CoLMDriver: LLM-based Negotiation Benefits Cooperative Autonomous Driving

Changxing Liu[1,2*] Genjia Liu[1,2*] Zijun Wang[1,2] Jinchang Yang[1,2] Siheng Chen[1,2†]
[1]Shanghai Jiao Tong University [2]Multi-Agent Governance & Intelligence Crew (MAGIC)
{cx-liu,LGJ1zed,wzjinsjtu,andreo_y,sihengc}@sjtu.edu.cn

Abstract

Vehicle-to-vehicle (V2V) cooperative autonomous driving holds great promise for improving safety by addressing the perception and prediction uncertainties inherent in single-agent systems. However, traditional cooperative methods are constrained by rigid collaboration protocols and limited generalization to unseen interactive scenarios. While LLM-based approaches offer generalized reasoning capabilities, their challenges in spatial planning and unstable inference latency hinder their direct application in cooperative driving. To address these limitations, we propose ***CoLMDriver****, the first full-pipeline LLM-based cooperative driving system, enabling effective language-based negotiation and real-time driving control. CoLMDriver features a parallel driving pipeline with two key components: (i) an LLM-based negotiation module under an critic-feedback paradigm, which continuously refines cooperation policies through feedback from previous decisions of all vehicles; and (ii) an intention-guided waypoint generator, which translates negotiation outcomes into executable waypoints. Additionally, we introduce* ***InterDrive****, a CARLA-based simulation benchmark comprising 10 challenging interactive driving scenarios for evaluating V2V cooperation. Experimental results demonstrate that CoLMDriver significantly outperforms existing approaches, achieving an 11% higher success rate across diverse highly interactive V2V driving scenarios. Code will be released on* `https://github.com/cxliu0314/CoLMDriver`.

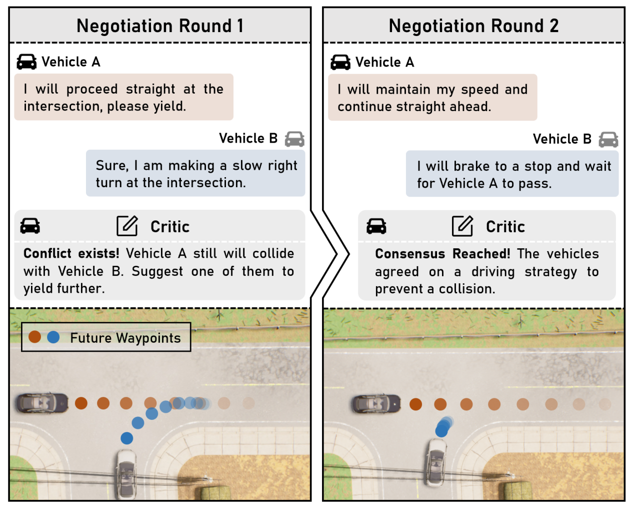


Figure 1. Negotiation with critic feedback. CoLMDriver refines cooperation policy by evaluating the negotiation outcomes.

1. Introduction

Vehicle-to-vehicle (V2V) cooperative autonomous driving (AD) aims to improve driving performance by allowing autonomous vehicles to communicate with surrounding vehicles. Unlike single-vehicle autonomous driving [1–5], where each vehicle makes driving decisions based solely on the observations from its own sensors, cooperative driving enables vehicles to exchange driving-related data [6, 7]. This collaborative information-sharing mechanism helps autonomous vehicles surmount the inherent limitations in single-vehicle driving, such as incomplete environmental perception [8–11] and uncertainty in forecasting the future states of surrounding traffic participants [12, 13].

Traditional cooperative driving approaches can be generally categorized into optimization-based and learning-based methods. Optimization-based cooperative driving methods [12, 14, 15] formulate multi-vehicle planning as constrained optimization problems to determine optimal actions. However, these methods depend on precise environmental modeling and require task-specific optimization objectives and constraints, making them inherently limited in handling unknown scenarios. Learning-based methods [16–18] employ reinforcement learning and imitation learning to develop cooperative driving policies. While these approaches have been applied to several driving tasks [19–21], they struggle with declined performance when encountering unseen multi-vehicle interaction patterns [22, 23]. These limitations underscore the exploration towards a more flex-

*Equal contribution
†Corresponding author

ible and generalizable cooperative driving framework.

Recently, large language models (LLMs) have gained significant attention in cooperative systems [24, 25] due to their remarkable reasoning abilities and vast knowledge. This advancement underscores the potential of LLM-based cooperative driving, where vehicles negotiate through natural language. Compared to optimization-based and learning-based approaches, LLM-based cooperation offers two key advantages. First, language-based cooperation offers greater flexibility compared to fixed-protocol communication [12], as it can incorporate both local motion details and global scene semantics. Second, with extensive pre-trained commonsense knowledge, LLMs have demonstrated strong capabilities in understanding traffic scenarios and making driving decisions [26–28]. This indicates their potential to handle diverse multi-vehicle driving scenarios, including complex cases such as navigating non-traffic-light intersections. However, integrating LLMs into cooperative driving faces three challenges. First, LLMs' limited ability to understand and plan in continuous road spaces makes direct application infeasible [29], requiring additional spatial information for effective cooperation. Second, redundant environmental information and unconstrained negotiation reduce efficiency, necessitating selective communication with relevant collaborators. Third, LLMs' long and unstable inference delays hinder high-frequency planning, demanding efficient negotiation and inference mechanisms to adapt to real-time control.

