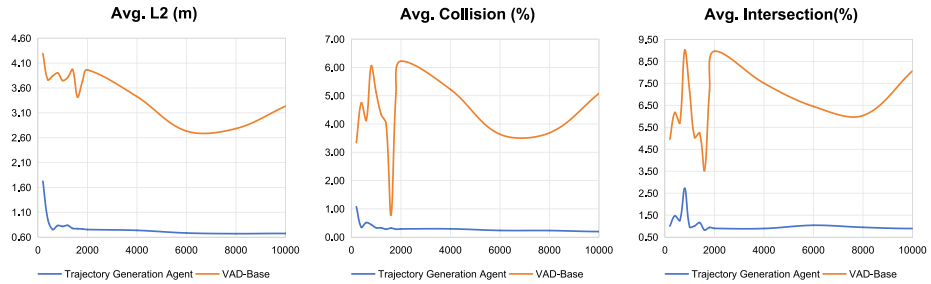

Fig. 4. Ablation of learning rate for Trajectory Discrimination Agent.

large models provides the best performance for the system, further confirming the crucial role of model selection in the effectiveness of planning.

4.3.3. Ablation of step for trajectory generation agent

To further validate the efficiency and outstanding generalization ability of our method, we conducted only a few iterations on the dataset, trained the Trajectory Generation Agent, and compared it with the state-of-the-art motion planning methods in VAD. The experimental results are shown in Fig. 4.

It is worth noting that our method exhibits faster convergence across all metrics. In fewer than 2000 iterations, all metrics have already converged to a satisfactory level. This contrasts sharply with other learning-based motion planning methods that rely on large datasets for training. Our method, after only minimal data fine-tuning, is able to demonstrate excellent performance on the complete validation set. This outstanding performance not only strongly proves the robustness and exceptional generalization ability of our method, but also clearly highlights its significant advantage in few-shot learning. This advantage is of great significance in practical application scenarios, especially when resources are limited. Our efficient learning capability significantly reduces the reliance on large amounts of labeled data, thereby effectively lowering the costs of data collection and labeling in many application scenarios. Traditional learning-based motion planning methods often require large amounts of labeled data, which typically involve substantial time, human, and material resources to obtain. In contrast, our method reduces the need for these expensive resources, thereby effectively lowering costs.

4.3.4. Ablation of learning rate for trajectory discrimination agent

As shown in Fig. 5, for the Trajectory Discrimination Agent, we trained the model with different learning rates and thoroughly evaluated key metrics, including L2 error, collision rate, and intersection rate with road boundaries. The results reveal that the choice of learning rate has a significant impact on model performance. When the learning rate is set to 2e-5, all metrics achieve their optimal performance. Specifically, in terms of L2 error — whether over 1-second, 2-second, or 3-second windows, or the average error — the learning rate of 2e-5 outperforms other settings, indicating the model's superior accuracy. Regarding collision rate, the 2e-5 learning rate results in significantly lower short-term and average collision rates compared to other settings, demonstrating its effectiveness in reducing potential collision risks. In terms of the intersection rate with road boundaries, a key indicator of trajectory validity and vehicle safety, the model performs best with a 2e-5 learning rate, demonstrating the model's ability to constrain the trajectory within a safe and reasonable range. Therefore, a learning rate of 2e-5 achieves the best balance between model accuracy and safety, optimizing performance across all key metrics.

4.4. Qualitative results

Fig. 6 shows the qualitative experimental results of the DAPlanner framework, demonstrating the system's behavior and optimization strategies in different scenarios. In the BEV image, the white solid line represents the ground truth trajectory of the ego vehicle, the magenta solid line represents the trajectory output by the Trajectory Generation Agent, and the green solid line represents the trajectory adjusted by the Trajectory Discrimination Agent. In Fig. 6 (a), the scenario involves a vehicle passing through an intersection where other vehicles have already started moving, but the Trajectory Generation Agent decides to remain stationary, creating a potential collision risk with the vehicle behind. In response, the Trajectory Discrimination Agent successfully identified this anomaly and proposed an optimization strategy: adjusting the ego vehicle's speed to follow the lead vehicle's departure, effectively avoiding the potential collision risk.

