DriveGPT4-V2: Harnessing Large Language Model Capabilities for Enhanced Closed-Loop Autonomous Driving

Zhenhua Xu[1,2] Yan Bai[3] Yujia Zhang[1] Zhuoling Li[1] Fei Xia[3]
Kwan-Yee K. Wong[1] Jianqiang Wang[2] Hengshuang Zhao[1*]
[1]The University of Hong Kong [2]Tsinghua University [3]Meituan

Abstract

Multimodal large language models (MLLMs) possess the ability to comprehend visual images or videos, and show impressive reasoning ability thanks to the vast amounts of pretrained knowledge, making them highly suitable for autonomous driving applications. Unlike the previous work, DriveGPT4-V1, which focused on open-loop tasks, this study explores the capabilities of LLMs in enhancing closed-loop autonomous driving. DriveGPT4-V2 processes camera images and vehicle states as input to generate low-level control signals for end-to-end vehicle operation. A multi-view visual tokenizer (MV-VT) is employed enabling DriveGPT4-V2 to perceive the environment with an extensive range while maintaining critical details. The model architecture has been refined to improve decision prediction and inference speed. To further enhance the performance, an additional expert LLM is trained for online imitation learning. The expert LLM, sharing a similar structure with DriveGPT4-V2, can access privileged information about surrounding objects for more robust and reliable predictions. Experimental results show that DriveGPT4-V2 outperforms all baselines on the challenging CARLA Longest6 benchmark. The code and data of DriveGPT4-V2 will be publicly available.

1. Introduction

Autonomous driving has seen rapid advancements over the past decade in both academia and industry [31, 56]. Today, autonomous vehicles are deployed commercially in diverse applications such as robotaxi services [62], goods delivery [33], and security patrols [45]. Additionally, assisted driving systems are increasingly integrated into consumer vehicles, enhancing safety and convenience. However, the majority of current autonomous driving systems still rely heavily on rule-based decision-making modules, where planning and control are governed by predefined algorithms and rules. This reliance restricts the ability of these systems to operate effectively in varied, real-world scenarios, limiting the broader advancement of autonomous driving.

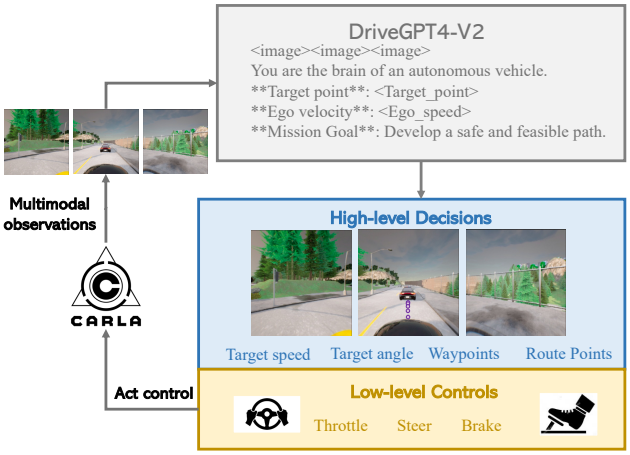


Figure 1. DriveGPT4-V2 for closed-loop autonomous driving. Taken as input multi-view camera images and vehicle state information, DriveGPT4-V2 predicts high-level vehicle decisions and converts them to low-level vehicle control signals in an end-to-end manner. DriveGPT4-V2 presents outstanding effectiveness and efficiency, serving as a reliable baseline method for future research on autonomous driving with LLMs.

In response, recent research has shifted toward end-to-end autonomous driving systems based on learning-based methods [8, 9, 12, 17, 21, 23, 45]. These methods enable more streamlined and intelligent system designs by directly learning control from input sensor data, without the need for complicated intermediate modules. From a learning theory perspective, end-to-end autonomous driving can optimize the entire system on the final outputs, rather than through the isolated optimization of individual modules, which potentially improves overall performance.

The rise of large language models (LLMs) has demonstrated their impressive ability to perform natural language processing (NLP) tasks [1, 2, 13, 34, 49]. Commercial LLMs, such as ChatGPT [34] and Claude3 [1], are already

*Corresponding author.

widely used in daily life. Motivated by their success in NLP tasks, LLMs are being extended to multimodal domains, enabling more generalized tasks in fields like image analysis [28–30], video comprehension [18, 26], and embodied AI [24, 25, 46, 55]. Given their versatility, multimodal LLMs have been applied to autonomous driving for tasks such as interpretability [15, 36, 44, 54], vehicle control [10, 27, 27], and risk assessment [3, 60]. While multimodal LLMs have demonstrated promising vehicle control performance due to their extensive pretrained knowledge, previous research has primarily concentrated on open-loop settings, which are not implemented for real-time vehicle control. Open-loop systems lack feedback mechanisms, making them less adaptable to changes and disturbances, limiting their applicability in real-world applications. Thus, multimodal LLMs need to be designed and evaluated specifically for closed-loop autonomous driving scenarios [27, 39, 43, 51].

Behavior cloning [35, 48], the most commonly employed imitation learning algorithm, is used for most end-to-end autonomous driving works [44, 54]. However, models trained only by behavior cloning can encounter drifting issues in closed-loop tasks, where errors accumulate, potentially leading the vehicle into unsafe situations, such as deviating from the path, colliding, or getting stuck.

To address the aforementioned issue, we incorporate online (or on-policy) imitation learning [5, 40, 41] into closed-loop autonomous driving. Leveraging the reasoning capabilities and pretrained knowledge of LLMs, we propose using a multimodal LLM as a fully automatic expert for online imitation learning without human efforts. Let M_E and M_A denote the model of expert and agent, respectively. M_E can access privileged information about nearby objects while M_A cannot, thus presenting a more powerful and reliable performance. After being trained by behavior cloning, M_A is deployed on the training environment for data aggregation. M_A might make incorrect predictions that lead the vehicle into exception states. M_E can provide M_A with on-policy supervision and take over the vehicle if the error made by M_A is large enough. When M_E takes over the vehicle, it collects one data sample at the moment and aggregates it into the dataset. The aggregated dataset is leveraged to further train the agent model to handle errors.