To address these challenges, we propose **CoLMDriver** (Cooperative Language-Model-based Driver), the *first* full pipeline (from sensor data to control signal) LLM-based cooperative driving system that accommodates real-time control with efficient planning negotiation. CoLMDriver consists of two parallel planning pipelines: i) *A cooperative planning pipeline*, implemented via an LLM-based negotiation module. ii) *An end-to-end driving pipeline* for real-time vehicle control, incorporated with an intention-guided waypoint planner to bridge language negotiation. To enhance the effectiveness and efficiency of negotiation, we propose three key techniques. First, we introduce an critic-feedback mechanism that evaluates negotiation outcomes and give feedback to the LLM-based negotiator, enabling continuous policy refinement as shown in Fig 1. This evaluation considers both high-level intentions and low-level waypoints, providing feedback from safety, efficiency, and multi-vehicle consensus perspectives. Second, we propose a dynamic grouping mechanism to select relevant collaborators for negotiation, improving efficiency by focusing on critical agents. Third, we integrate an auxiliary VLM-based intention planner to handle non-cooperation periods.

This system offers two key advantages. First, it effectively integrates LLM-based cooperative planning with fine-grained waypoint generation. LLM-derived driving intentions guide waypoints generation, and the waypoints provide feedback to refine cooperation strategies, forming an online optimization loop. Second, its parallel framework accommodates asynchronous planning, mitigating the inherent inference latency gap between the LLM and the end-to-end pipeline.

To evaluate performance in V2V scenarios, we introduce **InterDrive** (Interactive Driving) benchmark, which constructs 10 challenging traffic scenarios in the CARLA simulator [30]. These scenarios involve multiple autonomous vehicles with severely conflicting routes, testing an AD system's ability to handle highly interactive V2V situations. We evaluate CoLMDriver on both InterDrive benchmark and the public Town05 benchmark [2]. Results indicate that CoLMDriver surpasses existing single-vehicle and cooperative driving methods, achieving an 11% higher success rate across diverse scenarios.

To sum up, our contributions are:

- We propose CoLMDriver, the first full pipeline LLM-based cooperative driving system, featuring two main components: an LLM-based negotiator with critic-feedback, and an intention-guided waypoints planner to translate negotiation outcomes.
- We introduce InterDrive Benchmark, which includes 10 types of challenging scenarios to enable the evaluation of autonomous driving in handling V2V interactions.
- We conduct comprehensive experiments and validate that CoLMDriver achieves a superior success rate in various V2V driving scenarios.

2. Related works

2.1. End-to-end Autonomous Driving

A key research direction in end-to-end autonomous driving is imitation learning, which aims to replicate expert driving behaviors by fitting a model to driving data. Recent researches such as NEAT [31], TransFuser [2], UniAD [3], InterFuser [1], and ReasonNet [5], leverage transformer architectures to capture more nuanced representations of driving scenarios, enhancing the model's ability to process complex environments. Other approaches, like MP3 [32], UniAD, LAV [33], TCP [34], incorporate auxiliary tasks that provide additional learning signals to support the primary driving task, leading to better generalization. However, the IL approach struggles with low generalization to unseen scenarios and lacks causal reasoning. To overcome these issues, we propose an LLM-based method to achieve generalized reasoning ability in diverse interactive scenarios.

2.2. MLLMs-based Driving

In the field of autonomous driving (AD), recent research [26, 35–37] has integrated LLMs into AD systems to improve interpretability and facilitate human-like interactions. Some studies [28, 38–40] leverage VLMs to process multi-modal input data, providing both descriptive text and control signals suited to driving scenarios. LMDrive [27] integrates multi-modal sensor data with textual instructions, leveraging LLMs for closed-loop end-to-end AD.

Most current research focuses on using LLMs to enhance individual driving capabilities, while a few works explore driving cooperation. AgentsCoDriver [36] promotes lifelong memory updating through interaction with the environment, enabling simple negotiations between agents. CoDrivingLLM [41] centered around roadside units for vehicle-to-vehicle negotiations to resolve conflicts. However, both approaches are limited to discrete decision-making and cannot generate executable control signals. Co-MAL [42] employs LLMs to allocate roles and coordinate multi-agent planning. These works overlook the inference latency of LLM, making real-world deployment challenging. To bridge these gaps, we propose CoLMDriver, an LLM-based cooperative system that generates real-time driving signals through a parallel framework.

3. Problem Formulation

Consider N agents participate in the cooperation. Let $\mathcal{X}_i$ and $\mathcal{D}_i$ be the observation and the destination of the ith agent. The objective of collaborative driving is to achieve the maximized driving performance of all agents; that is,

$$\arg\max_{\theta,\mathcal{M}} \sum_{i=1}^{N} d(\Phi_\theta(\mathcal{X}_i, \mathcal{D}_i, \mathcal{M}_i^k)) \quad (1)$$

where $d(\cdot)$ is the driving performance metric, Φ is the driving framework with trainable parameter θ. $\mathcal{M}_i$ is the message exchange between agent i and other agents, which can iterate k rounds. Here we focus on leveraging the flexibility of language to achieve planning consensus and improve overall performance, where $\mathcal{M}_i^k = [\{\mathcal{M}_{i\leftrightarrow j}\}_{j=1}^{N}]^k$ represents a multi-round language-based negotiation process.

4. Methodology

This section introduces CoLMDriver, a cooperative driving system that leverages language-based negotiation and planning to enhance the collective driving capabilities of multiple autonomous vehicles. We start by outline the overall system architecture in Sec. 4.1, followed by detailed composition of two parallel pipelines in Sec. 4.2, 4.3.

4.1. Overall Architecture

As illustrated in Fig. 2, CoLMDriver operates through a parallel driving pipeline designed to tackle the latency challenges of negotiation without disrupting the normal execution of the downstream planner. The high-level guidance generation pipeline conducts reasoning and negotiation at a relatively low frequency to formulate comprehensive and consensus-driven driving intentions, while the low-level perception-planning-control pipeline runs at high frequency to ensure real-time vehicle control with guidance.