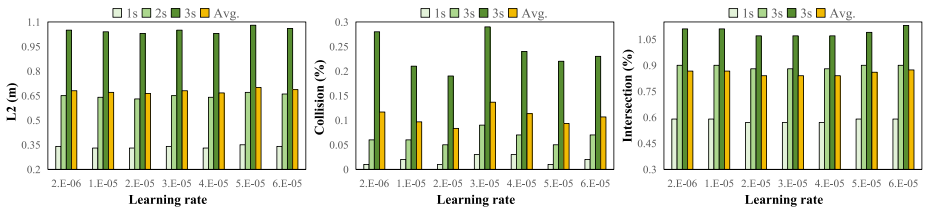

Fig. 5. Ablation of learning rate for Trajectory Discrimination Agent.

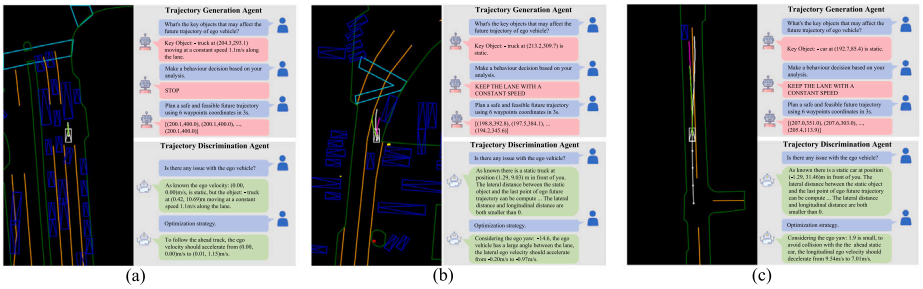

Fig. 6. Qualitative results of dual-agent planner.

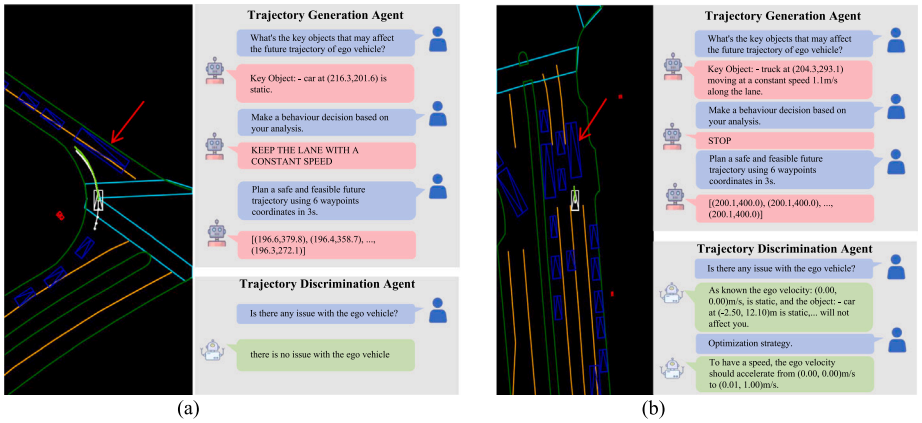

Fig. 7. Failure cases of dual-agent planner.

In Fig. 6(b), a stationary obstacle is present ahead in the scenario. The initial trajectory generated by the Trajectory Generation Agent presents a potential collision risk with the obstacle. The Trajectory Discrimination Agent promptly identified this issue and proposed an optimization strategy: increasing the leftward lateral acceleration to steer the vehicle onto a larger left turn trajectory, bypassing the obstacle and ensuring safe driving.

Fig. 6(c) shows another typical scenario, where the initial strategy generated by the Trajectory Generation Agent is to maintain a constant speed, but this trajectory poses a collision risk with the obstacle ahead. The Trajectory Discrimination Agent identified the issue and proposed an optimization strategy to decelerate, allowing the vehicle to stop before approaching the obstacle. Although different from the ground truth strategy (which involves circumventing the obstacle to the right), our method still effectively avoids collisions and ensures driving safety. This result demonstrates that DAPlanner is not only capable of flexibly adapting to diverse driving scenarios, but also possesses strong adaptability. Its decisions are not rigid imitations of human behavior; rather, they are based on learning multiple strategies and selecting the optimal solution for each real-world scenario.