In this paper, we introduce DriveGPT4-V2, an end-to-end autonomous driving system. DriveGPT4-V2 extends DriveGPT4-V1 [54] to closed-loop tasks and integrates online imitation learning for enhanced robustness and intelligence. The "4" in the system's name signifies its multimodal capacity, inspired by the MiniGPT4 model [65]. DriveGPT4-V2 is powered by multimodal LLMs, enabling it to directly generate low-level vehicle control signals (i.e., throttle, brake and steer) based on multimodal input data (i.e., vehicle states and camera images). The structure of DriveGPT4-V2 is refined for our task, including a multi-view visual tokenizer (MV-VT) that possesses a large enough perception range without losing details, as well as output decision heads (DeciHeads) specially designed for numerical vehicle decision prediction. An additional multimodal LLM with access to privileged information (e.g., ground truth of nearby objects in text) serves as an expert for online imitation learning, which is inspired by the LBC framework [5]. DriveGPT4-V2 achieves state-of-the-art performance on the challenging CARLA Longest6 benchmark [12], which evaluates systems over 36 long routes demanding high levels of intelligence and reliability. The overview of DriveGPT4-V2 is visualized in Fig. 1. The contributions of this paper are:

- We extend DriveGPT4-V1, an open-loop autonomous driving system with multimodal LLMs, and propose DriveGPT4-V2 for end-to-end closed-loop autonomous driving. DriveGPT4-V2 can better harness the capability of LLMs and be deployed for continuous vehicle control.
- We introduce the use of LLMs with privileged information as an expert for online (on-policy) imitation learning, facilitating the generation of labeled data for handling errors, which significantly enhances the performance and robustness of closed-loop autonomous driving.
- We optimize the model structure of MLLMs, including a multi-view visual tokenizer (MV-VT) that possesses a large perception range and preserves details, as well as output decision heads (DeciHeads) specially designed for numerical vehicle decision prediction.
- DriveGPT4-V2 demonstrates superior performance on the CARLA Longest6 benchmark, substantially outperforming existing methods.

2. Related Works

Large language models for autonomous driving. Owing to the powerful reasoning capabilities and extensive pretrained knowledge, large language models (LLMs) have found wide application in autonomous driving tasks, such as vehicle decision-making [7, 10, 15, 27, 32, 39, 43, 54, 59], scene understanding [20, 36, 44, 54], risk analysis [3, 60], and simulation [47, 50]. GPT-Driver [32] represents the entire surrounding environment textually and employs a fine-tuned ChatGPT model to predict vehicle trajectories. DriveGPT4 [54] leverages multimodal LLMs for end-to-end vehicle action prediction and scene understanding. It processes 8-frame video clips and human user commands to predict vehicle control signals and action explanations. However, most of these approaches are designed for open-loop scenarios and present unsatisfactory performance in closed-loop tasks, limiting their applicability to real-world driving deployment. Additionally, many existing methods face efficiency challenges due to heavy model architectures.
Closed-loop end-to-end autonomous driving. Closed-loop end-to-end autonomous driving has recently emerged

