V2V-GoT: Vehicle-to-Vehicle Cooperative Autonomous Driving with Multimodal Large Language Models and Graph-of-Thoughts

Hsu-kuang Chiu[1,2] Ryo Hachiuma[1] Chien-Yi Wang[1] Yu-Chiang Frank Wang[1]
Min-Hung Chen[1*] Stephen F. Smith[2*]

[1]NVIDIA, [2]Carnegie Mellon University, [*]equally advising

Abstract

Current state-of-the-art autonomous vehicles could face safety-critical situations when their local sensors are occluded by large nearby objects on the road. Vehicle-to-vehicle (V2V) cooperative autonomous driving has been proposed as a means of addressing this problem, and one recently introduced framework for cooperative autonomous driving has further adopted an approach that incorporates a Multimodal Large Language Model (MLLM) to integrate cooperative perception and planning processes. However, despite the potential benefit of applying graph-of-thoughts reasoning to the MLLM, this idea has not been considered by previous cooperative autonomous driving research. In this paper, we propose a novel graph-of-thoughts framework specifically designed for MLLM-based cooperative autonomous driving. Our graph-of-thoughts includes our proposed novel ideas of occlusion-aware perception and planning-aware prediction. We curate the V2V-GoT-QA dataset and develop the V2V-GoT model for training and testing the cooperative driving graph-of-thoughts. Our experimental results show that our method outperforms other baselines in cooperative perception, prediction, and planning tasks. Our code and dataset are publicly released to facilitate open-source research at https://eddyhkchiu.github.io/v2vgot.github.io/.

1. Introduction

Today's autonomous vehicles rely mainly on mounted cameras or LiDAR sensors to perceive the world, understand the dynamic surrounding scenes, and take driving decisions. Such reliance on the local sensors can be limiting in situations where potential obstacles are occluded by other large nearby objects, such as buses or trucks. To mitigate this safety issue, recent research has proposed V2V-enabled cooperative perception [2, 3, 5, 22–25, 27]. In a cooperative perception framework, each Connected Autonomous Vehicle (CAV) shares its individual perception information with others and improve overall detection and tracking accuracy.

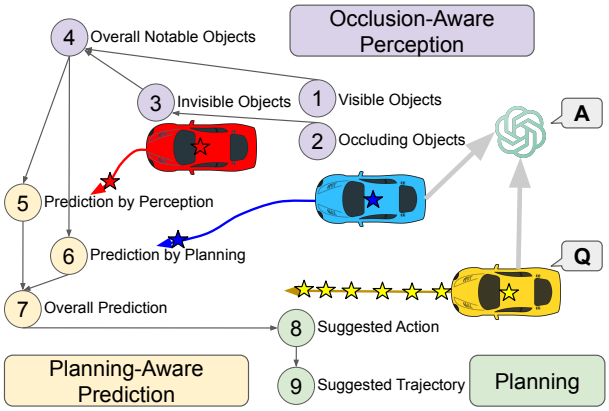


Figure 1. Illustration of our proposed graph-of-thoughts reasoning framework for cooperative autonomous driving. All Connected Autonomous Vehicles (CAVs) share their perception features with the Multimodal Large Language Model (MLLM), as illustrated by the grey arrows. Any CAV can ask the MLLM to provide a suggested future trajectory or answer perception or prediction questions. The MLLM fuses the perception features from all CAVs and performs inference by following the graph-of-thoughts. If two QA nodes are connected by a directed edge in the graph, as illustrated by black arrows, the answer of the parent node QA is used as the input context of the child node QA. Other colored curved arrows illustrate the predicted or suggested future trajectories. Color stars represent current locations of objects, predicted or suggested future waypoints.

How to best realize cooperative perception remains a question for research. One recent trend in autonomous driving research is using Large Language Models (LLMs) to build end-to-end driving systems that digest the raw sensor input and then generate the planning results. Such LLM-based autonomous driving models [16, 18, 19] are motivated by the belief that LLM-based approaches could benefit from the reasoning and generalization capabilities that large-scale pre-trained data provides. However, most LLM-based driving research focuses on individual autonomous vehicles without cooperative perception or planning.

