Edge-Cloud Collaborative Motion Planning for Autonomous Driving with Large Language Models

Jiao Chen*, Suyan Dai* Fangfang Chen*, Zuohong Lv* and Jianhua Tang*†
* Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, China
† Pazhou Lab, Guangzhou, China
{202110190459, 202066200091, wifannychen59, 202220159664}@mail.scut.edu.cn, jtang4@e.ntu.edu.sg

***Abstract*—Integrating large language models (LLMs) into autonomous driving enhances personalization and adaptability in open-world scenarios. However, traditional edge computing models still face significant challenges in processing complex driving data, particularly regarding real-time performance and system efficiency. To address these challenges, this study introduces EC-Drive, a novel edge-cloud collaborative autonomous driving system with data drift detection capabilities. EC-Drive utilizes drift detection algorithms to selectively upload critical data, including new obstacles and traffic pattern changes, to the cloud for processing by GPT-4, while routine data is efficiently managed by smaller LLMs on edge devices. This approach not only reduces inference latency but also improves system efficiency by optimizing communication resource use. Experimental validation confirms the system's robust processing capabilities and practical applicability in real-world driving conditions, demonstrating the effectiveness of this edge-cloud collaboration framework. Our data and system demonstration will be released at https://sites.google.com/view/ec-drive.**

***Index Terms*—Edge-cloud Collaboration, Autonomous Driving, Motion Planning, Large Language Models, LLaMA, GPT-4**

I. Introduction

As intelligent transportation and autonomous driving technologies rapidly advance, the motion planning system, as a critical component, faces increasingly complex environments and diverse challenges. Traditional motion planning methods often rely on fixed algorithms and models, making it difficult to fully address the dynamic changes in traffic conditions and the personalized needs of drivers [1].

Integrating large language models (LLMs) into autonomous vehicles not only enables artificial intelligence systems to control the driving process but also significantly enhances the system's personalization and adaptability. By understanding natural language commands, LLMs can dynamically adjust driving strategies to meet the personalized preferences of drivers or passengers, thereby improving the overall driving experience. Moreover, the integration of LLMs allows autonomous systems to better handle complex and dynamic open-world scenarios, making them more flexible in addressing diverse driving tasks.

The Transformer, originally designed for sequential data, has achieved state-of-the-art performance in natural language processing, driving the development of LLMs [2], [3]. These models pretrain Transformer architectures (encoder, encoder-decoder, and decoder) on vast corpora to capture extensive language statistics. Pretrained LLMs can be fine-tuned for specialized downstream tasks. The Vision Transformer (ViT) [4] applies the Transformer to image tasks, converting images into sequences of patches that the Transformer can process. CLIP [5], a multimodal model that matches textual descriptions with images, demonstrates strong transfer capabilities in many image classification tasks. Utilizing pretrained LLMs as a framework for multimodal tasks leverages their text generation capabilities, which is crucial for the question-answering tasks in our research. However, despite their impressive performance in many tasks, deploying these large models with typically over a billion parameters for real-time applications remains challenging.

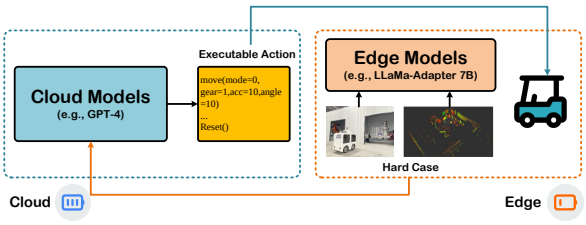

Fig. 1: Architecture of the EC-Drive system. LLM-based motion planning is performed on edge devices within the vehicle, while complex inference tasks are offloaded to the cloud, which has larger models and more extensive resources.

