BeVLM: GoT-based Integration of BEV and LLM for Driving with Language

Yang Dong[1], Hansheng Liang[1], Mingliang Zhai[1], Cheng Li[1], Meng Xia[1], Xinglin Liu[1],
Mengjingcheng Mo[2], Jiaxu Leng[2], Ji Tao[1], Xinbo Gao[2]
[1]Chongqing Changan Automobile Co., Ltd.
[2]Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications
dongyang@changan.com.cn hansheng.liang@utt.fr
zhaimingliang@bit.edu.cn licheng@bit.edu.cn qinxiameng@gmail.com
xlliu1336@163.com mo1031@live.com lengjx@cqupt.edu.cn
taoji@changan.com.cn gaoxb@cqupt.edu.cn

Abstract

The integration of large language models (LLMs) with autonomous driving systems has garnered significant attention. However, several challenges persist in this domain. First, aligning LLMs with multi-view visual sequences, commonly encountered in autonomous driving tasks, remains an open problem. Second, the multi-stage nature of autonomous driving, involving perception, prediction, planning, and behavior, necessitates robust contextual understanding. To address these challenges, we first introduce the Bird's Eye View (BEV) feature using a BEV encoder to improve visual context understanding. By fusing BEV representations with textual context, we enhance the alignment between LLMs and multi-view visual data. Second, we augment question-answering (QA) pairs to facilitate cross-modal alignment, bridging the gap between different modalities. Finally, we design a graph-of-thought (GoT) scheme that empowers multimodal LLMs to comprehensively understand both visual and textual contexts. Our experimental results demonstrate the effectiveness of our proposed methods.

1. Introduction

In recent years, integrating large language models into autonomous driving tasks [2, 10–12] has become a hot topic. While some existing approaches have achieved promising results by fusing image and LLM, there are relatively few studies that directly input multi-view visual sequences (4D features) to allow these language models to comprehend and make decisions. In the context of the "driving with language" task, we emphasize the critical role of the language model understanding of 4D features. To address those, we propose a GoT-based Integration of BEV and LLM approach which includes three contributions,

- **BEV Feature Enhancement** We introduce BEV encoder to enhance visual context understanding, facilitating alignment between LLMs and multi-view visual sequences.
- **Cross-Modality Alignment** To further enhance alignment, we introduce additional question-answer pairs during training. These pairs reinforce the semantic consistency between multimodal features and language cues.
- **GoT for Context Enhancement** We incorporate cognitive chains to strengthen context understanding. By considering sequential dependencies, our model gains a deeper understanding of the visual and text context.

2. Datasets

Collection We establish an extensive dataset that serves as a foundational resource for our research and development. The dataset is mainly composed of four distinct subsets [4, 7, 9, 10], which are derived from the nuScenes dataset. Additionally, We incorporate 600,000 in-house data samples into the nuscenes datasets, specifically optimizing for BEV encoder, whose weights are released on GitHub. In our case, the task is further complicated by the need to align the BEV representation of the scenes with the language feature spaces. This alignment is crucial for accurately interpreting and responding to questions posed in a natural language context.

Pre-processing First of all, we standardize all data to conform to the "llama" style format. Then, we add important objects for each question except on pre-training, such as *"I will give you some objects around the ego car, those objects will be considered for future reasoning.* $\langle c1, CAM\_FRONT, 981.7, 561.7\rangle \ldots$*, [QUESTION]"*. Through this data construction, we aim