The high-level pipeline orchestrates cooperative decision-making through two core components: i) a LLM-based negotiation module under the critic-feedback paradigm, where LLMs enable multi-round negotiation between vehicles to reach a consensus on driving policy, guided by feedback from the evaluator; ii) a VLM-based intention planner, which generate high-level driving intentions by synthesizing multi-modal environmental context. The VLM intention planner continuously refines driving intentions based on textual descriptions of the current state, detected objects from the low-level perception module and the front camera input. If conflicts are predicted, the LLM negotiation module first conduct dynamic graph grouping with surrounding vehicles to form negotiation groups, and then takes current driving intention and engages in a multi-round negotiation process with guidance from evaluator. The negotiation results and intention guidance are then fed back into the low-level waypoint planner to guide precise planning.

The low-level pipeline follows the perception-planning-control structure. When receiving the sensor data, the perception module generates object-level 3D information and BEV perception features, conducting spatial understanding as auxiliary inputs for planning tasks. To translate language-based information into actionable waypoints, the key component is the intention-guided waypoint planner, which leverages both perception features and high-level planning intentions to generate waypoints. These waypoints are converted into control signals by the control module, resulting in improved cooperative driving outcomes.

4.2. High-level Guidance Pipeline

The high-level guidance pipeline is responsible for strategic decision-making and cooperative negotiation, enhancing driving adaptability through semantic reasoning and multi-agent consensus. It consists of two core components: the VLM-based intention planner and the LLM-based negotiation module. The negotiation results guide the low-level planner during the negotiation process, while the VLM output takes precedence when no negotiation is activated.

4.2.1. LLM-based Negotiation Module

The LLM-based negotiation module engages in multiple rounds of dialogue with surrounding intelligent vehicles, resolving predicted conflicts by reaching a consensus on driving policies. Given the negligible latency in LLM inference, the negotiation system focuses on how to efficiently achieve consensus on an optimized driving policy. To ensure the generalizability of negotiations, we avoid imposing strict output formats or rigid communication rules. However, overly unrestricted negotiations may struggle to converge on a consensus. The key innovation lies in incorporating an **critic-feedback** paradigm within the negotiation system. The critic-feedback paradigm is inspired by the Actor-

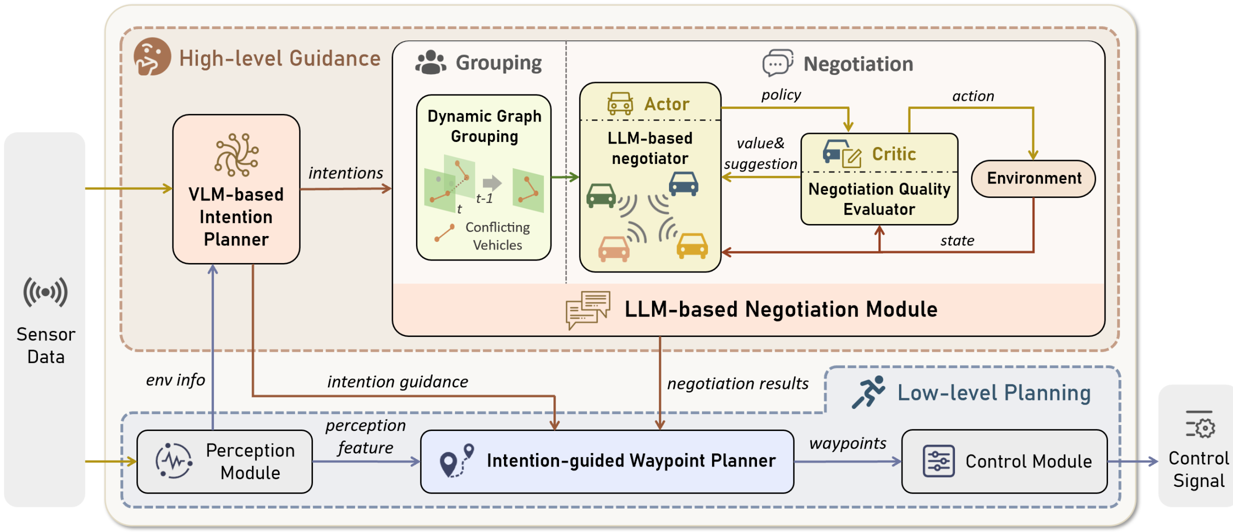


Figure 2. Overall architecture of CoLMDriver. CoLMDriver operates through a parallel driving pipeline, where language negotiation assists in the planning process through asynchronous connection of three component: i) an LLM-based negotiation module under the critic-feedback paradigm; ii) a VLM-based intention planner and iii) an intention-guided waypoint planner.

Critic approach in reinforcement learning where the ”actor” selects actions based on current policies, while the ”critic” evaluates the chosen actions by providing feedback on their quality, enabling faster convergence towards optimal outcomes. In our method, the evaluator act as the critic. By providing feedback based on dialogue quality, safety, and efficiency expectations, we leverage the in-context learning capabilities of LLMs to facilitate rapid convergence in the negotiation process. The LLM-based negotiation module consists of three main components: i) The dynamic graph grouping mechanism, which identifies agents with negotiation needs and establishes communication in dynamic traffic scenarios, ii) The LLM-based negotiator, which conducts negotiations with grouped agents using natural language and iii) The negotiation quality evaluator, which acts as a critic, providing feedback to the negotiator to accelerate consensus achievement.

Overall Process. Once the negotiation process begins, vehicles first form negotiation groups using a dynamic graph grouping mechanism. In each round, vehicles take turns ”speaking” in a designated order. The negotiation quality evaluator then assesses the situation, providing feedback on consensus, safety, and efficiency. The LLM-based negotiators incorporate this feedback into their input, adjust their driving intentions accordingly, and call the evaluator again. After several rounds, when the evaluator determines that consensus has been reached, the negotiation concludes, and the final driving intentions are passed on to the downstream planners of each vehicle.

Dynamic Graph Grouping Mechanism. It is crucial for vehicles to determine who and when to communicate with. To address this challenge, we prioritize vehicle groups that are most likely to conflict and build communication graph to promote effective negotiation. We assume that vehicles can automatically establish communication within the range of their hardware and are capable of broadcasting essential information, such as their planned future waypoints.