Although autonomous driving systems primarily rely on visual features, incorporating linguistic features can enhance system interpretability and aid in identifying new traffic situations. This advantage has sparked interest in integrating multimodal data to train language models as autonomous driving agents. DriveGPT4 [6] employs LLaMA as the backbone LLM, with CLIP as the visual encoder, using traffic scene videos and prompt texts as inputs to generate responses and low-level vehicle control signals. DriveMLM [7] utilizes multi-view images, LiDAR point clouds, traffic rules, and user instructions from a real simulator to perform closed-loop driving. This multimodal model is constructed with LLaMA and ViT as the image processor. GPT-Driver [8] reframes motion planning as a language modeling task, using GPT-3.5 to represent the planner's inputs and outputs as language tokens.

The corresponding author is Jianhua Tang.
Pre-print, manuscript submitted to IEEE.

However, these models utilize LLMs with over a billion parameters (such as GPT-3.5 [2] and LLaMA [3]) and expensive image encoders (such as CLIP and ViT), making them suitable mainly for latency-insensitive offline scenarios rather than latency-critical online scenarios. Recently, collaboration between large and small language models has garnered significant attention [9]. Inspired by dual-process cognitive theory, various methods can be integrated into a unified framework.

Our primary insight is to use data drift detection algorithms to upload a small number of difficult samples (e.g., new obstacles, changes in traffic patterns) to the cloud for processing by larger-scale models (e.g., GPT-4), while most samples are handled by smaller parameter LLMs at the edge. This approach, illustrated in Fig. 1, ensures low inference latency while improving the handling of dynamic environments. This method has potential applications in remote assistance for autonomous vehicles, enabling them to navigate complex and evolving scenarios more effectively. Our main contributions are as follows:

- We propose a novel edge-cloud collaborative autonomous driving system, EC-Drive, equipped with data drift detection capabilities. This efficient framework utilizes data drift detection algorithms to selectively upload a small number of challenging samples (e.g., new obstacles, changes in traffic patterns) to the cloud for processing by GPT-4, while most of the data is managed by smaller parameter LLMs on edge devices. This approach ensures low inference latency while effectively addressing the challenges of complex environments.
- We introduce a multimodal approach that integrates linguistic features with traditional visual data, enhancing the interpretability and decision-making capabilities of autonomous driving systems. This integration allows the system to better understand and respond to new traffic situations, improving adaptability and safety.
- Detailed experimental validation demonstrates the system's robust processing capabilities and its potential applicability in real-world driving scenarios, highlighting the practical advantages and feasibility of the proposed edge-cloud collaborative framework.

II. Related Works

This section reviews motion planning methods and the practical application of LLMs in autonomous driving, focusing on their strengths and challenges in complex traffic environments.

A. Motion Planning in Autonomous Driving

Autonomous driving utilizes various motion planning strategies for efficient vehicle navigation. (1) Rule-based method: This approach generates paths based on predefined rules that account for environmental constraints like road geometry and traffic signals [10]. While simple and efficient, it is rigid and struggles to adapt to unexpected changes. (2) Optimization-based method: Optimization algorithms compute optimal trajectories by minimizing a cost function considering factors such as time, energy, safety, and comfort [1]. Though precise, these methods are computationally intensive and may not suit real-time decision-making. (3) Learning-based method: This approach uses machine learning to adapt to dynamic environments by learning from past data [11]. Deep neural networks and reinforcement learning provide adaptability but require significant data and resources, often struggling with rare or novel scenarios.

B. Large Models

Large models (LMs) based on the Transformer, such as Large Language Models [2], [3], vision models [4], [5], time series models [12], [13], and multimodal models [14], have gained widespread attention due to their unique advantages. With billions to trillions of parameters, these models accumulate extensive knowledge through pre-trained on large datasets, significantly advancing the automation and diversification of data processing while reducing reliance on human expertise. Such capabilities have attracted broad interest in the industrial sector, fostering numerous studies targeting industrial intelligence.

The collaboration between large and small language models garners considerable attention. Inspired by dual-process cognitive theory, various methods can be integrated into a unified framework. Research indicates that the essential difference between large and small models lies in the control of uncertainty in next token predictions during the decoding process, and it highlights that collaborative interactions between models are most critical at the beginning of the generation process [9].