Some recent pioneering research works [4, 6, 7, 10] have started to explore LLM-based cooperative autonomous driving. However, the V2V cooperation in prior works [6, 10] only involves the LLM as a pure language-based negotiator to resolve planning conflicts, which has not taken advantage of the multimodal understanding ability of MLLMs. In contrast, previous work V2V-LLM [4] adopts MLLMs to fuse perception features from multiple CAVs and provide answers to perception and planning questions. Although V2V-LLM [4] shows promising results, it has not considered the additional benefit of applying chain-of-thoughts or graph-of-thoughts reasoning on MLLMs.

In this paper, we propose a new graph-of-thoughts reasoning framework designed for MLLM-based cooperative autonomous driving, as illustrated in Figure 1. We follow the problem setting from V2V-LLM [4], where all CAVs share perception features with a MLLM. Any CAV can ask the MLLM to answer a driving question. The MLLM fuses the perception features from all CAVs and performs inference to provide the answer. Unlike previous work [4, 16], we propose a novel graph-of-thoughts reasoning framework specifically designed for cooperative autonomous driving scenarios, as illustrated in Figure 1 and 2.

To verify the effectiveness of our proposed idea, we first curate the V2V-GoT-QA dataset, built upon the V2V4Real [26] cooperative driving dataset. Our V2V-GoT-QA includes 9 different types of QA samples, as illustrated in Figure 2. Our proposed **occlusion-aware perception** questions (Q1 - Q4) consider visible, occluding, and invisible objects. Our proposed **planning-aware prediction** questions (Q5 - Q7) include prediction by perception features and prediction by other CAVs' current planned future trajectories. Our planning questions (Q8 - Q9) provide the suggested action settings and suggested waypoints of future trajectories to avoid potential collisions. Different types of QA are connected by a directed edge in our graph-of-thoughts, as shown in Figure 1. The answer of the parent node QA is used as the input context of the child node QA.

In addition to the newly curated dataset, we also develop our baseline model V2V-GoT, as shown in Figure 3. Unlike the prior work V2V-LLM [4] and DriveLM [16] that only take the perception features at the current timestep as visual input, our V2V-GoT includes the perception features from both current and previous timesteps to better capture the temporal dynamics of the surrounding objects.

In our experiments, we compare the final planning performance of our proposed V2V-GoT and other baseline methods from the prior work V2V-LLM [4], including *no fusion*, *early fusion*, and *intermediate fusion*. Experimental results show that our model outperforms all other baselines in the planning tasks, achieving the lowest collision rates and L2 errors. Furthermore, our ablation study also verifies that the newly proposed QA types in our graph-of-thoughts, namely occlusion-aware perception QAs (Q1 - Q4) and planning-aware prediction QAs (Q5 - Q7), are helpful in achieving better cooperative perception, prediction, and planning performance for MLLM-based cooperative autonomous driving.

Our contribution can be summarized as follows.

- We present a novel graph-of-thoughts reasoning framework for MLLM-based cooperative autonomous driving. The approach includes QA types designed specifically for cooperative driving, including **occlusion-aware perception** and **planning-aware prediction**.
- We curate the V2V-GoT-QA dataset and develop the V2V-GoT model to establish the benchmark and baseline method for MLLM-based cooperative autonomous driving with graph-of-thoughts reasoning.
- Experimental results demonstrate the effectiveness of V2V-GoT in overall perception, prediction, and planning performance for cooperative autonomous driving.

2. Related Work

2.1. Vehicle-to-Vehicle Cooperative Perception

V2V cooperative perception algorithms are proposed to improve the overall detection and tracking accuracy in cooperative driving scenarios by sharing perception information between CAVs. Pioneering works F-cooper [1] and V2VNet [20] propose the intermediate fusion approach that shares perception feature maps, achieving a good balance between performance and communication cost. V2V4Real [26] is the first real cooperative driving dataset available worldwide with detection and tracking benchmarks that evaluate the real-world performance of numerous abundant cooperative detection [23–25] and cooperative tracking algorithms [3, 17]. However, this group of research focuses on perception and has not explored the planning part of autonomous driving. In contrast, our work covers both cooperative perception and planning.

2.2. LLM-based Autonomous Driving

A more recent autonomous driving research trend is applying an LLM to build end-to-end autonomous driving models due to its promising reasoning ability. Pioneering work GPT-driver [12, 13] encodes the state of the ego vehicle and its object detection results into text and allows LLM to identify notable objects and suggest a future trajectory. Other works [15, 21] apply MLLMs to perform scene understanding from the image input. More advanced work, DriveLM [16], and LingoQA [14] develop MLLM-based end-to-end models to further generate the suggested action or future trajectory as output. DriveLM [16] also considers graph-of-thoughts but only for a single autonomous vehicle without V2V cooperation. This group of research focuses only on individual autonomous vehicles without coopera-

tive perception or planning. In contrast, our work develops MLLMs-based models for V2V cooperative driving.