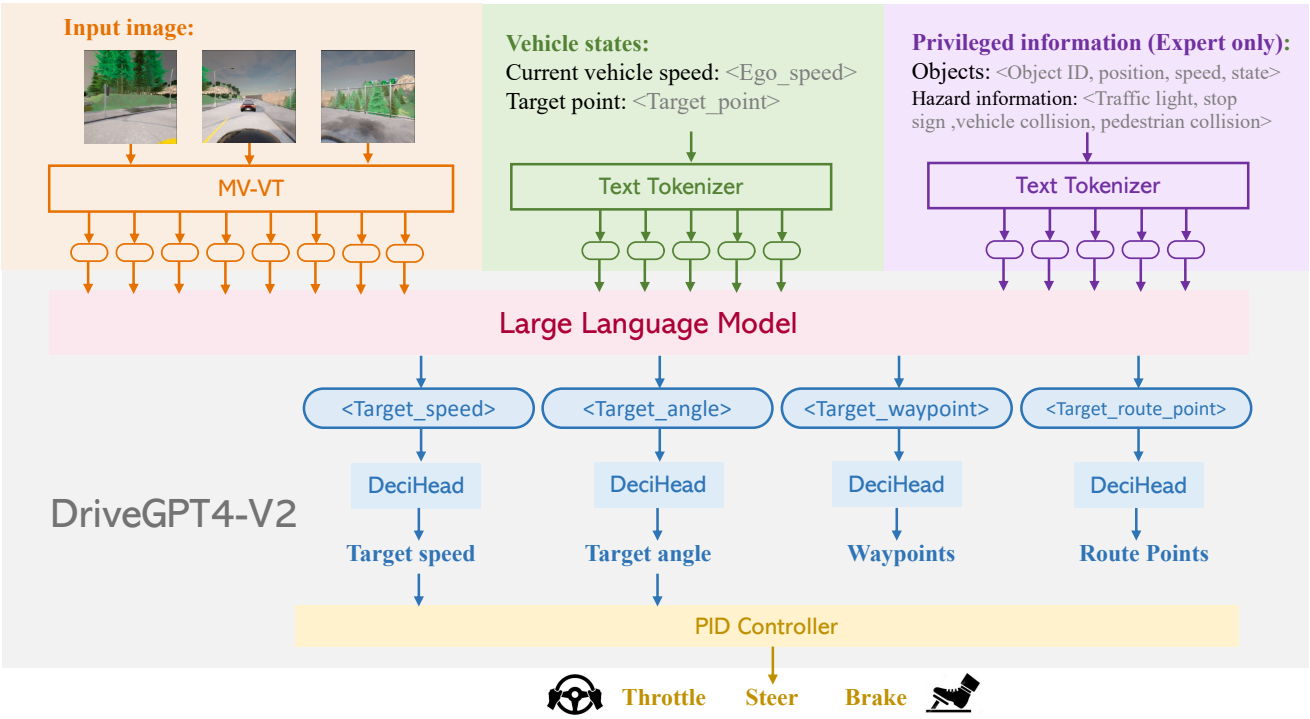


Figure 2. DriveGPT4-V2 diagram. DriveGPT4-V2 takes multimodal input data to generate numerical control signals for end-to-end vehicle driving. The input includes multi-view images and vehicle state information. The images are transformed into the text domain via a multi-view vision tokenizer (MV-VT). The current speed of the vehicle and target point are tokenized by the LLM tokenizer. The LLM then outputs four tokens. Each token is used to predict one vehicle decision by an MLP decision head (DeciHead). These predicted decisions are subsequently converted to low-level commands via PID controllers to operate the vehicle. The LLM expert model, which shares a similar structure to DriveGPT4-V2, has access to privileged information about surroundings (shown in the purple module). The expert provides on-policy supervision to DriveGPT4-V2 to enhance closed-loop performance. This figure is best viewed in color.

as a prominent research focus, addressing the challenge of bridging the gap between academic research and real-world applications. Various approaches improved transformer architectures and training methods to enhance driving performance [4, 5, 12, 16, 21, 22, 42, 58, 63]. For instance, Transfuser++ [21] integrates multimodal sensor inputs, including camera imagery, LiDAR point clouds and vehicle dynamics, to predict speed and trajectory, achieving state-of-the-art performance across multiple benchmarks. Even though most early LLM-based methods have primarily targeted open-loop tasks, recent studies [27, 39, 43, 51] have attempted closed-loop approaches. But most of them are deployed in simple environments like Highway-env [51], instead of real-world scenarios. LMDrive [43], pioneering work on end-to-end closed-loop driving with multimodal LLMs, demonstrated the potential of LLMs in this domain. It is deployed in the CARLA simulator [14] and controls the vehicle like human drivers. However, it relies solely on behavior cloning and presents unsatisfactory driving performance. Online imitation learning has been highlighted as a potential way to enhance model robustness.

Imitation learning in autonomous driving. Imitation learning [19] aims to train an agent to mimic the expert behavior. Behavior cloning is the most widely used imitation learning algorithm in autonomous driving [4, 12, 21, 39, 43, 44, 52–54]. In this method, data is off-policy sampled from expert trajectories for model training to replicate expert actions. Although behavior cloning can yield reasonable performance, it suffers from error accumulation during closed-loop operation, leading the model to encounter unseen situations that may compromise safety. To mitigate this issue, online imitation learning [40, 41] has been employed in various works [4, 5]. The LBC framework [5] uses the DAgger algorithm [41], and trains an agent alongside an expert network with privileged information that achieves superior performance. The expert provides on-policy supervision to the agent and collects additional data samples for data aggregation, which can be used to further train the agent model to handle errors. Inspired by LBC, DriveGPT4-V2 harnesses the capabilities of LLMs and leverages a privileged LLM expert to provide on-policy supervision, facilitating improved driving performance and robustness.

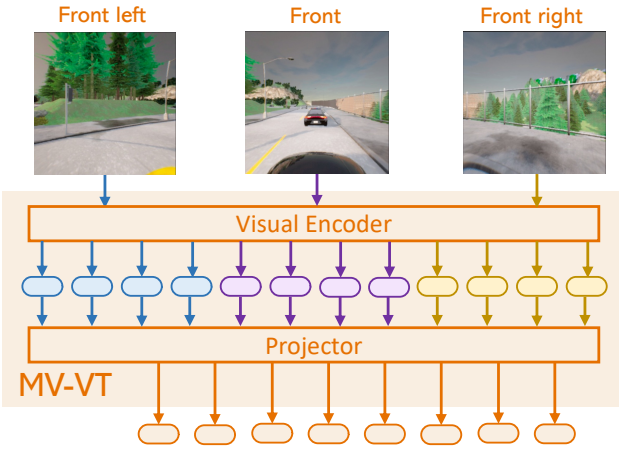


Figure 3. Multi-view visual tokenizer (MV-VT) structure. The input images consist of three front views. Each patch is processed through a visual encoder to extract features. Finally, a trained projection layer maps the downsampled feature into the text domain for further processing.

3. DriveGPT4-V2

Building on DriveGPT4-V1 [54], a flexible multimodal LLM designed for open-loop autonomous driving, DriveGPT4-V2 has been specifically developed for closed-loop driving tasks. This paper employs LLMs both as planners and as expert models for online imitation learning. A multi-view visual tokenizer (MV-VT) is leveraged for effective feature extraction. Unlike standard LLMs used for NLP, the architecture of DriveGPT4-V2 is tailored for vehicle planning, with additional output decision heads (DeciHeads) to directly optimize numerical decision-making predictions. This design enhances model efficiency by reducing the token count required for inference. The expert model, which shares a similar architecture to DriveGPT4-V2, has access to privileged ground-truth information about the surroundings and is used solely for on-policy supervision during training, not for inference.

3.1. Model Design

DriveGPT4-V2 takes multimodal input data and predicts high-level vehicle decisions for end-to-end autonomous driving. These predictions are subsequently converted into low-level control signals through PID controllers. The model structure is shown in Fig. 2.