C. Motion Planning with LLMs.

In recent years, significant progress has been made in the application of LLMs in the field of autonomous driving. Utilizing LLMs to enhance decision-making processes in autonomous vehicles has the potential to transform their operational methods. This approach offers personalized assistance, facilitates continuous learning, and improves decision intelligence [15]. PlanAgent [16] is a multimodal large language model-based autonomous motion planning agent system that enhances environmental understanding through Bird's Eye View (BEV) and lane-graph-based textual descriptions. It introduces a hierarchical Chain of Thought (CoT) [17] to guide the MLLM in generating planner code. Hu *et al.* [18] propose an LLM-driven collaborative driving framework for multiple vehicles, featuring lifelong learning capabilities. It allows different driving agents to communicate with each other, facilitating collaborative driving in complex traffic scenarios. DiLu [19] is the first framework to leverage knowledge-driven capabilities in autonomous driving decision-making. It combines reasoning and reflection modules, enhancing the capabilities of LLMs, enabling them to apply knowledge and perform causal reasoning in the autonomous driving domain. TrafficGPT [20] reveals the application potential of large language models in the smart transportation domain. These models possess the capability to view and process traffic data, providing profound decision support for urban traffic system management. Additionally, they assist in human decision-making during traffic control, demonstrating their practicality and efficacy in traffic management.

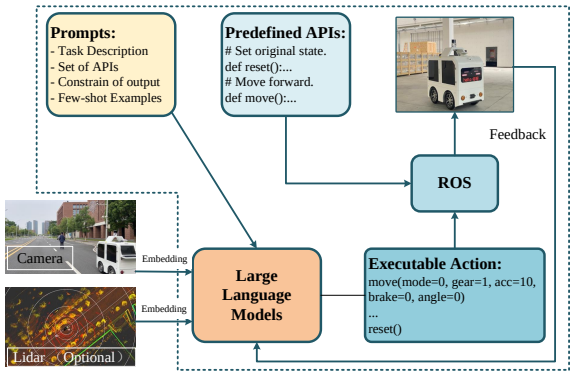

Fig. 2: Motion planning process on the edge through large language models, utilizing vision and LiDAR data for real-time decision-making and execution. ROS stands for Robot Operating System, which is used to execute actions and provide feedback on the execution results.

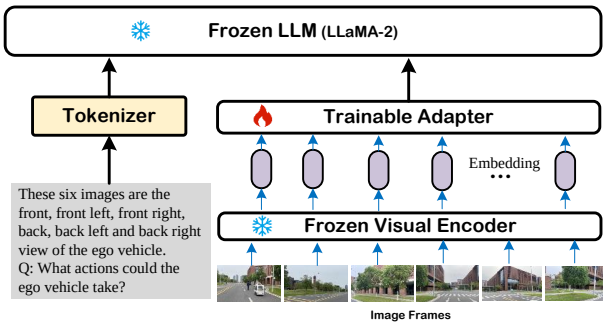

Fig. 3: Instruction tuning of pretrained LLaMA2 models for autonomous driving, using multi-view images and prompt for efficient adaptation to specific driving scenarios.

LimSim++ [21] is an open-source evaluation platform specifically designed for the research of autonomous driving with LVLMs, supporting scenario understanding, decision-making, and evaluation. DriveLM [22] introduces datasets using nuScenes and CARLA, presenting a vision-language models based baseline approach that concurrently addresses Graph visual question answering and end-to-end driving. The experiments showcased Graph visual question answering as a simple and principled framework for scene reasoning. CODA-LM [23] demonstrates that even the most advanced autonomous driving perception systems struggle with handling complex road corner cases.

III. Edge-Cloud Collaborative Motion Planning for Autonomous Driving

In this section, we elaborate on the methodologies and technologies employed in the EC-Drive system, emphasizing the use of edge and cloud models as well as the collaborative process between them. This approach ensures efficient and safe decision-making, even in complex driving environments.