To better clarify the mutual influence between vehicles, we conduct dynamic grouping by constructing a spatiotemporal vehicle graph. Each vehicle is treated as a node, and vehicles that could potentially conflict in the future are connected by edges, which are calculated based on their safety scores derived from their waypoints. At any given moment, we build the spatial vehicle graph and apply Depth-First Search (DFS) to gather all connected vehicles into groups. To avoid inconsistent driving policies due to the constantly changing nature of dynamic groups, we preserve historical groups and merge intersecting groups across temporal dimension, obtaining a comprehensive grouping result. The communication graph $\mathcal{G}$ at time T is constructed iterative:

$$\mathcal{G} = \mathcal{H}^T, \quad \mathcal{H}^t = \Phi(\mathcal{H}^{t-1} \cup \mathcal{C}^t) \tag{2}$$

$$\mathcal{C}^t = \bigcup_k \mathrm{DFS}(V^t, \{(v_i, v_j) \mid S_s(v_i, v_j) \geq \theta\}) \tag{3}$$

where safety score S_s determines edges and $\Phi(\cdot)$ merges all groups that intersect between history group $\mathcal{H}^{t-1}$ and current group $\mathcal{C}^t$. Negotiations are then carried out within each group, allowing for local optimization of driving policies, which contributes to improved overall performance.

LLM-based negotiator. The LLM-based negotiator conducts human-like language negotiation with other vehicles in the group. Inputs include ego vehicle's current speed, intention, other cars' broadcast information, history conversation and critic's suggestion if exist. Since the inference time of an LLM is proportional to the output length, we have carefully designed the prompts to ensure concise information transmission and employed prompt caching techniques to maintain timeliness. The LLM-based negotiator integrates the shared information from group members, consider past conversations, and combine feedback from evaluators to output information that may include self actions, requests or responses to others. In a group that has n vehicles, the negotiator output of the i_{th} vehicle at round K is:

$$O_{\mathrm{LLM}_i^K} = LLM_i(f_{\mathrm{P}}(\bigcup_{j=0}^{n} I_j, \bigcup_{k=0}^{K}\bigcup_{j=0}^{n} O_{\mathrm{LLM}_j^k}, S^{K-1})) \quad (4)$$

where I denotes the current information shared by vehicles in the group, including speed, intention, and position, and S the evaluator's suggestion, f_{P} the prompt generation process. Since the used LLM is not trained on a specific domain, this paradigm differs from previous multi-vehicle cooperative driving approaches by not requiring each vehicle to be equipped with a specific model, demonstrating the versatility and broad applicability of LLM.

Negotiation Quality Evaluator. The negotiation quality evaluator acts as a critic, assessing the negotiation performance based on future planning and generating feedback related to consensus, safety, and efficiency concerns. The evaluation process follows three key steps: sum, score, and criticize. To initiate the evaluation, the evaluator can be activated on a random vehicle within the group. Based on the current round conversation, the evaluator first sums each vehicle's actions using LLM, transforming them into driving intention formats, and then distributes the results to all vehicles. Each vehicle's waypoint planner uses the summed intentions as input, generates planned waypoints, and broadcasts these plans to assist in the evaluation. The evaluator conducts the scoring process by assessing three key aspects — consensus, safety, and efficiency. Consensus score S_c is judged by LLM, indicating whether every vehicle in the group is willing to execute the reached policy. Both safety score S_s and efficiency score S_e are derived from the waypoints, calculated by the carefully designed formula:

$$S_c^k = LLM_c(\bigcup_{j=0}^{n} O_{\mathrm{LLM}_j^k}), [S_s^k, S_e^k] = \mathcal{F}(\bigcup_{j=0}^{n} W_j) \quad (5)$$

where W is the predicted waypoints and $\mathcal{F}$ the score calculation formula. Finally, the evaluator provides feedback $\mathcal{R}$ through a classifier Ψ, criticizing scores that fail to meet the required standards.

$$\mathcal{R} = \Psi(S_c^k, S_s^k, S_e^k) \quad (6)$$

This criticism is used as input for the next round of negotiation, guiding the system towards an optimal driving policy by encouraging faster convergence.

4.2.2. VLM-based Intention Planner

The VLM-based intention planner utilizes the generalized knowledge embedded in language models to recognize unusual objects and deal with complex scenes, providing more holistic decision support. The focus is to provide optimal high-level driving intention to accurately guide the downstream planner. To comprehensively and efficiently activate the understanding and decision-making capabilities of the VLM-based intention planner, we have carefully designed a hierarchical prompt generation process and limited output format. The prompt contains perception results written in an intelligible format, providing accurate environment information. To collect reasonable driving intention in different environments, we use V2Xverse [8] platform and employ an expert agent [1] to record driving data, capturing a wide range of urban scenarios. Driving intentions are defined as navigation intentions and speed intentions. Navigation intentions are derived from the ground truth navigation instructions, while speed intentions are extracted from the expert's driving speed. To adapt the VLM to the specific task of driving intention assessment, we utilize the processed driving data for transfer learning based on LoRA.

4.3. Low-level Planning Pipeline

The low-level planning pipeline focuses on real-time execution, translating high-level intentions into geometrically feasible trajectories and control commands. The key component is the intention-guided waypoint planner, efficiently conduct precise planning guided by driving intentions.

4.3.1. Intention-guided Waypoint Planner

The Intention-guided waypoint planner acts a bridge connecting high-level driving intentions and low-level implementation paths. The challenge lies in how to precisely map high-level intentions to specific scenarios as usable waypoints. Our design consists of two main parts: intention-to-waypoint data generation and the model structure.