Multimodal input. The input consists of two modalities: image and text. The image input is captured by three front-view cameras at a resolution of 384×384, while the text input includes the vehicle's current speed and the target point. These three front-view images ensure coverage of both sides of the vehicle, reducing the risk of missing adjacent vehicles or potential collisions. The image data is mapped into the text domain via a visual tokenizer, as implemented in LLaVA [30], while text inputs are tokenized with standard LLM tokenizers.

(a) Waypoints. (b) Route points.

Figure 4. Visualization of waypoints and route points. The ego vehicle is represented by the green rectangle, and the red point denotes the target point. The grey line is the route that the vehicle should follow. (a) Waypoints (blue points) represent vehicle positions in a 4-second future. (b) Route points are evenly sampled from the global planned route in front. These two kinds of points can better supervise the training of DriveGPT4-V2.

Multi-view visual tokenizer (MV-VT). The visual tokenizer is visualized in Fig. 3. It first employs pretrained visual encoders, such as CLIP [37] or SigLIP [57], to extract features from the input images. The size of each image is specifically optimized for compatibility with the SigLIP encoder (i.e., 384×384). The three images are acquired from front-left, front, and front-right cameras respectively, ensuring comprehensive coverage of the vehicle's frontal and lateral views. In CARLA benchmarks, back-view images exhibit limited utilization frequency and demonstrate no measurable improvement in planning performance. To maintain computational efficiency, MV-VT deliberately excludes back-view inputs.

Output representations. DriveGPT4-V2 generates three low-level control signals to drive the vehicle, i.e., throttle, brake, and steering. However, direct prediction of these signals requires temporal consistency; otherwise, the vehicle may display unstable zigzag trajectories. To ensure stable control, DriveGPT4-V2 instead predicts high-level decision variables, which are then converted to low-level commands by PID controllers. The predictions of the model include:

- Target Speed: The desired speed for the vehicle. A speed PID controller adjusts longitudinal control based on the predicted target speed and current speed.
- Target Angle: The direction in which the vehicle should steer. A steering PID controller uses it for lateral control.
- Waypoints: Predicted x-y positions of the vehicle in the future. Waypoints consist of eight points.
- Route Points: Points evenly sampled along the global route in front, enhancing the vehicle's perception ability.

Waypoints and route points are visualized in Fig. 4.

Decision heads (DeciHeads). For closed-loop autonomous driving, DriveGPT4-V2 employs dedicated output heads (MLPs) to predict numerical vehicle decisions. The original LLM's vocabulary output head is replaced with four decision heads, each predicting a decision variable (i.e., target speed, target angle, waypoints, and route points) based on one token. This setup allows DriveGPT4-V2 to produce only four tokens per prediction cycle, achieving a huge

speed increase over traditional text-based outputs. For example, a float number with 3 digits (e.g., 1.538) needs 5 tokens for text output representation but only requires a single token for our design. DeciHead also enables direct optimization of decision variables using regression loss.

Expert model and online imitation learning. The expert model has a similar architecture to DriveGPT4-V2 but has access to privileged information. The privileged information consists of ground truth (GT) of nearby objects and hazard information provided by the CARLA simulator (e.g., traffic light and stop sign violation; vehicle or pedestrian collision). This privileged data allows the expert to better perceive the environment, so that it can focus on vehicle decisions and planning. The privileged information greatly boosts the model capacity, resulting in enhanced closed-loop performance. The expert model provides on-policy supervision to DriveGPT4-V2 during training.

When DriveGPT4-V2 runs on training scenarios and routes, the expert model receives identical input and generates concurrent predictions. If the discrepancy between DriveGPT4-V2's prediction and the expert's exceeds a threshold, the expert's prediction is used instead to control the vehicle. Such a situation is marked as an exception case. A data sample of the current moment is added to the training set, where the expert's output is used as the label. DriveGPT4-V2 is fine-tuned on this aggregated dataset under the expert's on-policy supervision for enhanced closed-loop autonomous driving performance.

The expert LLM is used exclusively for on-policy supervision during training and is not accessed during inference.

3.2. Training

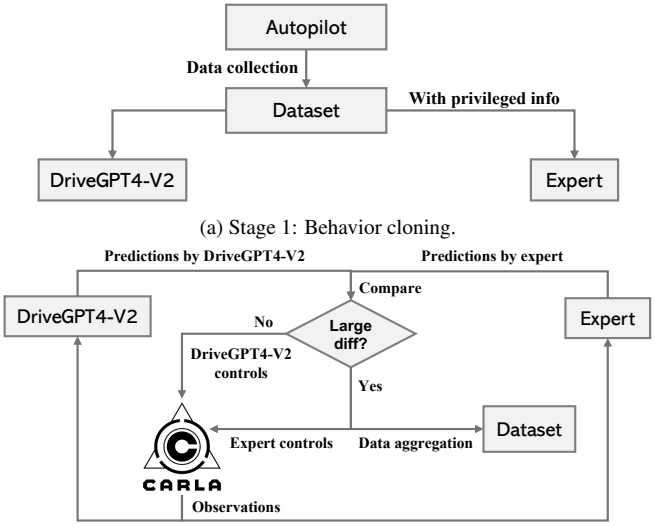
The training process of DriveGPT4-V2 and the expert LLM includes two stages. In the first stage, both models are trained on data collected by a rule-based autopilot using behavior cloning. In the second stage, DriveGPT4-V2 runs on the training scenarios and routes under the expert's on-policy supervision, and generates additional data for error correction. The training process is visualized in Fig. 5.

3.2.1. Stage 1: behavior cloning

In the first stage, a rule-based autopilot [21] is deployed in CARLA to collect data. The autopilot agent records data at fixed time intervals, including camera images, current vehicle speed, privileged information on surrounding objects (e.g., vehicles and traffic lights), and a target point. Labels for the vehicle decision variables are calculated using rule-based algorithms. After data collection, both DriveGPT4-V2 and the expert LLM are trained on the dataset, with the expert having access to privileged information.