A. Problem Statement

In edge-cloud collaborative intelligent driving systems, we deploy small-scale LLMs on edge devices for real-time motion planning and large-scale LLMs on the cloud to provide efficient support. Edge devices, when processing real-time driving data, may encounter distribution shifts or decreased model confidence due to natural variations (such as changes in lighting or weather) or sensor degradation, which can affect model performance.

Two primary scenarios necessitate the request for support from large models in the cloud:

(1) When the vehicle encounters new or previously unseen objects or situations, increasing decision-making complexity, and the edge model may be insufficient for accurate inference.

(2) For instance, visual obstructions or lighting variations may reduce the accuracy and reliability of edge model predictions. Under such circumstances, leveraging large models in the cloud for deeper analysis can enhance system performance and safety.

B. System Architecture

The proposed system architecture, illustrated in Fig. 2, integrates edge and cloud components to enhance the overall performance of autonomous driving systems. The vehicle employs small-scale LLMs, fine-tuned using instruction-based approaches as shown in Fig. 3, to manage routine driving tasks and process real-time sensor data for immediate decision-making.

Edge Models: We employ LLaMA-Adapter [24], a parameter-efficient tuning mechanism based on the LLaMA language model. LLaMA-Adapter is specifically designed for scenarios where computational resources are constrained, such as autonomous driving. It introduces small, zero-initialized attention modules, which are fine-tuned to adapt to new tasks without modifying the entire pre-trained model. This approach minimizes the additional computational overhead, making it ideal for real-time motion planning on edge devices. The model processes real-time sensor data, including text, vision and LiDAR inputs, to make preliminary driving decisions under normal conditions. Pre-print, manuscript submitted to IEEE.

Cloud Models: In the cloud, large-scale LLMs such as GPT-4 offer advanced computational power for handling more complex and dynamic driving scenarios. Real-time data from various onboard sensors, including cameras, LiDAR, and radar, is collected and preprocessed to extract pertinent features. This preprocessing converts the raw sensor data into a structured format that is amenable to model inference. The processed data is then input into the edge model for initial inference, facilitating efficient and timely driving decisions under varying conditions.

Edge-Cloud Collaboration Workflow: Inspired by [25], we utilize the Alibi Detect library [26] to monitor edge model performance. If anomalies or low-confidence predictions are detected, the system flags those instances and uploads the data to the cloud. The cloud model then performs detailed inference to generate optimized decisions, which are integrated with the edge model's outputs to update the vehicle's driving plan, ensuring safe and efficient operation.

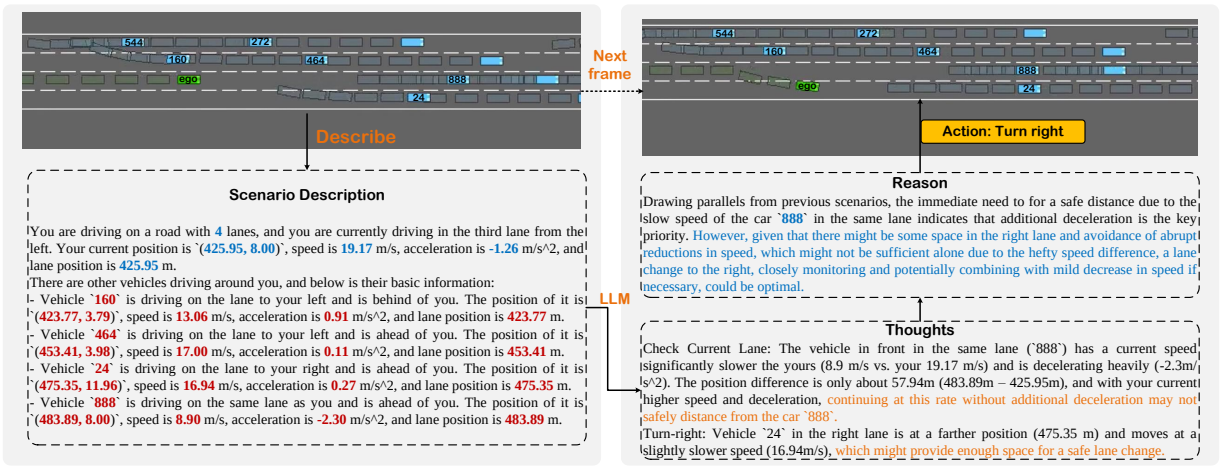

Fig. 4: Edge model performs step-by-step reasoning and decision making in a complex traffic environment