2.3. LLM-based Cooperative Autonomous Driving

A few very recent works [4, 6, 7, 10] start exploring LLM-based cooperative autonomous driving. CoDrivingLLM [6] uses an LLM as a pure language-based conflict coordinator in simulated cooperative driving scenarios in an intersection. CoLMDriver [10] uses a Vision Language Model (VLM) as the initial planner of an individual autonomous vehicle without cooperation, while the V2V cooperation module still uses an LLM as a negotiator based on pure language in simulated environments. Neither prior work has taken advantage of the multimodal understanding ability of MLLMs in their V2V cooperation modules nor verified their methods in real-world cooperative driving datasets. Another pioneering work V2V-LLM [4] proposes an MLLM-based cooperative driving model and shows promising results in real-world datasets. However, the chain-of-thoughts or graph-of-thoughts reasoning capabilities have not been explored in this prior work. In contrast, our work proposes a novel graph-of-thoughts reasoning framework, including the special QAs designed for cooperative driving, such as occlusion-aware perception and planning-aware prediction.

3. V2V-GoT-QA Dataset

3.1. Problem Setting

We build on the MLLM-based cooperative autonomous driving problem setting from V2V-LLM [4]. The MLLM fuses the perception features from all CAVs and performs inference to answer a question. Unlike the prior work V2V-LLM [4], we propose a novel graph-of-thoughts reasoning framework, including the ideas of **occlusion-aware perception** and **planning-aware prediction**. We design 9 types of question-answer pairs in our V2V-GoT-QA dataset, including perception, prediction, and planning, as illustrated in Figure 2. Different types of QAs are connected by directed edges in the graph-of-thoughts, as shown in Figure 1 and 4a. The answer of the parent node QA is used as the input context of the child node QA.

3.2. Dataset Details

Following V2V-LLM [4], we use V2V4Real [26] as the base dataset and curate various QA pairs. V2V4Real [26] has 7105 training frames and 1993 testing frames. For each frame that is 3 seconds before the end of each driving sequence in V2V4Real [26], we create one QA pair for each CAV per QA type. Overall, the resulting V2V-GoT-QA dataset has 110610 training and 31014 testing QA pairs.

3.3. Dataset Curation

We use the ground-truth bounding box annotations, each CAV and object's future trajectory, and their geometric relationships in V2V4Real [26] to curate the QA pairs of V2V-GoT-QA dataset. In our proposed graph-of-thoughts structure, the answer of the parent node QA is used as the input context of the child node QA. For training data, we use the ground-truth answer to generate the input context of the child node QA. During inference, we use the model inference output of the parent node QA to generate the input context of the child node QA.

The QA types included in the V2V-GoT-QA dataset span tasks relating to Perception (Q1 - Q4), Prediction (Q5 - Q7), and Planning (Q8 - Q9). Details of each QA design rationale, how they are connected, dataset curation, and evaluation metrics are described in the following.