Fig. 7 shows some failure cases of our method. In Fig. 7(a), in a sharp turn scenario, there is a large vehicle parked on the side of the road. There is a large vehicle parked on the side of the road, indicated by the red arrow in the figure. The vehicle partially crosses the lane boundary, occupying the ego vehicle's current lane, resulting in a very narrow drivable area. Although the trajectory generated by the Trajectory Generation Agent has a small deviation from the ground truth, it still collides with the obstacle on the right side. The Trajectory Discrimination Agent also fails to recognize this risk, assuming that there is no issue with the ego vehicle, and thus does not alter the Trajectory Discrimination Agent's output, leading to a collision. This may be due to the Agent's insensitivity to small numerical deviations, which could cause issues in scenarios requiring fine-grained accuracy. Fig. 7(b) is an intersection scenario where the traffic light turns from red to green. The Trajectory Discrimination Agent assumes that the vehicle should start slowly, but the vehicle ahead remains stationary for the entire planning horizon of 3 s. The vehicle does not follow the expected behavior of starting to cross the intersection, leading to a collision with the vehicle ahead after 2 s. As a result, in some unconventional situations, the Agent may make overly simplistic assumptions, leading to failure. To alleviate the issues of the large model's limited sensitivity to small numerical deviations and the narrow scope of abnormal scenarios covered by the current Trajectory Discrimination Agent, future work will begin by developing a closed-loop simulation system. This will include introducing rigid feedback mechanisms, such as collision penalties, and improving robustness in complex scenarios through iterative optimization. Next, we will refine trajectory selection through a multi-dimensional evaluation system. Finally, we will expand

the abnormal situations dataset and refine the Trajectory Discrimination Agent's strategy, particularly for corner cases such as sharp turns, narrow passages, and starting at congested intersections.

5. Conclusion

In this article, we propose the DAPlanner, a planning system based on multiple large language model agents for mid-to-mid autonomous driving decision planning. The DAPlanner builds static map data and dynamic traffic participant data in Frenet Space, effectively utilizing prior knowledge of road structures to express the driving scenario in a more concise and efficient manner, which aids the backend model in understanding the environment. By using the transformed data, BEV images and corresponding textual cues are generated, preserving a rich and diverse array of input information. Through the collaborative operation of the trajectory generation agent and the trajectory discrimination agent, and leveraging a chain-of-thought specifically designed for autonomous driving planning tasks, the planning trajectory is gradually generated. The trajectory is then analyzed and evaluated based on the current scene. If issues are detected, targeted optimization strategies are proposed, enhancing the stability and rationality of the planned trajectory and ensuring the safety of vehicle operation. Simultaneously, we constructed new autonomous driving planning datasets for the training and evaluation of the Trajectory Generation Agent and the Trajectory Discrimination Agent. These datasets encompass BEV image and text pairs, as well as issue scenarios for motion planning and data related to the analysis of these issues, and corresponding improvement strategies. We conducted a series of experiments that verify our proposed method outperforms previous approaches across various metrics. Ablation experiments also confirm the effectiveness of each of the proposed modules.

Our method is highly dependent on high definition map, and it may encounter issues in scenarios where high definition map is unavailable or when there are changes to the map. Therefore, relying solely on high-precision maps in autonomous driving systems has significant limitations, particularly regarding incomplete map coverage and outdated updates. Additionally, the process of collecting and creating high-precision maps is complex and costly, making it a challenging problem to solve. Future solutions must prioritize the synergy between perception technology and map data, and how to ensure the safety and efficiency of autonomous driving systems in low-precision map environments through real-time perception and intelligent decision-making. As a result, the effective extraction of perception layer information, integration with low-precision maps, and the enhancement of autonomous driving robustness through deep learning and multi-sensor fusion will be the primary focus of our future research. Additionally, the datasets and metrics we use are based on open-loop systems, which do not account for the feedback of the model's output on the environment. This limits the ability to simulate dynamic feedback and continuous decision-making processes in actual driving, restricting a comprehensive evaluation of the model's capabilities in real-world applications. Furthermore, since our method relies on two large language models, real-time performance remains a critical issue that needs to be resolved. To address these limitations, we plan to incorporate real-time perception to construct local maps and extend the method to closed-loop tasks in future work. Additionally, the datasets and metrics we use are based on open-loop systems, which do not account for the feedback of the model's output on the environment. This limits the ability to simulate dynamic feedback and continuous decision-making processes in actual driving, restricting a comprehensive evaluation of the model's capabilities in real-world applications. Furthermore, since our method relies on two large language models, real-time performance remains a critical issue that needs to be resolved. To address these limitations, we plan to incorporate real-time perception to construct local maps to replace high definition map and extend the method to closed-loop tasks in future work. Additionally, we will utilize reinforcement learning from human feedback and knowledge distillation techniques to minimize model latency.