Let $\mathbf{x}$ denote the preprocessed driving data, and the inference result of the edge model is $a = f_{\text{edge}}(\mathbf{x})$, where f_{edge} represents the edge model. We use the Alibi Detect library to perform anomaly detection. If the prediction result $p = cd.\text{predict}(\mathbf{x})$ indicates the presence of data drift or low confidence, cloud model support is requested, resulting in an enhanced decision $a' = f_{\text{cloud}}(\mathbf{x})$, where f_{cloud} represents the cloud model. The overall process is shown in Algorithm 1.

Algorithm 1: EC-Drive: Edge-Cloud Collaborative Motion Planning

Initialization:
Deploy small-scale LLM on edge for motion planning
Deploy large-scale LLM on cloud for motion planning
Initialize Alibi Detect detector cd with reference data
Main Process:
while driving **do**
  Collect driving data $\mathcal{D}$
  Preprocess data: $\mathbf{x} \leftarrow \text{preprocess}(\mathcal{D})$
  Perform edge model inference on $\mathbf{x}$ and execute decision $a = f_{\text{edge}}(\mathbf{x})$
  Performance Monitoring:
  $p \leftarrow cd.\text{predict}(\mathbf{x})$
  if p indicates drift **or** low confidence of edge model **then**
    Request cloud model support and execute enhanced decision $a' = f_{\text{cloud}}(\mathbf{x})$
  end if
end while

IV. Experiments

In this section, we present experimental investigations into the real-time operational capabilities of autonomous driving systems using LLMs under different computational paradigms: Edge, Cloud, and Edge-Cloud Collaborative scenarios. Each subsection details distinct approaches and methodologies—ranging from handling in-vehicle data processing at the edge to leveraging cloud computational power for intensive data analysis and decision-making. This comparative study aims to highlight the efficiency, scalability, and reliability of each model under varied driving conditions and their implications on autonomous driving technologies.

A. Driving on Edge

Scene Description: We transcribe the current driving scene into descriptive text, including the current speed, acceleration, position of the ego vehicle, and information about surrounding vehicles. For example, the ego vehicle is traveling in the rightmost lane of a four-lane road at a speed of 25.0 m/s, with an acceleration of 0.0 m/s², and its lane position is 361.18 m. The information for other vehicles includes their speed, acceleration, and relative position, such as vehicle 496 in the left lane, ahead by 372.81 m, traveling at a speed of 21.2 m/s, with an acceleration of 0.2 m/s².

Reasoning and Thinking: The scene description is embedded into vectors and input into the LLaMA-Adapter. Using CoT techniques, LLaMA-Adapter generates sequential reasoning logic and performs step-by-step logical reasoning. For instance, it first assesses whether the vehicle can accelerate. If not, it evaluates the safety of maintaining the current speed. If necessary, it further evaluates the possibility and safety of changing lanes.

Decision Making: As shown in Fig. 2, the system decodes the final decision from the LLM's response and translates it into corresponding vehicle actions, following the outlined process.

As shown in Fig 4, we demonstrate how the LLM performs step-by-step logical reasoning and decision-making in a complex traffic environment.