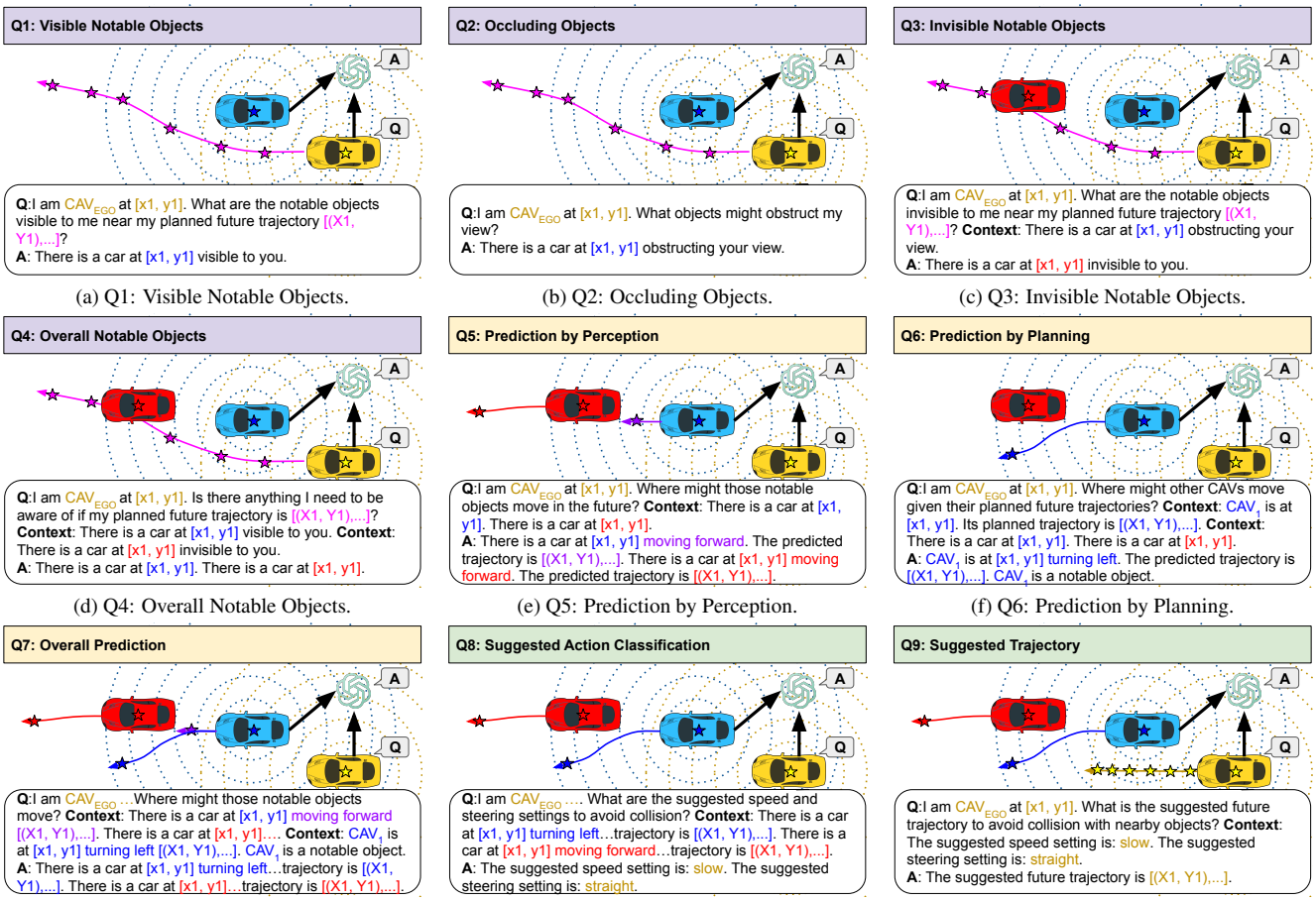
3.4. Perception QAs

The main goal of the perception QAs is to identify notable objects near ego CAV's current planned future trajectory. We propose the idea of **occlusion-aware perception** to the graph-of-thoughts reasoning and divide this task into subtasks of identifying notable objects that are visible (Figure 2a) and invisible (Figure 2c) to the ego CAV separately and then merge the results. This divide-and-conquer design can potentially make it easier for the model to learn which part of the perception feature maps to focus on in order to find the notable objects.

Q1. Visible Notable Objects (Figure 2a): We ask the MLLM to identify notable objects that are visible to the ego CAV and near its current planned future trajectory. Six waypoints from a planned future trajectory in the next 3 seconds are used as the reference trajectory in the question. To generate the ground-truth answer in the training data, at most 3 ground-truth objects within 10 meters of the reference trajectory that are detected by the ego CAV's individual 3D object detector are used as the ground-truth answer. We use the F1 score as the evaluation metric.

Q2. Occluding Objects (Figure 2b): Before identifying notable objects invisible to the ego CAV, locating the occluding objects first may provide useful information. To generate the ground-truth answer in the training data, at most 3 ground-truth objects closest to the ego CAV that are not occluded by other objects are used as the answer. We use the F1 score as the evaluation metric.

Q3. Invisible Notable Objects (Figure 2c): We ask the MLLM to identify notable objects that are invisible to the ego CAV and near its current planned future trajectory. In addition to the question, we also provide the input context from the answer of Q2. Occluding Objects (Figure 2b). Such context could be useful for the model to identify notable objects that are occluded from the ego CAV's field of view by focusing on the occluded region but in other CAVs'



(a) Q1: Visible Notable Objects. (b) Q2: Occluding Objects. (c) Q3: Invisible Notable Objects.