3.2.2. Stage 2: DAgger with on-policy supervision

The expert, with access to privileged information, achieves notably higher performance by leveraging reliable priors on surrounding objects, allowing it to handle challenging scenarios effectively. DriveGPT4-V2, trained only by behavior cloning, may struggle with error accumulation and exceptional situations (e.g., vehicle collisions or getting stuck). Inspired by LBC [5], the expert LLM is used as a teacher for on-policy supervision. On-policy supervision is a training paradigm in imitation learning, where supervision or feedback is provided to an agent while it is acting according to its current policy. This approach contrasts with off-policy methods, where supervision comes from data collected by a different policy or behavior. During the operation of DriveGPT4-V2, the expert and DriveGPT4-V2 receive identical input, and their outputs are compared. If the discrepancy in predicted decisions exceeds a threshold, the expert takes control to avoid exceptions. One data sample is collected at that moment and aggregated into the dataset. The output of the expert is used as the training label. After traversing the training routes under expert supervision, DriveGPT4-V2 is fine-tuned on the aggregated dataset.



(a) Stage 1: Behavior cloning.

(b) Stage 2: DAgger online imitation learning.

Figure 5. Diagram of the two-stage training process. (a) In the first stage, both DriveGPT4-V2 and the expert LLM are trained on data collected by a rule-based autopilot. (b) In the second stage, DriveGPT4-V2 runs on the training scenarios and routes. When the discrepancy between DriveGPT4-V2's predictions and those of the expert exceeds a predefined threshold, the expert's predictions are used to control the vehicle. Data from these cases is then added to the dataset for data aggregation.

3.3. Dataset

Following Transfuser++ [21], in stage 1, the rule-based autopilot agent is used to collect data samples in various CARLA simulator towns under varying weather, lighting, and daytime conditions. Vehicle speed and position are estimated by Kalman filters based on IMU data. Each sam-

ple records objects within a fixed radius, including vehicles, pedestrians, traffic lights, and stop signs. Target speeds and angles are calculated by the autopilot's PID controller. The ground-truth waypoints contain 8 points and record the vehicle position in a 4-second future. Route points have 10 points sampled along the planned route in front.

In stage 2, DriveGPT4-V2 runs on selected training routes for data aggregation. Simple routes are not considered in this stage for data collection efficiency. If the difference between the predicted decisions of DriveGPT4-V2 and the expert exceeds a threshold, the expert takes control and records the data sample. DriveGPT4-V2 is retrained on the aggregated dataset for better performance. Under common circumstances, DriveGPT4-V2 achieves satisfactory performance after one round of DAgger.

3.4. Loss Function

During training, the L1 loss function supervises all predicted decision variables, including target speed (TS), target angle (TA), waypoints (WP), and route points (RP). The overall loss is calculated by:

$$\mathcal{L} = \mathcal{L}_{TS} + \mathcal{L}_{TA} + \mathcal{L}_{WP} + \mathcal{L}_{RP} \quad (1)$$

This approach allows the model to optimize final numerical vehicle decision variables directly, instead of relying on vocabulary-based classification as in NLP tasks.

4. Experiments

4.1. Dataset and Benchmark

All experiments in this study were conducted in the CARLA simulator [14]. We followed Transfuser++ for training data preparation. In stage 1, a rule-based autopilot was employed to collect 350K behavior-cloning samples. The autopilot traversed multiple routes in Towns 1, 2, 3, 4, 5, 6, 7, and 10 of CARLA, encountering various scenarios such as different weather and lighting conditions. Data were collected at a frequency of 2 Hz and subsequently filtered to remove incorrect or redundant samples, such as those captured when the autopilot was stuck. This process resulted in a dataset of 300K samples.

In stage 2, DriveGPT4-V2 trained by behavior cloning runs on selected training scenarios and routes, utilizing the expert for on-policy supervision. The expert can access privileged information about objects within a 30-meter distance. Approximately 150K samples were gathered in this stage. These aggregated data were then combined with the original dataset to further fine-tune DriveGPT4-V2, enhancing its performance.

All methods were evaluated on the challenging CARLA Longest6 benchmark [12], which consists of 36 extended routes encompassing various scenarios. The length and complexity of routes make them suitable for a comprehensive evaluation of closed-loop autonomous driving.

Table 1. Infraction penalty factors.

Scene	Penalty	Scene	Penalty
Pedestrian collision	0.5	Vehicle collision	0.6
Static object collision	0.65	Traffic light violation	0.7
Stop sign violation	0.7		

4.2. DriveGPT4-V2 Configuration

LLM. For efficient data collection and inference, we used tiny-scale LLMs in DriveGPT4-V2, such as Qwen-0.5B [2] and TinyLLaMA [61]. These models were fine-tuned for multimodal understanding, following methods outlined in LLaVA [29, 30] and TinyLLaVA [64]. Large-scale LLMs, like LLaMA3-8B [49] and vicuna-7B [11], are not considered at the moment due to their high computational demands and reduced inference speed.

Training. DriveGPT4-V2 was fully fine-tuned in an end-to-end manner, except for the SigLIP 384×384 visual encoder [57]. The model was trained with a learning rate of $2e^{-5}$ over 60 epochs by employing a cosine annealing schedule for better learning rate control. The training takes about 75 hours on 16 A800 GPUs.

Inference. Two PID controllers were designed to convert DriveGPT4-V2's predicted decision variables into low-level control signals (i.e., throttle, steer, and brake). These control signals were used to directly drive the ego vehicle in the CARLA simulator. In the experiments, only the predicted target speed and target angle were used by the PID controllers, while waypoints and route points served as additional supervision for improved training results. The model runs inference on a single A800 GPU with FP16 precision.