B. Driving on Cloud

As depicted in Fig. 5, edge models face significant challenges in real incremental scenarios. Through the identification module, the system selectively uploads data to the cloud-based foundational model, powered by GPT-4, for queries, thereby enhancing motion planning performance. The inference process of the cloud model in real scenarios encompasses three critical

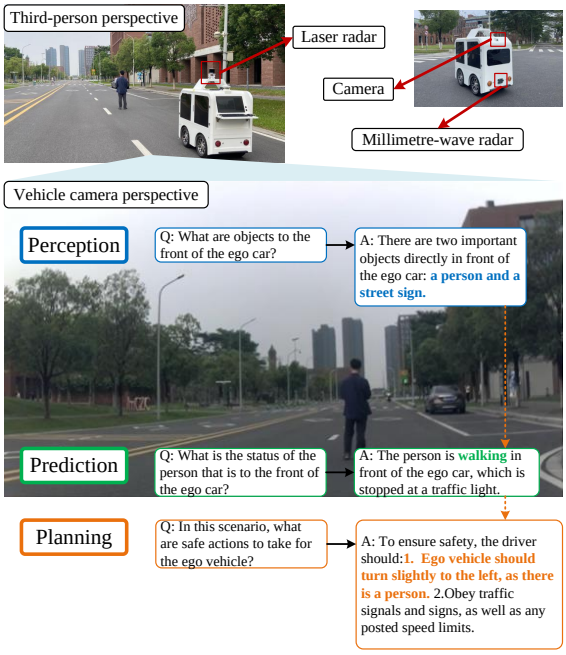

Fig. 5: The cloud model addresses incremental driving scenarios, and the yellow dotted line shows the logical dependencies between stages.

stages: perception, prediction, and planning. These stages are essential for ensuring the model's efficient response.

C. Edge-Cloud Collaborative Motion Planning

This project utilizes data collected by autonomous vehicles at the Guangzhou International Campus of South China University of Technology as the testing benchmark. The dataset comprises images captured from the perspective of autonomous vehicles, with a lower camera angle that aligns closely with practical autonomous driving applications such as delivery and patrol.

Fig. 6 illustrates the inference outcomes of different models in the same scenario. In most cases, edge models (LLaMA-Adapter [24]) demonstrate performance comparable to cloud models (GPT-4 [2]), where invoking cloud models offers limited improvement to the driving task and may lead to resource wastage and unnecessary delays.

Although the edge model is capable of making quick inferences in most cases, the cloud model demonstrates extremely high accuracy when dealing with complex scenarios. For instance, in pedestrian recognition and complex road planning (as shown in the second case of Fig. 6), the cloud model can correct the inference errors made by the edge model, thereby enhancing the overall safety and reliability of the system.

D. Further Analysis and Discussion

Inspired by [27], we evaluate LLMs based on three metrics. Gaze: Assessing the accuracy of LLMs in identifying areas of focus during the driving process. Scene Understanding: Evaluating the precision of LLMs in describing elements present in the current driving scene. Logic: Analyzing the

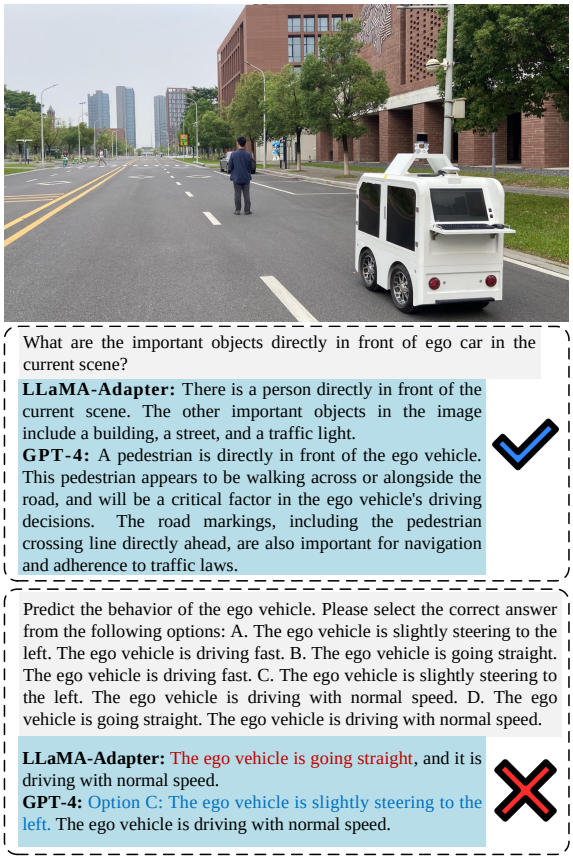

Fig. 6: Comparison of inference results between edge and cloud models in the same scenario.

correctness of the reasoning employed by LLMs in making driving decisions.