(d) Q4: Overall Notable Objects. (e) Q5: Prediction by Perception. (f) Q6: Prediction by Planning.

(g) Q7: Overall Prediction. (h) Q8: Suggested Action Classification. (i) Q9: Suggested Trajectory.

Figure 2. Illustration of V2V-GoT-QA's 9 types of QA pairs: Perception (Q1 - Q4), Prediction (Q5 - Q7), and Planning (Q8 - Q9). The black arrows pointing at the MLLM indicate the perception data from CAVs. Other colored arrows represent predicted or suggested future trajectories.

perception feature maps. To generate the ground-truth answer in the training data, at most 3 ground-truth objects within 10 meters of the reference trajectory that are not detected by the ego CAV's individual 3D object detector are used as the answer. We use the F1 score as the evaluation metric.

Q4. Overall Notable Objects (Figure 2d): This question simply takes the answers from Q1. Visible Notable Objects (Figure 2a) and Q3. Invisible Notable Objects (Figure 2c) as the input context and merges them to generate the final notable object identification answer, which are then used as the input context for the subsequent prediction questions. We use the F1 score as the evaluation metric.

3.5. Prediction QAs

Predicting the future trajectories of notable objects is critical for the final planning task to suggest a future trajectory for the ego CAV that can avoid potential collisions. We propose the idea of **planning-aware prediction** to the graph-of-thoughts reasoning and divide the prediction task into two parts: prediction by perception (Figure 2e) and prediction by planning (Figure 2f), and then merge their results to generate the final prediction output.

Q5. Prediction by Perception (Figure 2e): We ask the MLLM to predict the future trajectories of the notable objects and classify their movement into one of the following 4 categories: *moving forward*, *turning left*, *turning right*, and *staying at the same location*. In addition to the question, the MLLM also takes the answer of Q4. Overall Notable Objects (Figure 2d) as context. With this context, the MLLM knows where the notable objects are at the current timestep. This prediction process relies mainly on the perception features in the current and previous timesteps from all CAVs. We use the ground-truth future trajectories of the notable objects and perform motion classification with heuristic threshold values on the trajectories as the ground-truth answers. We use L2 error as the evaluation metric.

Q6. Prediction by Planning (Figure 2f): If a notable object suddenly accelerates, decelerates, or changes directions, it may be difficult to predict its future trajectory by only observing its current and past locations and motions. However, if a notable object is also a CAV, it can directly

share its planned future trajectory with other CAVs. That planning information could be used as a more accurate predicted future trajectory to help other CAVs' prediction and planning procedures. In Q6, we include the current planned future trajectories of other CAVs and the answer of Q4. Overall Notable Objects (Figure 2d) as the input context in the question. To generate the answer, we perform motion classification in the same manner as for Q5. We further identify whether other CAVs are notable objects or not in Q6's answer. To measure performance, we calculate the binary classification accuracy in determining whether the model can identify if other CAVs are notable objects or not.

Q7. Overall Prediction (Figure 2g): This question takes the answers from Q5. Prediction by Perception (Figure 2e) and Q6. Prediction by Planning (Figure 2f) as the input context and merges them to generate the final prediction answer. If the answer of Q6 indicates that another CAV is a notable object, the model is expected to learn to use that CAV's planned future trajectory as its predicted future trajectory. The merged prediction output is then used as the input context for subsequent planning tasks. We compute L2 errors to measure the performance.

3.6. Planning QAs

We first classify the suggested action's speed and steering settings, and then use them as the input context to provide the suggested future trajectory.

Q8. Suggested Action Classification (Figure 2h): We use the prediction results from the answer of Q7. Overall Prediction (Figure 2g) as the input context and ask the MLLM to provide the suggested speed and steering settings. We adopt a similar approach from DriveLM [16]. The speed setting can be one of the following 5 categories: *fast*, *moderate*, *slow*, *very slow*, and *stop*. The steering setting can be one of the following 5 categories: *left*, *slightly left*, *straight*, *slightly right*, and *right*. We classify the speed and steering settings by applying heuristic threshold values on the average difference between consecutive waypoint coordinates. To measure performance, we calculate the L1 errors between the output and ground-truth answer indices on the suggested speed and steering settings. For example, if the output speed and steering settings of a sample are *fast* and *slightly left*, while the ground-truth answer settings are *very slow* and *right*, the L1 error of this sample is $|0-3|+|1-4|=6$.