4.3. Evaluation Criteria

Following the CARLA leaderboard, the methods were evaluated mainly based on three metrics: Driving Score (DS), Route Completion (RC), and Infraction Score (IS). RC indicates the percentage of routes completed, while IS penalizes infractions such as traffic violations and collisions during inference. Each infraction multiplicatively reduces IS by a penalty factor. The penalty factors are listed in Tab. 1. DS, as the product of RC and IS, provides a comprehensive evaluation score. In the evaluation, we also report ratios of commonly seen exception events, including pedestrian collision (Ped), vehicle collision (Veh), static object collision (Stat), red light violation (Red), route deviations (Dev), scenario time out (TO) and agent block (Block).

4.4. Comparison Experiments

Baselines. We selected several representative prior works as baselines for comparison:

- WOR [6] (ICCV 2021): WOR uses commands instead of target points for global planning instructions, providing

Table 2. Closed-loop experiments performance on CARLA Longest6. "Visual" indicates visual input modalities, while "C" and "L" represent camera and LiDAR, respectively. **Bold numbers** highlight SOTA metric scores of all models; while underlined numbers represent the best metric scores of baseline methods. * denotes models implemented by ourselves based on official open-sourced code. † represents the model without data augmentation.

Method	Visual	DS ↑	RC ↑	IS ↑	Ped ↓	Veh ↓	Stat ↓	Red ↓	Dev ↓	TO ↓	Block ↓
WOR [6]	C	21	48	0.56	0.18	1.05	0.37	1.28	0.88	0.08	0.20
LAV v1 [4]	C&L	33	70	0.51	0.16	0.83	0.15	0.96	0.06	0.12	0.45
Interfuser [42]	C	47	74	0.63	0.06	1.14	0.11	0.24	**0.00**	0.52	**0.06**
TransFuser [12]	C&L	47	**93**	0.50	0.03	2.45	0.07	0.16	**0.00**	**0.06**	0.10
LAV v2 [4]	C&L	58	83	0.68	**0.00**	0.69	0.15	0.23	0.08	0.32	0.11
Perception PlanT [38]	C&L	58	88	0.65	0.07	0.97	0.11	0.09	**0.00**	0.13	0.13
Transfuser++* [21]	C&L	65	90	0.72	**0.00**	0.99	**0.01**	0.07	**0.00**	0.10	0.12
Transfuser++*† [21]	C&L	58	89	0.65	0.01	1.15	**0.01**	0.10	**0.00**	0.14	0.13
LMDrive* [43]	C&L	36	69	0.52	0.07	1.03	0.18	1.01	0.09	0.11	0.22
DriveGPT4-V2	C	**70**	91	**0.77**	**0.00**	**0.80**	**0.01**	**0.04**	**0.00**	0.07	0.09

enriched labels and supervision for possible actions.

- LAV v1 & v2 [4] (CVPR 2022): LAV predicts waypoints by integrating information from nearby vehicles, which is further refined through a GRU.
- Transfuser [12] (TPAMI 2022): Transfuser is a widely used baseline model for closed-loop driving that has inspired subsequent works.
- Perception PlanT [38] (CoRL 2022): PlanT is based on imitation learning with object-level input representation. Such representation is elegant and effective.
- InterFuser [42] (CoRL 2023): InterFuser predicts additional density maps and traffic rule flags to enhance waypoint prediction.
- Transfuser++ [21] (ICCV 2023): Transfuser++ predicts target speeds and routes instead of waypoints, integrating point cloud and camera images. It is currently the state-of-the-art (SOTA) method for CARLA Longest6. Transfuser++ utilizes triple training data by repeatedly collecting expert trajectories for augmentation.
- LMDrive [43] (CVPR 2024): LMDrive is a pioneering work for closed-loop driving with multimodal LLMs, training a vicuna-7B LLM for end-to-end vehicle control.

Comparison results. The comparison results on the CARLA Longest6 benchmark are shown in Tab. 2. It is observed that DriveGPT4-V2 outperforms all baseline models by a large margin. Thanks to the huge amount pretrained knowledge of LLMs and visual encoders, DriveGPT4-V2 can better handle complex urban scenes under various conditions, such as nighttime, rainy weather, etc. For most baselines, we list the results reported on the CARLA Longest6 leaderboard. However, some baselines cannot achieve reported performance by our implementation and test, thus we report our implemented results separately. LMDrive [43] is the only open-sourced LLM method that can be tested in a closed-loop environment. But it fails to obtain satisfactory performance. It first converts multi-modal data to BEV and projects BEV to the text domain. However, BEV does not have good-quality pretrained multimodal LLMs. Training from scratch severely degrades its final performance, which is also discussed by DriveGPT4-V1 [54]. Transfuser++ [21] is the SOTA method by fusing point clouds and camera images for closed-loop control. Transfuser++ uses triple training data and achieves 65 DS, but only has 58 DS with the same amount of training data (denoted by †) as our method. Thanks to the pretrained knowledge and reasoning ability of LLMs, DriveGPT4-V2 can outperform all baselines with camera images as input.

Table 3. Effciency analysis.

LLM	DS	Train	FPS
LLaVA-LLaMA3.1-8B	65	11.2h/epoch	0.4
TinyLLaVA-LLaMA-1.5B	63	3.0h/epoch	2.9
LLaVA-Qwen-0.5B	63	1.3h/epoch	8.1

4.5. Model Efficiency

In our experiments, DriveGPT4-V2 and the expert are implemented by tiny LLMs. LLMs with 7B or even more parameters show powerful reasoning and generalization ability, but they severely enlarge the time consumption and demand much more training resources, which is unacceptable in real-world applications. In this section, we report efficiency results of DriveGPT4-V2 with LLaVA-0.5B, TinyLLaVA-1.5B and LLaVA-7B LLMs. The results are shown in Tab. 3. It is found that merely scaling up the LLM model does not gain corresponding performance enhancement, but slows down the whole system more than 10 times. Therefore, DriveGPT4-V2 mainly considers 0.5B LLMs as the planner.