Tab. I presents the inference results of models of varying sizes within the dataset. The performance of cloud-based LLMs significantly surpasses that of edge-based small-scale models: As shown in the table, cloud-based LLMs (such as GPT-4 and GPT-4o) achieve higher scores across all three metrics (Gaze, Scene Understanding, and Logic) compared to edge-based small-scale models. Specifically, GPT-4 scores 87.1 in Gaze and 88.9 in Scene Understanding, significantly outperforming the highest scores of edge-based models, which are 66.8 and 59.4, respectively.

Edge-based small-scale LLMs exhibit advantages in specific scenarios: Despite the superior overall performance of cloud-based LLMs, edge-based small-scale models demonstrate significant benefits in environments with limited computational resources or where low-latency responses are required. For instance, edge-based models such as Phi-2-2.7B and TinyLlama-1.1B provide relatively stable performance under constrained resources.

V. Conclusion

This study extensively investigates the application performance of LLMs in autonomous driving systems, leveraging edge computing, cloud computing, and edge-cloud collaborative

Table I: Performance comparison of edge and cloud models in autonomous driving, focusing on relevant driving metrics.

Type	LLM	Gaze (↑)	Scene Understanding (↑)	Logic (↑)
Edge	Moondream	54.7	52.6	49.6
Edge	OpenELM-450M [28]	59.5	52.1	50.5
Edge	TinyLlama-1.1B [29]	61.7	53.9	54.1
Edge	Gemma-2B	65.5	58.6	59.9
Edge	Phi-2-2.7B	66.8	59.4	61.2
Cloud	LLava-7B	72.3	74.2	61.5
Cloud	LLama-Adapter	75.1	79.4	69.6
Cloud	GPT-4o	85.3	86.5	80.3
Cloud	GPT-4	87.1	88.9	81.6

processing. In the edge computing environment, the system swiftly processes real-time driving data, utilizing CoT for logical inference and decision-making. The cloud model exhibits exceptional perception, prediction, and planning capabilities when handling complex driving scenarios. Notably, the edge-cloud collaboration selectively uploads critical data to the cloud, not only enhancing inference speed and conserving communication resources but also significantly reducing system latency. This collaboration also markedly improves the edge model's understanding of incremental and complex scenarios, thereby enhancing overall system performance in motion planning. The experimental results validate the effectiveness and efficiency of the model in practical applications. These findings provide crucial theoretical and practical guidance for the future development of autonomous driving technologies.

REFERENCES

[1] K. Xiong, S. Leng, X. Chen, C. Huang, C. Yuen, and Y. L. Guan, "Communication and computing resource optimization for connected autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 12 652–12 663, 2020.

[2] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[3] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," 2023. [Online]. Available: https://arxiv.org/abs/2307.09288

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, Virtual, May 2020.

[5] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. ICML*, Virtual, Jul. 2021, pp. 8748–8763.

[6] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," 2023. [Online]. Available: https://arxiv.org/abs/2310.01412

[7] W. Wang, J. Xie, C. Hu, H. Zou, J. Fan, W. Tong, Y. Wen, S. Wu, H. Deng, Z. Li *et al.*, "Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving," 2023. [Online]. Available: https://arxiv.org/abs/2312.09245

[8] J. Mao, Y. Qian, H. Zhao, and Y. Wang, "Gpt-driver: Learning to drive with gpt," 2023. [Online]. Available: https://arxiv.org/abs/2310.01415

[9] K. Zhang, J. Wang, N. Ding, B. Qi, E. Hua, X. Lv, and B. Zhou, "Fast and slow generating: An empirical study on large and small language models collaborative decoding," 2024. [Online]. Available: https://arxiv.org/abs/2406.12295

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

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

[12] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," in *Proc. NeurIPS*, vol. 36, New Orleans, LA, USA, Dec. 2023.