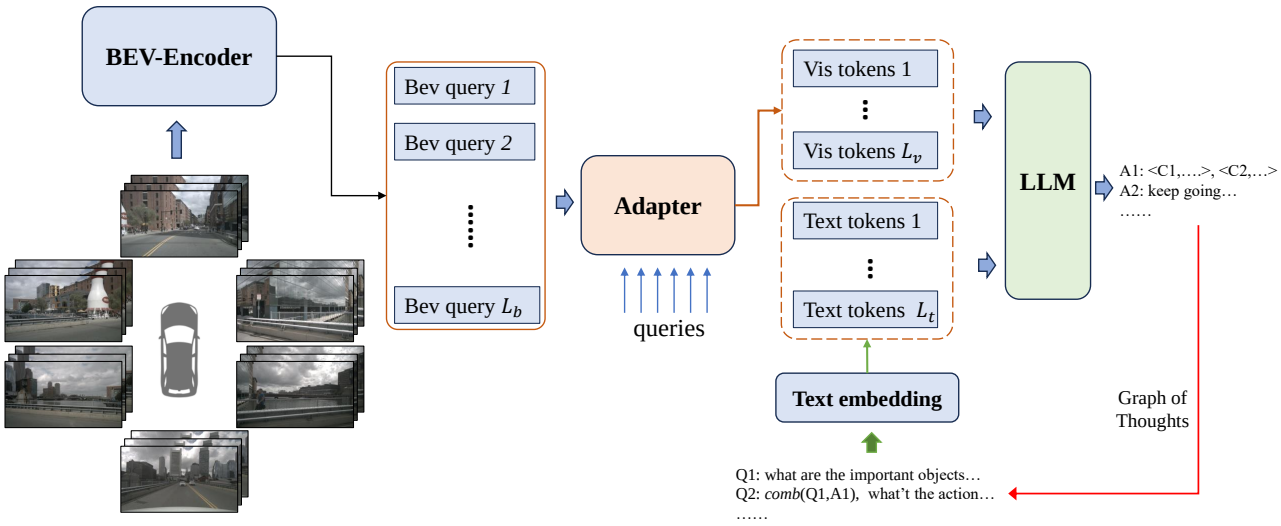

Figure 1. The pipeline of our BeVLM. Our method takes multi-view images as input and extracts features separately using the BEV encoder. Next, we utilize an adapter to extract visual tokens, aligning visual features with textual features. Finally, we input all features into the LLM to obtain the answer.

to make the pre-trained model sensitive to the camera perspective and position of important objects.

3. Methods

3.1. Overview

The overall pipeline of BeVLM is illustrated in Figure 1. The multi-view video is processed by a Large Vision Language Model (LVLM) to perform a special GoT reasoning to device the driving tasks. The BeVLM involves a BEV encoder [5], a LLM [3] and a BEV-Adapter. BeVLM incorporates BEV queries as visual prompt for enhanced scene understanding capability, and further refines advance reasoning ability with a novel GoT process.

3.2. BEV-LLM Alignment

As shown in the adapter module in Figure 1, the input from multi-view videos are first processed by a BEV encoder to extract features, resulting in BEV queries $q_b \in \mathbb{R}^{l_b \times d}$, where l_b is the number of BEV queries, d is the feature dimension of each query. Next, the features are transformed using an adapter module $\mathcal{A}$ to align them with textual features f_t. And we initialize 90 learnable queries q_l for interacting with BEV features, the formula is described as follows:

$$f_b = \mathcal{A}(concat(q_b, q_l)).$$

This adapter module plays a crucial role in aligning 4D visual features with textual features. After combining the visual and textual features f_t, they are fed into a large language model $\mathcal{M}$ to generate answers $A = \mathcal{M}(concat(f_b, f_t))$ corresponding to the given questions.

3.3. Graph of Thoughts

After referencing existing work on graph of thoughts [1, 3] and observing the training data, we further refine the graph-of-thought scheme based on DriveLM[9], as shown in Figure 2, aiming to enhance VQA performance in the planning stage. For each question in the graph of thoughts, instead of merely using the QA pairs from the previous stage as a prompt for the question, we consider the specific information directly needed to answer the question and accordingly provide that information as the prompt. This approach is designed to more precisely guide the reasoning process, ensuring that questions are answered accurately. Meanwhile, this approach reduces the need of token length, enabling us to train the model with a relatively short context length.

4. Experiments

4.1. Experimental Setup

Dataset The dataset is divided into three parts: train set, test set, and validation set. Notably, the test set comprises 66 question-and-answer pairs. Due to submission constraints, some of our ablation experiments were conducted on the test set. It's worth explaining that we intentionally removed the test set from the training set.