Q9. Suggested Trajectory (Figure 2i): We use the suggested speed and steering setting from the answer of Q8. Suggested Action Classification (Figure 2h) as context and ask the LLM to provide the 6 waypoints of the suggested future trajectory for the next 3 seconds that avoids collisions. We use the ground-truth future trajectory as the ground-truth answer. To measure performance, we calculate the L2 errors and collision rates.

4. V2V-GoT Model

4.1. Architecture

We build our V2V-GoT model with LLaVA-v1.5-7b [11] as the base MLLM architecture, as shown in Figure 3. Unlike the original LLaVA [11] that uses an image encoder, we apply a LiDAR-based 3D object detector, PointPillars [9], to extract perception features from each individual CAV's point cloud. Unlike V2V-LLM [4]'s model that only uses the perception features at the current timestep, our model uses the perception features at the current and previous timesteps from all CAVs as the input of the project layers in the MLLM to generate the visual tokens. The MLLM takes the visual tokens and the language tokens from the question and the context as input and generates the final answer in the natural language format.

4.2. Training and Inference

We follow the similar training settings and hyperparameters from V2V-LLM [4] and LLaVA [11]. During training, we only train the projector layers and the LoRA [8] parts of the model and freeze the remaining parts. Our model is trained with NVIDIA H100-80GB GPUs and a batch size of 32. We use Adam optimizer with a starting learning rate $2e^{-5}$ and train our model for 10 epochs on our V2V-GoT-QA training dataset. During testing inference, we follow our proposed graph-of-thoughts illustrated in Figure 1 and 4a. If two QA nodes are connected by a directed edge in the graph-of-thoughts, the inference model output of the parent node QA is used as the input context of the child node QA.

5. Experiment

5.1. Baseline Methods

We adopt V2V-LLM [4] and its baseline methods with different fusion approaches, such as *no fusion*, *early fusion*, and *intermediate fusion*, to compare with our proposed V2V-GoT. In *no fusion*, the MLLM only takes a single CAV's perception features as visual input and answers the driving-related question. This baseline shows the performance in driving scenarios without cooperative perception. In *early fusion*, the union of the point cloud of both CAVs is used to extract perception features, which are then used as the visual input of the MLLM. This approach usually requires high communication costs and is thus impractical for actual deployment. In *intermediate fusion*, such as CoBEVT [23], V2X-ViT [24], and AttFuse [25], cooperative 3D object detectors are used to extract perception features as MLLM's visual input. These approaches usually achieve a good balance of performance and communication cost. V2V-LLM [4] was proposed as a new fusion method that uses MLLM to fuse the scene-level and object-level features from multiple CAVs and achieves the best perfor-

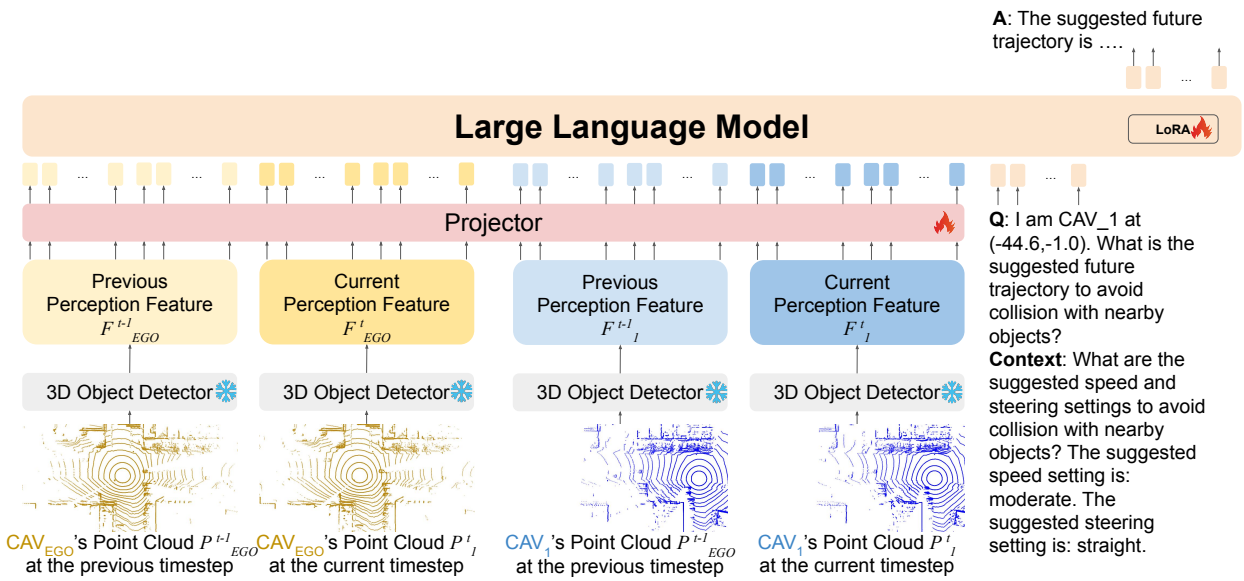


Figure 3. Model architecture of V2V-GoT.