[13] J. He, J. Chen, Q. Liu, S. Dai, J. Tang, and D. Liu, "Continual learning with diffusion-based generative replay for industrial streaming data," 2024. [Online]. Available: https://arxiv.org/pdf/2406.15766

[14] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. NeurIPS*, vol. 36, New Orleans, LA, USA, Dec. 2023.

[15] C. Cui, Y. Ma, X. Cao, W. Ye, and Z. Wang, "Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 902–909.

[16] Y. Zheng, Z. Xing, Q. Zhang, B. Jin, P. Li, Y. Zheng, Z. Xia, K. Zhan, X. Lang, Y. Chen *et al.*, "Planagent: A multi-modal large language agent for closed-loop vehicle motion planning," 2024. [Online]. Available: https://arxiv.org/abs/2406.01587

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[18] S. Hu, Z. Fang, Z. Fang, X. Chen, and Y. Fang, "Agentscodriver: Large language model empowered collaborative driving with lifelong learning," 2024. [Online]. Available: https://arxiv.org/abs/2404.06345

[19] L. Wen, D. Fu, X. Li, X. Cai, T. Ma, P. Cai, M. Dou, B. Shi, L. He, and Y. Qiao, "Dilu: A knowledge-driven approach to autonomous driving with large language models," 2023. [Online]. Available: https://arxiv.org/abs/2309.16292

[20] S. Zhang, D. Fu, W. Liang, Z. Zhang, B. Yu, P. Cai, and B. Yao, "Trafficgpt: Viewing, processing and interacting with traffic foundation models," *Transport Policy*, vol. 150, pp. 95–105, 2024.

[21] D. Fu, W. Lei, L. Wen, P. Cai, S. Mao, M. Dou, B. Shi, and Y. Qiao, "Limsim++: A closed-loop platform for deploying multimodal llms in autonomous driving," 2024. [Online]. Available: https://arxiv.org/abs/2402.01246

[22] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," 2023. [Online]. Available: https://arxiv.org/abs/2312.14150

[23] Y. Li, W. Zhang, K. Chen, Y. Liu, P. Li, R. Gao, L. Hong, M. Tian, X. Zhao, Z. Li *et al.*, "Automated evaluation of large vision-language models on self-driving corner cases," 2024. [Online]. Available: https://arxiv.org/abs/2404.10595

[24] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue *et al.*, "Llama-adapter v2: Parameter-efficient visual instruction model," 2023. [Online]. Available: https://arxiv.org/abs/2304.15010

[25] J. Chen, F. Mao, Z. Lv, and J. Tang, "EdgeFD: An edge-friendly drift-aware fault diagnosis system for industrial IoT," in *IEEE International Conference on Communication Technology*, Wuxi, China, Oct. 2023, pp. 390–396.

[26] A. Van Looveren, J. Klaise, G. Vacanti, O. Cobb, A. Scillitoe, R. Samoilescu, and A. Athorne, "Alibi detect: Algorithms for outlier, adversarial and drift detection," 2019. [Online]. Available: https://github.com/SeldonIO/alibi-detect

[27] W. Han, D. Guo, C.-Z. Xu, and J. Shen, "Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving," 2024. [Online]. Available: https://arxiv.org/abs/2401.03641

[28] S. Mehta, M. H. Sekhavat, Q. Cao, M. Horton, Y. Jin, C. Sun, I. Mirzadeh, M. Najibi, D. Belenko, P. Zatloukal *et al.*, "Openelm: An efficient language model family with open-source training and inference framework," 2024. [Online]. Available: https://arxiv.org/abs/2404.14619

[29] P. Zhang, G. Zeng, T. Wang, and W. Lu, "Tinyllama: An open-source small language model," 2024. [Online]. Available: https://arxiv.org/abs/2401.02385