Implementation details For the visual feature extraction backbone, we use the pre-trained model following the original setting [5, 8]. The number of BEV queries l_b is 1600, and the dimension of BEV feature d is set to 256. The maximum of sequence length is 512, and the adapter module

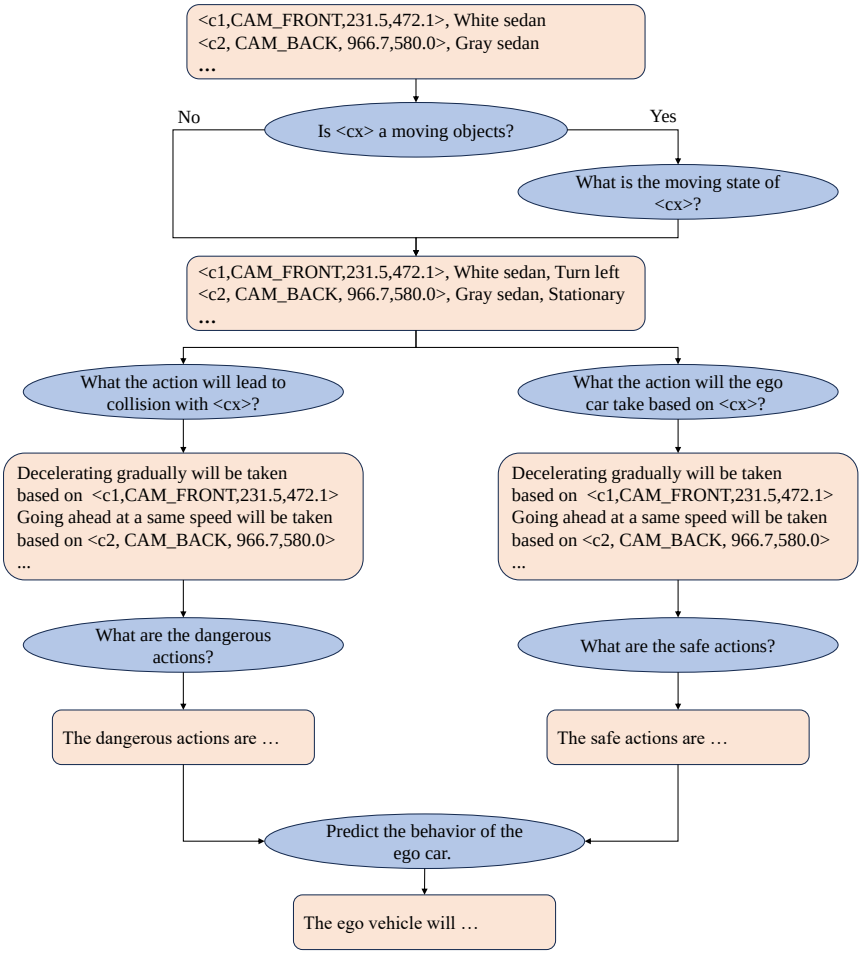

Figure 2. Graph of thoughts for autonomous driving tasks.

uses a 6-layer self-attention. As for pretraining, we use the AdamW [6] optimizer with an initial learning rate of 1e-4 and warmup 1 epoch, and trainable parameters are the norm, bias, lora layer of LLM, visual projection, BEV projection, and adapter. As for fine-tuning, we set a learning rate of 1e-5 and only train the norm, bias of LLM. All of our experiments were conducted on 8 Tesla A100-SXM 40G GPUs.

4.2. Effectiveness of the Framework

As mentioned in section 3.1, We collect 2 million question-answer pairs and perform end-to-end pretraining on our model using these data. Subsequently, we fine-tune the model using DriveLM train data. Table 1 presents various encoder ablation experiments, with all tests conducted on the test set. From Table 1, EXP ID 1 $\sim$ 4, it is evident that even without pretraining on the additional 2 million data samples, the BEV-encoder outperforms ClipVit. Based on this result, we do not retest the BEV-encoder after pretraining it on the 2 million data samples. On the other hand, while the BEV-encoder enhances the visual perception of the multimodal model, adjustments to the query data within the adapter impact the alignment between vision and language. Consequently, the BEV-encoder does not perform as well in terms of language scores.

In Table 1, we demonstrate the significance of visual features in large multimodal models. EXP ID 5 corresponds to the original configuration with 10 visual tokens. By increasing the number of visual tokens to 90, the final score improved by 6.29 (EXP ID 6).