mance in V2V-QA [4]'s perception and planning tasks.

Our proposed V2V-GoT-QA dataset and the prior V2V-QA [4] dataset use the same base dataset V2V4Real [26] and have the same final cooperative planning tasks: our Q9. Suggested Trajectory (Figure 2i) and V2V-QA [4]'s Q5. Therefore, we can directly compare our method with the baseline methods on the final planning performance. Note that prior work V2V-LLM [4] only takes the perception features at the current timestep as the visual input. To have a fair comparison with our V2V-GoT, we modify their baseline models to also take the perception features at both of the current and previous timesteps as visual input, following our model architecture shown in Figure 3.

5.2. Quantitative Results

Table 1 shows the testing performance of V2V-GoT in the planning task of V2V-GoT-QA in comparison with baseline methods. Our V2V-GoT is seen to achieve the best final planning performance with the lowest L2 errors and collision rates compared to all baselines with different fusion approaches. In particular, the results indicate that the introduction of graph-of-thoughts reasoning designed for cooperative autonomous driving indeed boosts the performance in cooperative autonomous driving planning tasks.

5.3. Ablation Study

To further verify the impacts of our newly proposed ideas of **occlusion-aware perception** and **planning-aware prediction** in our graph-of-thoughts reasoning framework, as illustrated in Figures 1 and 4a, we perform an ablation study on two additional graph structures: a *simplified perception graph* shown in Figure 4b and a *simplified prediction graph* shown in Figure 4c. The ablation results are given in Table 2.

5.3.1. Simplified Perception Graph

To verify the impact of **occlusion-aware perception**, we first create additional training data samples of Q4. Overall Notable Objects (Figure 2d) but without context. Then we train another model with newly generated data samples and the existing training dataset, and perform testing inference following the *simplified perception graph* Figure 4b. From Table 2, we can see that the *simplified perception graph* results in worse perception performance in Q4 and worse subsequent prediction and planning performance in Q7 and Q9, compared to our proposed V2V-GoT. This result indicates the importance of our occlusion-aware perception design in our final proposed graph-of-thoughts.

5.3.2. Simplified Prediction Graph

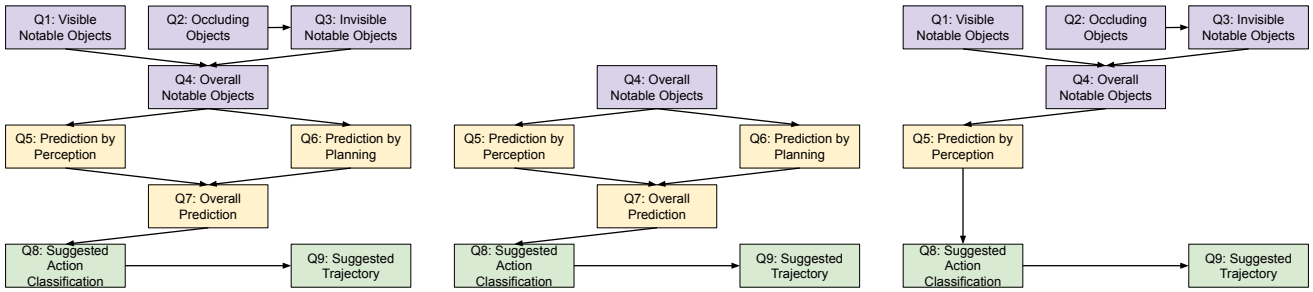
To verify the impact of **planning-aware prediction**, we use the same V2V-GoT model but run testing inference on the *simplified prediction graph* (Figure 4c). From Table 2, we can see that this *simplified prediction graph* approach also results in poorer planning performance than our proposed V2V-GoT, due to the poorer prediction result, indicating the importance of our planning-aware prediction design. Furthermore, note that use of either simplified graph individually already offers performance improvement over the baseline V2V-LLM [4] (compare respective CR(%) values in Tables 1 and 2).