Intention-to-waypoint Data Generation. To achieve precise intention-guided waypoints generation, we use waypoints of expert agent as a reference and generate waypoints that align with the intended action while satisfying practical scenario constraints. Based on the observation that acceleration is influenced by surrounding objects density, we extract the actual waypoints of the referenced vehicle and interpolate them using an environment-adaptive acceleration model, which generates elaborate waypoints corresponding to different driving intentions. Given a ground-truth waypoints W, the data generation process can be expressed as $W_g = \Phi(W, a)$. Here, the acceleration $a = f(I, x, \sigma)$ is guided by the intention I and generated by the environment-adaptive acceleration model f, considering the distance x to the nearest vehicle and the vehicle density σ. The generated waypoints W_g is interpolated by function Φ, which

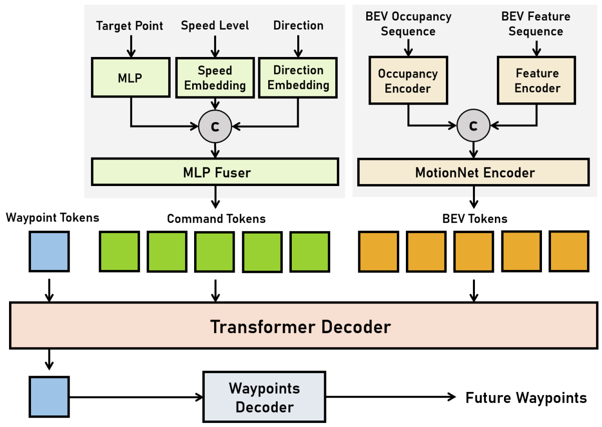


Figure 3. Model architecture of the low-level Transformer-based intention-guided waypoint planner.

conforms to driving norms and adapts to environment.

Model Structure. To ensure waypoints align with different driving intentions within the same scenario, we developed a Transformer-based, intention-guided waypoint planner, as shown in Fig. 3. The model effectively takes input from the BEV occupancy map and BEV features from previous frames, which are processed by the MotionNet [43] encoder to capture the environmental context. Additionally, goal-oriented inputs, including target points, navigation intentions, and speed intentions, are fused through a MLP Fuser to form the guidance context. A multi-layer Transformer decoder performs cross-attention between a waypoint query and the environmental/guidance contexts, followed by a Waypoints Decoder to generate a sequence of waypoints. These waypoints are then passed to the control module to produce the necessary control signals.

5. InterDrive Test Benchmark

To evaluate the capabilities of autonomous driving systems in handling multi-vehicle interaction, we present the InterDrive benchmark on top of V2Xverse simulation platform. This benchmark encompasses 10 types of typical multi-vehicle scenarios, each involving multiple under-test vehicles. We assign these vehicles with largely overlapped target paths to encourage conflicts, and randomly deploy additional traffic participants as obstacles. These scenarios are constructed to simulate realistic traffic scenarios where several on-road vehicles are autonomous-driven.

5.1. Scenarios

Fig 4 visualizes the 10 scenarios in InterDrive Benchmark, where we construct traffic scenarios with reference to the pre-crash typology of the US National Highway Traffic Safety Administration. We assess three typical scenarios in handling multi-vehicle interaction, including intersection crossing, lane merging, and lane changing.

***Intersection Crossing (IC).** Several vehicles enter, meet, and then exit an intersection from different directions. Four distinct types of scenarios are incorporated, shown in Fig 4(a)-(d). To ensure comprehensive evaluation diversity, we carefully design different combinations of entry and exit directions for the vehicles at the intersection.

***Lane Merging (LM).** Vehicles merge into the same lane from different directions, see in Fig 4(e)-(h). We construct scenarios in different road topologies, including parallel straight-ahead lanes, T-junctions, and highway ramps.

***Lane Changing (LC).** Multiple vehicles initially maintain parallel trajectories while traveling in the same direction. Subsequently, they are required to execute lane-changing maneuvers, intersecting the trajectories of adjacent vehicles to reach their respective destinations. See in Fig 4(i),(j).

InterDrive benchmark extends each scenario through diverse configurations, varying in route waypoints, the number of vehicles under test, and additional obstacles, ultimately generating 92 distinct test tasks. The number of interactive test vehicles is configured to range from 2 to 8, simulating the typical number of vehicles with which a single vehicle may have direct conflicts simultaneously.

5.2. Metrics

InterDrive incorporates five metrics: Route Completion, Infraction Score, Driving Score (adopted from CARLA Leaderboard [44]), and an additional metric - Success Rate.

Route Completion (RC) is the percentage of the total planned route distance completed by the under-test vehicles.

Infraction Score (IS) starts at 1.0 for each task and is reduced upon collisions by a predefined discount factor, serving to evaluate the safety performance of all test vehicles.

Driving Score (DS) serves as the primary ranking metric, and is calculated as the product of Route Completion and Infraction Score, capturing both task progress and safety.

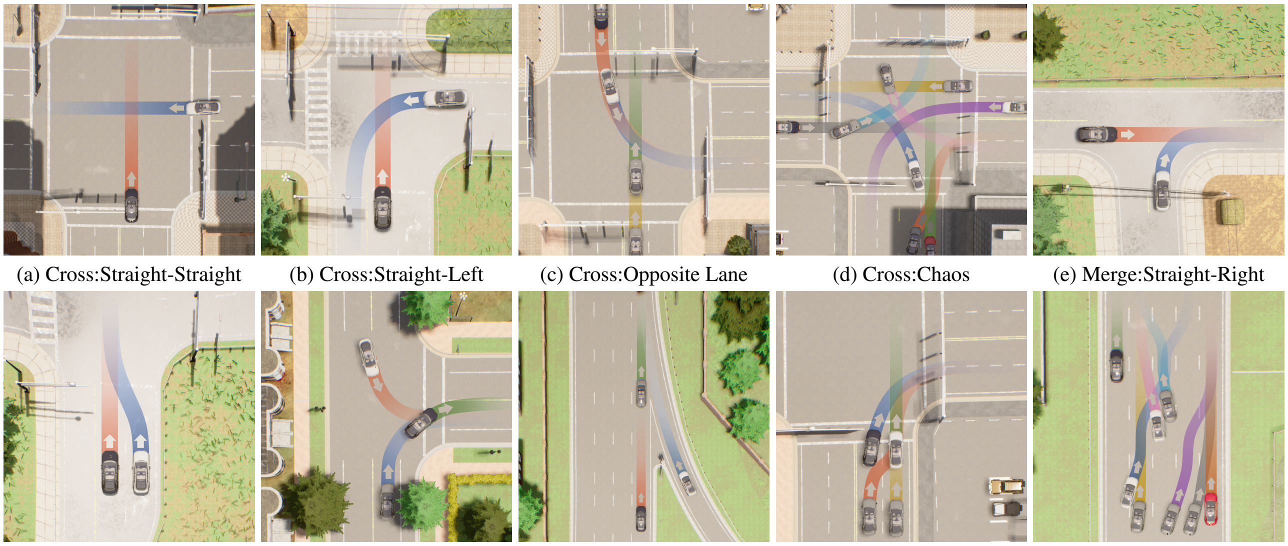
Success Rate (SR) is the percentage of tasks completed with a full-mark Driving Score, reflecting the consistency of the system to achieve reliable driving performance.