4.6. Ablation Studies and Discussion

We conducted multiple ablation studies to justify the design of DriveGPT4-V2. The results are presented in Tab. 4, 5 and 6. Due to huge time consumption, DriveGPT4-V2 in ablation studies is not trained with DAgger data by default.

LLM visual pretraining. DriveGPT4-V2 relies on LLaVA [30, 64] as the planner, which is pretrained on a huge amount of visual understanding data. The visual pretraining benefits DriveGPT4-V2 on the autonomous driving task with multimodal input. Directly training DriveGPT4-V2 on scratch (i.e., LLM for NLP tasks without multimodal ability) slows down convergence and degrades the final results.

Visual tokenizer. The input camera images are processed by the proposed visual tokenizer. Inspired by LLaVA 1.5 [30], the features of images are extracted separately by the visual encoder and then concatenated as the input. This design ensures a sufficient perception range without compromising image detail. If the input is a single image with a large scope, it might lead to information loss during preprocessing steps (e.g., resize or crop); if the input is a small square image, the vehicle cannot have a sufficient perception range for effective planning.

Waypoints and route points. Although DriveGPT4-V2 does not rely on waypoints and route points for generating low-level control signals, they provide valuable supervision during training. Predicting waypoints and route points helps the model perceive the target route and anticipate future actions. Removing them negatively impacts performance.

Expert on-policy supervision. The expert model, benefiting from privileged information, provides robust on-policy supervision to DriveGPT4-V2. Without expert guidance, DriveGPT4-V2 relies solely on behavior cloning instead of DAgger imitation learning, which can lead to error accumulation and degraded performance.

PID controllers. Previous works often generated low-level control signals based on waypoints or route points [12, 21] by calculating target speed and angle from these predictions. However, even minor prediction errors in waypoints or route points can lead to substantial errors in speed and angle. As shown in Tab. 5, using waypoints or route points for PID control can make control signals noisy and unstable, potentially driving the vehicle into exceptional states. Therefore it presents inferior results.

Decision heads. During inference, DriveGPT4-V2 generates four tokens in an auto-regressive manner, mapped to vehicle decision variables by four decision heads. Predictions of target speed and angle involve one-to-one mappings (i.e., one token for one number), while waypoint and route point predictions involve one-to-many mappings (e.g., waypoints contain 8 x-y points, thus each token is mapped to 16 numbers). Another design would involve predicting each waypoint and route point one at a time, requiring 8 tokens for waypoints and 10 tokens for route points. Such design does not significantly improve performance and severely affects efficiency. Using text to represent output numbers would require approximately 160 tokens, making it impractical for real-time autonomous driving. The evaluation results are shown in Tab. 6

Table 4. Ablation studies of DriveGPT4-V2. "WP" and "RP" represent waypoints and route points, respectively.

	DS	RC	IS
Baseline	47	78	0.60
+ LLM Visual Pretraining	56	87	0.64
+ Visual Tokenizer	60	88	0.68
+ WP&RP	63	90	0.70
+ Expert Supervision	**70**	**91**	**0.77**

Table 5. Ablation studies on PID controllers. "WP" indicates utilizing predicted waypoints for PID control; while "TS&RP" means PID control by predicted target speed and route points.

PID Controller	DS	RC	IS
WP	53	85	0.62
TS & RP	59	88	0.67
DriveGPT4-V2	63	90	0.70

Table 6. Ablation studies on decision heads. "Additional tokens" indicates using more output tokens for prediction.

	DS	RC	IS	FPS
Additional tokens	64	91	0.70	1.4
DriveGPT4-V2	63	90	0.70	8.1

5. Conclusion and Future Work

In this paper, we presented DriveGPT4-V2, a novel LLM-based framework for closed-loop, end-to-end autonomous driving. DriveGPT4-V2 processes camera images and vehicle states as input to generate low-level control signals for direct vehicle operation. Leveraging the extensive pretrained knowledge of MLLMs, DriveGPT4-V2 demonstrates the ability to navigate complex urban scenarios under diverse and challenging conditions. The model architecture is specifically optimized for precise numerical vehicle decision prediction. An additional expert LLM, which shares a structure similar to DriveGPT4-V2, has been trained to provide on-policy supervision. Experimental results highlight that DriveGPT4-V2 achieves state-of-the-art performance on the challenging CARLA Longest6 benchmark, outperforming all baselines. In the future, we aim to extend the capabilities of DriveGPT4-V2 for broader applications in closed-loop autonomous driving tasks, including video games and real-world deployment.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (No. 62422606, 62201484, 52221005).

References

[1] Anthropic. Claude. https://www.anthropic.com, 2024.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv:2309.16609*, 2023.

[3] Xu Cao, Tong Zhou, Yunsheng Ma, Wenqian Ye, Can Cui, Kun Tang, Zhipeng Cao, Kaizhao Liang, Ziran Wang, James M Rehg, et al. Maplm: A real-world large-scale vision-language benchmark for map and traffic scene understanding. In *CVPR*, 2024.

[4] Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *CVPR*, 2022.

[5] Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *CoRL*, 2020.

[6] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *ICCV*, 2021.

[7] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. In *ICRA*, 2024.

[8] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *TPAMI*, 2024.

[9] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv:2402.13243*, 2024.

[10] Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. Asynchronous large language model enhanced planner for autonomous driving. In *ECCV*, 2024.

[11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org*, 2023.

[12] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 2022.

[13] Google DeepMind. Gemini ai. https://deepmind.com, 2024.

[14] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, 2017.