4.3. Optimization of Perception

After analysis, we found that improved perceptual capability is crucial, particularly for addressing the question *"what are the important objects. . . "*. Consequently, we conducted a detailed analysis of the distribution of "import objects" within the training set from various perspectives. Specifi-

TEST SET							
EXP ID	Visual Encoder	Pretrain	Final	Acc	ChatGPT	Match	Language
1	ClipVit$_{large}$ [8]	-	59.16	88.89	61.44	35.63	48.42
2	ClipVit$_{large}$	2M QA	63.12	88.89	67.56	38.88	52.71
3	BEV-encoder [5]	-	72.50	81.82	93.63	36.88	56.53
4	BEV-encoder	2M QA	-	93.75	-	-	53.42
VAL SET							
5	BEV-encoder	2M QA	51.22	68.38	52.91	35.45	46.45
6	+ 90 visual token (sec. 3.2)	2M QA	57.51	86.67	60.01	34.29	46.55
7	+ Perception opt (sec. 4.3)	2M QA	70.9	75.78	62.52	95.21	58.49
8	+ caption opt (sec. 4.3)	2M QA	75.19	75.78	62.58	93.25	81.77
9	+ Planning opt + GoT (sec. 4.4)	2M QA	76.17	75.78	62.4	93.56	86.79
10	+ ensemble	2M QA	77.59	75.97	64.72	95.81	86.71

Table 1. Ablation of visual encoder on **TEST set** due to submission constraints. 2M QA refers to the all the datasets mentioned in 2.

cally, we investigated the following dimensions: (1) the area distribution of 2D detection bounding boxes, (2) the distribution of important objects across different views, (3) the offset distribution between the center points of 2D detection boxes and ground truth, and (4) the category distribution of important objects. Based on these distributions, we devised several optimization strategies.

s1. We utilize the pre-trained grounding DINO model to perform 2D object detection on both the training set and validation set.

s2. We use the training set's bounding box coordinates and corresponding visual descriptions to fine-tune the Qwen-VL model, enhancing its ability to generate accurate visual descriptions based on input images and coordinates.

s3. Using the fine-tuned Qwen-VL model, we generate visual descriptions for all detected objects.

s4. Based on the distribution patterns of important objects, we selectively filter out a subset of significant objects.

s5. We construct an augmented training set by expanding the number of instances for the "import object" question by $\times 10 \sim \times 20$.

s6. Finally, we perform end-to-end training for this question using the BEV-encoder.

The step *s2* and *s3* above are designed to generate more accurate captions, resulting in significant improvements in language scores, which could be found in Table 1 EXP ID 8 and EXP ID 7. Additionally, other steps focus on perception optimization. Comparing the results of EXP ID 6 and EXP ID 7 in Table 1, we observe significant improvements across multiple metrics after perceptual optimization.

4.4. Optimization of Planning and GoT

We choose Bunny-Llama3-8B as the VLM for GoT inference in the planning stage, due to it performs better than our base model when answering planning questions on the validation set. The training on the competition dataset is divided into two phases: fine-tuning with all the original QA data, including perception, prediction, planning and behavior stage, and then continuing fine-tuning on the constructed GoT data. We construct the prompts for each question in GoT using the ground truth from the training set. During inference, we follow the sequence outlined in Figure 2, using the answers from preceding questions as the prompts for subsequent questions. Notably, we do not use all the answers from the previous stages, as discussed in 3.3 .

As the result of EXP ID 9 shown in Table 1, after the two-phase fine-tuning, the GPT score reaches 64.72. This indicates that for complex reasoning problems like "What are dangerous/safe actions" incorporating explicit prompts during the reasoning process is helpful for achieving correct reasoning outcomes.

5. Conclusion

The BeVLM introduced in this report achieved excellent results in the CVPR2024 Challenge. By developing a method that aligns BEV features with the LLM, along with GoT, our approach effectively tackles the complex task of comprehending spatial relationships in driving scenarios. The innovative use of an adapter module to bridge the gap between 4D visual features and textual information has proven to be a key component in enhancing the model's predictive accuracy and planning capabilities.

References

[1] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate prob-

lems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690, 2024. 2

[2] Yaodong Cui, Shucheng Huang, Jiaming Zhong, Zhenan Liu, Yutong Wang, Chen Sun, Bai Li, Xiao Wang, and Amir Khajepour. Drivellm: Charting the path toward full autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*, 2023. 1

[3] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 2

[4] 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 *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024. 1

[5] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18580–18590, 2023. 2, 4

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[7] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 1

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 2021. 2, 4

[9] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023. 1, 2

[10] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*, 2024. 1

[11] Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*, 2024.

[12] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, MA Tao, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. Dilu: A knowledge-driven approach to autonomous driving with large language models. In *The Twelfth International Conference on Learning Representations*, 2023. 1