CRediT authorship contribution statement

Pin Zhang: Writing – review & editing, Writing – original draft, Software, Methodology, Conceptualization. **Ke Lin:** Writing – review & editing, Visualization, Methodology, Conceptualization. **Duantengchuan Li:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Zixun Fu:** Writing – review & editing, Validation, Resources, Data curation. **Yuefeng Cai:** Writing – review & editing, Visualization, Validation, Data curation. **Bing Li:** Validation, Supervision, Project administration, Funding acquisition. **Huan Yu:** Software, Formal analysis, Data curation. **Ming Li:** Visualization, Supervision, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under (No. 62032016).

Data availability

Data will be made available on request.

References

[1] K. Lin, Y. Li, S. Chen, D. Li, X. Wu, Motion planner with fixed-horizon constrained reinforcement learning for complex autonomous driving scenarios, IEEE Trans. Intell. Veh. 9 (1) (2024) 1577–1588, http://dx.doi.org/10.1109/TIV.2023.3273857.

[2] D. Parekh, N. Poddar, A. Rajpurkar, M. Chahal, N. Kumar, G.P. Joshi, W. Cho, A review on autonomous vehicles: Progress, methods and challenges, Electronics 11 (14) (2022).

[3] M. Reda, A. Onsy, A.Y. Haikal, A. Ghanbari, Path planning algorithms in the autonomous driving system: A comprehensive review, Robot. Auton. Syst. 174 (2024) 104630.

[4] G. Lan, Q. Hao, End-to-end planning of autonomous driving in industry and academia: 2022–2023, 2023, arXiv e-prints arXiv:2401.08658.

[5] M. Treiber, A. Hennecke, D. Helbing, Congested traffic states in empirical observations and microscopic simulations, Phys. Rev. E 62 (2000) 1805–1824.

[6] A. Sauer, N. Savinov, A. Geiger, Conditional affordance learning for driving in urban environments, in: A. Billard, A. Dragan, J. Peters, J. Morimoto (Eds.), Proceedings of the 2nd Conference on Robot Learning, in: Proceedings of Machine Learning Research, vol. 87, 2018, pp. 237–252.

[7] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, Q. Kong, Baidu apollo em motion planner, 2018, arXiv:1807.08048.

[8] D. Li, S. Wang, W. Zhao, L. Kang, L. Dong, J. Wang, X. Wang, Adgaze: Anisotropic gaussian label distribution learning for fine-grained gaze estimation, Pattern Recognit. 164 (2025) 111536, http://dx.doi.org/10.1016/j.patcog.2025.111536.

[9] J. Zhang, Y. Tan, D. Li, G. Xu, F. Zhou, Eanet: Integrate edge features and attention mechanisms multi-scale networks for vessel segmentation in retinal images, IET Image Process. 19 (1) (2025) e70056, http://dx.doi.org/10.1049/ipr2.70056, https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ipr2.70056.

[10] D. Li, C. Deng, X. Wang, Z. Li, C. Zheng, J. Wang, B. Li, Joint inter-word and inter-sentence multi-relation modeling for summary-based recommender system, Inf. Process. Manage. 61 (3) (2024) 103631.

[11] H. Liu, C. Zheng, D. Li, X. Shen, K. Lin, J. Wang, Z. Zhang, Z. Zhang, N.N. Xiong, Edmf: Efficient deep matrix factorization with review feature learning for industrial recommender system, IEEE Trans. Ind. Inform. 18 (7) (2022) 4361–4371.

[12] F. Shi, D. Li, X. Wang, B. Li, X. Wu, Tgformer: A graph transformer framework for knowledge graph embedding, IEEE Trans. Knowl. Data Eng. 37 (1) (2025) 526–541, http://dx.doi.org/10.1109/TKDE.2024.3486747.

[13] D. Li, T. Xia, J. Wang, F. Shi, Q. Zhang, B. Li, Y. Xiong, Sdformer: A shallow-to-deep feature interaction for knowledge graph embedding, Knowl.-Based Syst. 284 (2024) 111253.

[14] K. Lin, D. Li, Y. Li, S. Chen, X. Wu, Fhcpl: An intelligent fixed-horizon constrained policy learning system for risk-sensitive industrial scenario, IEEE Trans. Ind. Inform. 20 (4) (2024) 5794–5804.

[15] K. Lin, D. Li, Y. Li, S. Chen, Q. Liu, J. Gao, Y. Jin, L. Gong, Tag: Teacher-advice mechanism with gaussian process for reinforcement learning, IEEE Trans. Neural Netw. Learn. Syst. 35 (9) (2024) 12419–12433, http://dx.doi.org/10.1109/TNNLS.2023.3262956.