[15] Yuchao Feng, Zhen Feng, Wei Hua, and Yuxiang Sun. Multimodal-xad: Explainable autonomous driving based on multimodal environment descriptions. *TITS*, 2024.

[16] Jiawei Fu, Yanqing Shen, Zhiqiang Jian, Shitao Chen, Jingmin Xin, and Nanning Zheng. Interactionnet: Joint planning and prediction for autonomous driving with transformers. In *IROS*, 2023.

[17] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023.

[18] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, 2024.

[19] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 2017.

[20] Yuichi Inoue, Yuki Yada, Kotaro Tanahashi, and Yu Yamaguchi. Nuscenes-mqa: Integrated evaluation of captions and qa for autonomous driving datasets using markup annotations. In *WACV*, 2024.

[21] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden biases of end-to-end driving models. In *ICCV*, 2023.

[22] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *CVPR*, 2023.

[23] 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 *ICCV*, 2023.

[24] Zhuoling Li, Liangliang Ren, Jinrong Yang, Yong Zhao, Xiaoyang Wu, Zhenhua Xu, Xiang Bai, and Hengshuang Zhao. Virt: Vision instructed transformer for robotic manipulation. *arXiv:2410.07169*, 2024.

[25] Zhuoling Li, Xiaogang Xu, Zhenhua Xu, SerNam Lim, and Hengshuang Zhao. Larm: Large auto-regressive model for long-horizon embodied intelligence. *arXiv:2405.17424*, 2024.

[26] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv:2311.10122*, 2023.

[27] Yuanfei Lin, Chenran Li, Mingyu Ding, Masayoshi Tomizuka, Wei Zhan, and Matthias Althoff. Drplanner: Diagnosis and repair of motion planners for automated vehicles using large language models. *RA-L*, 2024.

[28] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *arXiv:2310.03744*, 2023.

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[30] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next, 2024.

[31] Tianyu Liu, Qing hai Liao, Lu Gan, Fulong Ma, Jie Cheng, Xupeng Xie, Zhe Wang, Yingbing Chen, Yilong Zhu, Shuyang Zhang, et al. The role of the hercules autonomous vehicle during the covid-19 pandemic: An autonomous logistic vehicle for contactless goods transportation. *IEEE Robotics & Automation Magazine*, 2021.

[32] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. In *NeurIPS Workshop*, 2023.

[33] Maxim Mnyakin. Challenges and opportunities of integrating autonomous vehicles into urban retail delivery services. *Reviews of Contemporary Business Analytics*, 2023.

[34] OpenAI. Chatgpt. https://chat.openai.com, 2024.

[35] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 2018.

[36] SungYeon Park, MinJae Lee, JiHyuk Kang, Hahyeon Choi, Yoonah Park, Juhwan Cho, Adam Lee, and DongKyu Kim. Vlaad: Vision and language assistant for autonomous driving. In *WACV*, 2024.

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[38] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. In *CoRL*, 2022.

[39] 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:2406.10165*, 2024.

[40] Stephane Ross and J. Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. In *arXiv:1406.5979*, 2014.

[41] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

[42] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *CoRL*, 2023.

[43] 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 *CVPR*, 2024.

[44] 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. In *ECCV*, 2024.

[45] Robert Soler, Alae Moudni, Gabriel Roskowski, Xinrui Yu, Mikhail Gormov, and Jafar Saniie. Autonomous patrol and threat detection through integrated mapping and computer vision. In *IEEE International Conference on Electro Information Technology*, 2024.

[46] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *ICCV*, 2023.

[47] Yafu Tian, Alexander Carballo, Ruifeng Li, Simon Thompson, and Kazuya Takeda. Rsg-search plus: An advanced traffic scene retrieval methods based on road scene graph. In *IV*, 2024.

[48] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *IJCAI*, 2018.

[49] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv:2302.13971*, 2023.

[50] Yuxi Wei, Zi Wang, Yifan Lu, Chenxin Xu, Changxing Liu, Hao Zhao, Siheng Chen, and Yanfeng Wang. Editable scene simulation for autonomous driving via collaborative llm-agents. In *CVPR*, 2024.

[51] 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. In *ICLR*, 2024.

[52] Zhenhua Xu, Yuxuan Liu, Lu Gan, Yuxiang Sun, Ming Liu, and Lujia Wang. Rngdet: Road network graph detection by transformer in aerial images. *TGRS*, 2022.

[53] Zhenhua Xu, Yuxuan Liu, Yuxiang Sun, Ming Liu, and Lujia Wang. Rngdet++: Road network graph detection by transformer with instance segmentation and multi-scale features enhancement. *RA-L*, 2023.

[54] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *RA-L*, 2024.

[55] Yijun Yang, Tianyi Zhou, Kanxue Li, Dapeng Tao, Lusong Li, Li Shen, Xiaodong He, Jing Jiang, and Yuhui Shi. Embodied multi-modal agent trained by an llm from a parallel textworld. In *CVPR*, 2024.

[56] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*.

[57] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

[58] Jimuyang Zhang, Zanming Huang, and Eshed Ohn-Bar. Coaching a teachable student. In *CVPR*, 2023.

[59] Jimuyang Zhang, Zanming Huang, Arijit Ray, and Eshed Ohn-Bar. Feedback-guided autonomous driving. In *CVPR*, 2024.

[60] Jiawei Zhang, Chejian Xu, and Bo Li. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles. In *CVPR*, 2024.

[61] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv:2401.02385*, 2024.

[62] Xiang Zhang, Haojie Sun, Xiaoyang Pei, Linghui Guan, and Zihao Wang. Evolution of technology investment and development of robotaxi services. *Transportation Research Part E: Logistics and Transportation Review*, 2024.

[63] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *ICCV*, 2021.

[64] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv:2402.14289*, 2024.

[65] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024.