These metrics collectively provide a comprehensive view of navigation performance in multi-vehicle driving scenarios.

6. Experiments

6.1. Experimental Settings

We implement and evaluate our method on the CARLA simulator of version 0.9.10.1 [30]. The ideal simulation frequency is set to 5 Hz for all experiments except latency-aware ablation. For the low-level pipeline in the CoLM-Driver, we deploy PointPillars [45] to encode point clouds. We use Lora finetuning [46] for InternVL2-4B [47] as the VLM intention planner and Qwen2.5-3B [48] as the LLM negotiator. For the intention-guided waypoint planner, we use 256 for embedding size and medium feature size, with



(a) Cross:Straight-Straight (b) Cross:Straight-Left (c) Cross:Opposite Lane (d) Cross:Chaos (e) Merge:Straight-Right

(f) Merge:Neighbor Lane (g) Merge:Left-Right (h) Merge:Highway (i) Change:Right-Straight (j) Change:Highway

Figure 4. The 10 types of scenarios in the proposed InterDrive benchmark. These scenarios evaluate the three key skills in handling interaction among nearby vehicles, including going cross intersections (a-d), lane merging (e-h), and lane changing (i-j).

Table 1. Driving performance in InterDrive Benchmark. CoLMDriver achieves the highest success rate in all scenarios.

Method	InterDrive-total				InterDrive-IC				InterDrive-LM				InterDrive-LC			
	DS↑	RC↑	IS↑	SR↑	DS↑	RC↑	IS↑	SR↑	DS↑	RC↑	IS↑	SR↑	DS↑	RC↑	IS↑	SR↑
VAD[4]	25.37	75.00	0.33	0.02	15.49	54.72	0.29	0.00	37.24	92.85	0.40	0.05	17.93	76.00	0.26	0.00
UniAD[3]	35.17	88.30	0.38	0.11	37.24	91.63	0.41	0.11	42.50	84.41	0.47	0.15	12.19	90.57	0.12	0.00
TCP [34]	73.68	90.54	0.82	0.50	77.64	82.83	**0.94**	0.50	82.18	95.18	0.86	0.70	43.52	96.30	0.45	0.00
LMDrive [27]	48.83	58.02	0.85	0.20	44.72	57.94	0.79	0.17	60.88	69.43	0.86	0.30	27.96	29.70	**0.95**	0.00
CoDriving [8]	74.13	**96.31**	0.76	0.57	66.32	90.57	0.71	0.61	96.18	**100.00**	0.96	0.75	36.57	**100.00**	0.37	0.00
Rule-based	78.38	91.85	0.80	0.72	80.06	**95.93**	0.81	**0.72**	94.44	**100.00**	0.94	0.90	34.43	62.29	0.42	0.25
CoLMDriver(Ours)	**88.53**	94.05	**0.90**	**0.80**	**82.07**	88.78	0.86	**0.72**	**98.27**	99.93	**0.98**	**0.92**	**59.21**	82.50	0.597	**0.50**

Table 2. Ablation study of system components. *Nego* : negotiation, *S/E* :safety/efficiency score, *Cons*: consensus score.

ID	Nego	Grouping	Critic S/E	Critic Cons	DS↑	RC↑	IS↑	SR↑
1					47.64	**96.43**	0.485	0.130
2	✓				9.33	10.37	0.935	0.000
3	✓	✓			77.29	95.22	0.784	0.652
4	✓	✓	✓		83.46	91.93	0.860	0.739
5	✓	✓	✓	✓	**88.53**	94.05	**0.903**	**0.804**

20 output waypoints at 5 Hz. The communication range is 70m, and the maximum negotiation rounds is 3.

6.2. Quantitative Results of Closed-loop driving

Performance in Highly Interaction Traffic Scenarios. Tab. 1 presents CoLMDriver's driving performance in our proposed InterDrive Benchmark (no NPC), compared to other advanced closed-loop driving baselines, including TCP [34], VAD [4], UniAD [3], CoDriving [8], and another VLM-based method, LMDrive [27]. To prove the necessity of negotiation, we build the Rule-based method as comparison, which arranges the order of vehicle passage based on traffic rules. Optimization-based cooperative planning methods are not compared due to being closed-source or on other platform. The table shows the overall score under InterDrive and separate performance for InterDrive-IC, InterDrive-LM and InterDrive-LC. CoLMDriver achieves SOTA performance on driving score(DS) across all interactive scenarios due to its language negotiation capability, outperforming other baselines by at least 10.15% in DS. The three cooperative driving methods all outperform single-agent driving approaches, demonstrating the effectiveness of cooperation in conflict resolution. Other baselines face challenges such as target recognition issues, leading to lower route completion(RC), or collision incidents due to the lack of negotiation, resulting in low infraction scores(IS). TCP achieves a relatively high driving score but struggles with a low success rate (SR), indicating frequent collisions among scenes. LMDrive benefits from its multi-view, multimodal input and VLM capability, achieving a high infraction score, but encounters challenges

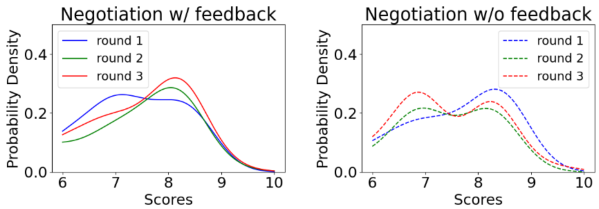


Figure 5. Experiment of negotiation convergence guided by critic-feedback.