[16] K. Lin, Y. Li, Q. Liu, D. Li, X. Shi, S. Chen, Almost surely safe exploration and exploitation for deep reinforcement learning with state safety estimation, Inform. Sci. 662 (2024) 120261.

[17] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, A. Alahi, Social gan: Socially acceptable trajectories with generative adversarial networks, 2018, arXiv:1803.10892.

[18] L. Chen, Y. Li, C. Huang, B. Li, Y. Xing, D. Tian, L. Li, Z. Hu, X. Na, Z. Li, S. Teng, C. Lv, J. Wang, D. Cao, N. Zheng, F.-Y. Wang, Milestones in autonomous driving and intelligent vehicles: Survey of surveys, IEEE Trans. Intell. Veh. 8 (2) (2023) 1046–1056.

[19] K. Muhammad, A. Ullah, J. Lloret, J.D. Ser, V.H.C. de Albuquerque, Deep learning for safe autonomous driving: Current challenges and future directions, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 4316–4336.

[20] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, M. Liu, Rethinking imitation-based planners for autonomous driving, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 14123–14130.

[21] S. Hagedorn, M. Hallgarten, M. Stoll, A. Condurache, The integration of prediction and planning in deep learning automated driving systems: A review, 2024, arXiv:2308.05731.

[22] J. Achiam OpenAI, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H.W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S.P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S.S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Ł.ukasz. Kaiser, A. Kamali, I. Kanitscheider, N.S. Keskar, T. Khan, L. Kilpatrick, J.W. Kim, C. Kim, Y. Kim, J.H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, Ł.ukasz. Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C.M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S.M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F. de Avila Belbute Peres, M. Petrov, H.P. de Oliveira Pinto, Michael, Pokorny, M. Pokrass, V.H. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F.P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M.B. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J.F.C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J.J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, Gpt-4 technical report, 2024, arXiv:2303.08774.

[23] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C.C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P.S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E.M. Smith, R. Subramanian, X.E. Tan, B. Tang, R. Taylor, A. Williams, J.X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, 2023, arXiv:2307.09288.

[24] Y. Liu, D. Li, K. Wang, Z. Xiong, F. Shi, J. Wang, B. Li, B. Hang, Are llms good at structured outputs? a benchmark for evaluating structured output capabilities in llms, Inf. Process. Manage. 61 (5) (2024) 103809, http://dx.doi.org/10.1016/j.ipm.2024.103809.

[25] F. Shi, P. Qing, D. Yang, N. Wang, Y. Lei, H. Lu, X. Lin, D. Li, Prompt space optimizing few-shot reasoning success with large language models, in: Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 1836–1862, http://dx.doi.org/10.18653/v1/2024.findings-naacl.119, https://aclanthology.org/2024.findings-naacl.119.

[26] L. Chen, O. Sinavski, J. Hünermann, A. Karnsund, A.J. Willmott, D. Birch, D. Maund, J. Shotton, Driving with llms: Fusing object-level vector modality for explainable autonomous driving, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 14093–14100.

[27] J. Mao, Y. Qian, J. Ye, H. Zhao, Y. Wang, Gpt-driver: Learning to drive with gpt, 2023, arXiv:2310.01415.

[28] H. Sha, Y. Mu, Y. Jiang, L. Chen, C. Xu, P. Luo, S.E. Li, M. Tomizuka, W. Zhan, M. Ding, Languagempc: Large language models as decision makers for autonomous driving, 2023, arXiv:2310.03026.

[29] H. Shao, Y. Hu, L. Wang, G. Song, S.L. Waslander, Y. Liu, H. Li, Lmdrive: closed-loop end-to-end driving with large language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 15120–15130.

[30] M. McNaughton, C. Urmson, J.M. Dolan, J.-W. Lee, Motion planning for autonomous driving with a conformal spatiotemporal lattice, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 4889–4895.

[31] Y. Zhang, H. Sun, J. Zhou, J. Pan, J. Hu, J. Miao, Optimal vehicle path planning using quadratic optimization for baidu apollo open platform, in: 2020 IEEE Intelligent Vehicles Symposium, IV, 2020, pp. 978–984.

[32] M. Bansal, A. Krizhevsky, A. Ogale, Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst, 2018, arXiv:1812.03079.

[33] O. Scheel, L. Bergamini, M. Wolczyk, B. Osiński, P. Ondruska, Urban driver: Learning to drive from real-world demonstrations using policy gradients, in: Proceedings of the 5th Conference on Robot Learning, in: Proceedings of Machine Learning Research, vol. 164, 2022, pp. 718–728.