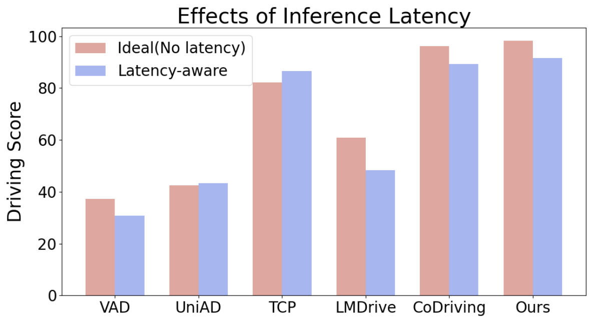


Figure 6. The driving performance with/without (Latency-aware/Ideal) accounting for inference latency.

in driving interruption where two cars come to a stop due to close proximity, each yielding to the other without progressing. Both intention-conflict collisions and dual-yielding issues can be resolved through language negotiation. Performances on InterDrive (with NPC) is shown in appendix.

Consensus Convergence. Fig. 5 presents the negotiation quality score distribution of the evaluator for system with or without critic-feedback. When the LLM updates its negotiation messages based on conversation alone, the negotiation quality score fluctuates randomly across rounds. However, when guided by evaluator feedback, the score exhibits a steady increase as negotiation iterates.

System Component Ablation. Tab. 2 evaluates the impact of different system components on performance. A system without negotiation (ID 1) performs closely to LMDrive on DS, demonstrating solid baseline performance. However, negotiation without the dynamic grouping mechanism leads to continuous stopping, resulting in lower route completion. Incorporating the critic-feedback into the negotiation module further enhances the driving score.

Latency-aware Ablation. We compare the performance in ideal computing situation(no inference latency) and situation with inference latency (on RTX 3090) in Fig. 6 on the InterDrive-LM. Our CoLMDriver experiences only a 6.62% drop in driving score and still keeps driving score over 90, demonstrating the inference efficiency of the proposed system. In our framework, the low-level planning pipeline can continuously generate precise execution based on intention guidance within varying environment. TCP, operating faster than our ideal simulation, slightly increase its performance. See more details in appendix.

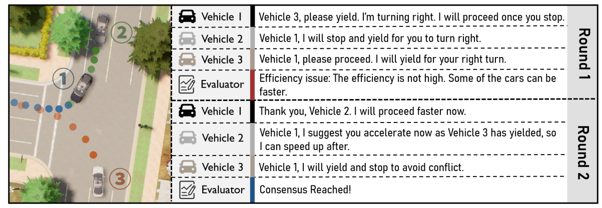


Figure 7. Qualitative Result.

Qualitative Result. Figure 7 illustrates a representative negotiation process in a Merge: Left-Right scenario involving 3 vehicles. Initially, the vehicles are in conflict. In the first negotiation round, the solution fails to meet the efficiency requirement. However, in the second round, the agents achieve improved efficiency while maintaining safety and consensus, thanks to the critic-feedback mechanism.

Performance on public benchmark. We further investigate the general navigation capability of CoLMDriver on the public Town05 benchmark [2]. To enable V2V communication in this single-vehicle driving benchmark, we enable the surrounding vehicles to transmit their driving intention to the ego vehicle but do not change their own behaviors. Tab. 3 compares CoLMDriver with two SOTA single-vehicle driving methods. We can see that CoLMDriver achieves a superior Driving Score in both long and short routes, and surpasses ReasonNet by 11% in Town05 Long. This is because CoLMDriver receives driving intention from neighbors, reducing the uncertainty in planning.

Table 3. Driving performance on Town05 benchmark [2]

Method	Town05 Short		Town05 Long	
	DS↑	RC↑	DS↑	RC↑
InterFuser [1]	94.95	95.19	68.31	94.97
ReasonNet [5]	95.71	96.23	73.22	95.88
CoLMDriver(Ours)	**96.14**	**96.45**	**81.49**	**96.72**

7. Conclusion and Limitation

In this paper, we present CoLMDriver, an innovative autonomous driving system that leverages multimodal LLMs for effective language-based cooperative planning in end-to-end driving. CoLMDriver utilizes multi-round negotiation to achieve consensus in highly interactive scenarios, and employs high-level driving intention to guide low-level waypoints generation. Meanwhile, we construct the InterDrive Benchmark to evaluate autonomous driving systems in highly interactive environments. Extensive closed-loop experiments demonstrate the effectiveness of CoLMDriver, highlighting the significant potential of language-based negotiation for advancing cooperative driving. One current limit is the diversity of language interaction demonstrations, which we aim to expand in future work by constructing more complex and interactive scenarios, further enhancing the system's capability and adaptability.

References

[1] Hao-Chiang Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Tang Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, 2022.

[2] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7073–7083, 2021.

[3] Yi Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wen Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. 2022.

[4] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[5] Hao Shao, Letian Wang, Ruobing Chen, Steven L. Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning, 2023.

[6] Ruiqi Zhang, Jing Hou, Florian Walter, Shangding Gu, Jiayi Guan, Florian Röhrbein, Yali Du, Panpan Cai, Guang Chen, and Alois Knoll. Multi-agent reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2408.09675*, 2024.

[7] Yushan Han, Hui Zhang, Huifang Li, Yi Jin, Congyan Lang, and Yidong Li. Collaborative perception in autonomous driving: Methods, datasets, and challenges. *IEEE Intelligent Transportation Systems Magazine*, 15(6):131–151, 2023.

[8] Genjia Liu, Yue Hu, Chenxin Xu, Weibo Mao, Junhao Ge, Zhengxiang Huang, Yifan Lu, Yinda Xu, Junkai Xia, Yafei Wang, et al. Towards collaborative autonomous driving: Simulation platform and end-to-end system. *arXiv preprint arXiv:2404.09496*, 2024.

[9] Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in Neural Information Processing Systems*, 2022.

[10] Yifan Lu, Yue Hu, Yiqi Zhong, Dequan Wang, Siheng Chen, and Yanfeng Wang. An extensible framework for open heterogeneous collaborative perception. *arXiv preprint arXiv:2401.13964*, 2024.

[11] Sizhe Wei, Yuxi Wei, Yue Hu, Yifan Lu, Yiqi Zhong, Siheng Chen, and Ya Zhang. Asynchrony-robust collaborative perception via bird's eye view flow. *Advances in Neural Information Processing Systems*, 36, 2024.

[12] Haichao Liu, Zhenmin Huang, Zicheng Zhu, Yulin Li, Shaojie Shen, and Jun Ma. Improved consensus admm for cooperative motion planning of large-scale connected autonomous vehicles with limited communication. *IEEE Transactions on Intelligent Vehicles*, 2024.

[13] Chaoyi Chen, Qing Xu, Mengchi Cai, Jiawei Wang, Jianqiang Wang, and Keqiang Li. Conflict-free cooperation method for connected and automated vehicles at unsignalized intersections: Graph-based modeling and optimality analysis. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):21897–21914, 2022.

[14] Zhenmin Huang, Shaojie Shen, and Jun Ma. Decentralized ilqr for cooperative trajectory planning of connected autonomous vehicles via dual consensus admm. *IEEE Transactions on Intelligent Transportation Systems*, 24(11):12754–12766, 2023.

[15] Zhenmin Huang, Haichao Liu, Shaojie Shen, and Jun Ma. Parallel optimization for cooperative autonomous driving at unsignalized roundabouts with hard safety guarantees. *arXiv e-prints*, pages arXiv–2303, 2023.

[16] Rui Zhao, Yun Li, Fei Gao, Zhenhai Gao, and Tianyao Zhang. Multi-agent constrained policy optimization for conflict-free management of connected autonomous vehicles at unsignalized intersections. *IEEE Transactions on Intelligent Transportation Systems*, 25(6):5374–5388, 2023.

[17] Zhi Zheng and Shangding Gu. Safe multi-agent reinforcement learning with bilevel optimization in autonomous driving. *IEEE Transactions on Artificial Intelligence*, 2024.

[18] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv preprint arXiv:1910.03088*, 2019.

[19] Shanxing Zhou, Weichao Zhuang, Guodong Yin, Haoji Liu, and Chunlong Qiu. Cooperative on-ramp merging control of connected and automated vehicles: Distributed multi-agent deep reinforcement learning approach. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 402–408. IEEE, 2022.

[20] Jiaqi Liu, Peng Hang, Xiaoxiang Na, Chao Huang, and Jian Sun. Cooperative decision-making for cavs at unsignalized intersections: A marl approach with attention and hierarchical game priors. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[21] Dong Chen, Kaixiang Zhang, Yongqiang Wang, Xunyuan Yin, Zhaojian Li, and Dimitar Filev. Communication-efficient decentralized multi-agent reinforcement learning for cooperative adaptive cruise control. *IEEE Transactions on Intelligent Vehicles*, 2024.

[22] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

[23] Dibya Ghosh, Jad Rahme, Aviral Kumar, Amy Zhang, Ryan P Adams, and Sergey Levine. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in neural information processing systems*, 34:25502–25515, 2021.

[24] Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9, 2024.

[25] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024.

[26] 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.

[27] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng 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*, pages 15120–15130, 2024.

[28] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023.

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

[30] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio M. López, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, 2017.

[31] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15773–15783, 2021.

[32] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14398–14407, 2021.

[33] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17201–17210, 2022.

[34] Peng Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 2022.

[35] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023.

[36] Senkang Hu, Zhengru Fang, Zihan Fang, Yiqin Deng, Xianhao Chen, and Yuguang Fang. Agentscodriver: Large language model empowered collaborative driving with lifelong learning. *arXiv preprint arXiv:2404.06345*, 2024.

[37] 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.

[38] Katrin Renz, Long Chen, Ana-Maria Marcu, Jan Hünermann, Benoit Hanotte, Alice Karnsund, Jamie Shotton, Elahe Arani, and Oleg Sinavski. Carllava: Vision language models for camera-only closed-loop driving. *arXiv preprint arXiv:2406.10165*, 2024.

[39] Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*, 2023.

[40] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024.

[41] Shiyu Fang, Jiaqi Liu, Mingyu Ding, Yiming Cui, and Chen Lv. Towards interactive and learnable cooperative driving automation: a large language model-driven decision-making framework. *arXiv preprint arXiv:2409.12812*, 2024.

[42] Huaiyuan Yao, Longchao Da, Vishnu Nandam, Justin Turnau, Zhiwei Liu, Linsey Pang, and Hua Wei. Comal: Collaborative multi-agent large language models for mixed-autonomy traffic. In *Proceedings of the 2025 SIAM International Conference on Data Mining (SDM)*, pages 409–418. SIAM, 2025.

[43] Pengxiang Wu, Siheng Chen, and Dimitris N Metaxas. Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11385–11395, 2020.

[44] CARLA leaderboard. https://leaderboard.carla.org/leaderboard/.

[45] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12689–12697, 2018.

[46] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[47] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024.

[48] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.