[34] M. Vitelli, Y. Chang, Y. Ye, A. Ferreira, M. Wołczyk, B. Osiński, M. Niendorf, H. Grimmett, Q. Huang, A. Jain, P. Ondruska, Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies, in: 2022 International Conference on Robotics and Automation, ICRA, 2022, pp. 897–904.

[35] K. Renz, K. Chitta, O.-B. Mercea, A.S. Koepke, Z. Akata, A. Geiger, Plant: Explainable planning transformers via object-level representations, 2022, arXiv:2210.14222.

[36] K. Chitta, A. Prakash, A. Geiger, Neat: Neural attention fields for end-to-end autonomous driving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 15793–15803.

[37] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, D. Tao, St-p3: End-to-end vision-based autonomous driving via spatial–temporal feature learning, in: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, 2022, pp. 533–549.

[38] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, H. Li, Planning-oriented autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2023, pp. 17853–17862.

[39] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, X. Wang, Vad: Vectorized scene representation for efficient autonomous driving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV, 2023, pp. 8340–8350.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023, arXiv:1706.03762.

[41] Y. Wang, J. He, L. Fan, H. Li, Y. Chen, Z. Zhang, Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 14749–14759.

[42] H. Liu, C. Li, Q. Wu, Y.J. Lee, Visual instruction tuning, 2023, arXiv:2304.08485.

[43] R. Luo, Z. Zhao, M. Yang, J. Dong, D. Li, P. Lu, T. Wang, L. Hu, M. Qiu, Z. Wei, Valley: Video assistant with large language model enhanced ability, 2023, arXiv:2306.07207.

[44] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, Y. Qiao, Videochat: Chat-centric video understanding, 2024, arXiv:2305.06355.

[45] C. Singh, A. Askari, R. Caruana, J. Gao, Augmenting interpretable models with large language models during training, Nat. Commun. 14 (1) (2023) 7913.

[46] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y.K. Wong, Z. Li, H. Zhao, Drivegpt4: Interpretable end-to-end autonomous driving via large language model, IEEE Robot. Autom. Lett. 9 (10) (2024) 8186–8193.

[47] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, Y. Qiao, Dilu: A knowledge-driven approach to autonomous driving with large language models, 2024, arXiv:2309.16292.

[48] R. Zhang, X. Guo, W. Zheng, C. Zhang, K. Keutzer, L. Chen, Instruct large language models to drive like humans, 2024, arXiv:2406.07296.

[49] M. Werling, J. Ziegler, S. Kammel, S. Thrun, Optimal trajectory generation for dynamic street scenarios in a frenét frame, in: 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 987–993.

[50] Y. Zhang, H. Sun, J. Zhou, J. Hu, J. Miao, Optimal trajectory generation for autonomous vehicles under centripetal acceleration constraints for in-lane driving scenarios, in: 2019 IEEE Intelligent Transportation Systems Conference, ITSC, 2019, pp. 3619–3626.

[51] M. Moghadam, A. Alizadeh, E. Tekin, G.H. Elkaim, An end-to-end deep reinforcement learning approach for the long-term short-term planning on the frenet space, 2020, arXiv:2011.13098.

[52] A. Dosovitskiy, An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.

[53] H. Liu, C. Li, Y. Li, Y.J. Lee, Improved baselines with visual instruction tuning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 26296–26306.

[54] Y. Zheng, Z. Xing, Q. Zhang, B. Jin, P. Li, Y. Zheng, Z. Xia, K. Zhan, X. Lang, Y. Chen, D. Zhao, Planagent: A multi-modal large language agent for closed-loop vehicle motion planning, 2024, arXiv:2406.01587.

[55] J. Mao, J. Ye, Y. Qian, M. Pavone, Y. Wang, A language agent for autonomous driving, 2024, arXiv:2311.10813.

[56] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, J.M. Alvarez, Is ego status all you need for open-loop end-to-end autonomous driving? in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2024, pp. 14864–14873.

[57] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, R. Urtasun, End-to-end interpretable neural motion planner, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019.

[58] P. Hu, A. Huang, J. Dolan, D. Held, D. Ramanan, Safe local motion planning with self-supervised freespace forecasting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 12732–12741.

[59] T. Khurana, P. Hu, A. Dave, J. Ziglar, D. Held, D. Ramanan, Differentiable raycasting for self-supervised occupancy forecasting, in: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, 2022, pp. 353–369.