6. Communication Cost

The communication cost of V2V-GoT is the same as the cost of the prior work V2V-LLM [4]. Although V2V-LLM [4] only uses the perception features at the current timestep and V2V-GoT uses the features at the current and the previous timesteps, the same perception features only

Table 1. Testing performance of V2V-GoT in the planning task of V2V-GoT-QA dataset, in comparison with baseline methods, which are adopted from V2V-LLM [4]. To have a fair comparison to our proposed V2V-GoT, we modified baseline methods to also take perception features at the current and the previous timesteps as visual input. L2: L2 distance error. CR: Collision rate. Comm: Communication cost. In each column, the **best** results are in boldface, and the second-best results are in underline.

Method	L2 (m) ↓				CR (%) ↓				Comm(MB) ↓
	1s	2s	3s	Average	1s	2s	3s	Average	
No Fusion	3.47	5.79	8.26	5.84	1.48	4.24	7.72	4.48	**0**
Early Fusion	3.48	5.61	7.82	5.63	1.16	3.51	5.66	3.44	1.9208
Intermediate Fusion									
AttFuse [25]	3.65	6.21	8.75	6.20	1.19	4.41	6.38	3.99	0.4008
V2X-ViT [24]	3.46	5.80	8.19	5.81	1.45	4.24	6.59	4.09	0.4008
CoBEVT [23]	3.38	5.42	7.46	5.42	1.31	4.41	5.75	3.82	0.4008
LLM Fusion									
V2V-LLM [4]	2.90	4.91	6.98	4.93	0.75	2.87	4.93	2.85	0.4068
V2V-GoT (Ours)	**1.65**	**2.63**	**3.59**	**2.62**	**0.12**	**1.92**	**3.45**	**1.83**	0.4068



(a) Our proposed graph-of-thoughts. (b) Simplified perception graph. (c) Simplified prediction graph.

Figure 4. Different graph-of-thoughts structures for cooperative autonomous driving. The QA types include Perception (Q1 - Q4), Prediction (Q5 - Q7), and Planning (Q8 - Q9). If two nodes are connected by a directed edge, the answer of the parent node QA is used as the input context of the child node QA.

Table 2. Testing performance of V2V-GoT in all question-answering tasks of V2V-GoT-QA dataset, in comparison with different inference graphs. F1: F1 score. L2: L2 distance error. L1: L1 distance error. CR: Collision rate. Comm: Communication cost. In each column, the **best** results are in boldface.

Method	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9		Comm(MB) ↓
	F1 ↑	F1 ↑	F1 ↑	F1 ↑	L2 (m) ↓	Accuracy	L2 (m) ↓	L1 ↓	L2 (m) ↓	CR (%) ↓	
Simplified Perception	-	-	-	58.0	8.18	82.0	7.66	0.0958	2.84	1.89	0.4068
Simplified Prediction	52.5	30.1	44.0	**60.8**	**8.05**	-	-	0.0900	3.24	2.17	0.4068
V2V-GoT (Ours)	52.5	30.1	44.0	**60.8**	**8.05**	**87.4**	**7.61**	**0.0876**	**2.62**	**1.83**	0.4068

need to be transferred from a CAV to the MLLM once. The MLLM can save and reuse the perception features received in the current timestep for multiple QAs of the graph-of-thoughts in the current and the next timesteps. The texts of the intermediate questions and answers do not need to be transferred between CAVs and the MLLM unless requested. The MLLM can perform the inference with the graph-of-thoughts reasoning and send the final planning answer to the CAVs in the end. Therefore, the overall communication cost of V2V-GoT is the same as prior work V2V-LLM [4].

7. Qualitative Results

Figures 5, 7, and 6 show the qualitative testing results of our proposed V2V-GoT in Q4. Overall Notable Objects (Figure 2d), Q7. Overall Prediction (Figure 2g), and Q9. Suggested Trajectory (Figure 2i), respectively. The input context information of each question comes from the model inference output of the associated parent question(s). In general, we can observe that our method generates perception, prediction, and planning output results close to ground-truth answers. More qualitative results can be found in the accompanying video.

8. Conclusion

In this work, we propose a novel graph-of-thoughts reasoning framework for MLLM-based cooperative autonomous driving. Our proposed graph-of-thoughts includes occlusion-aware perception and planning-aware pre-

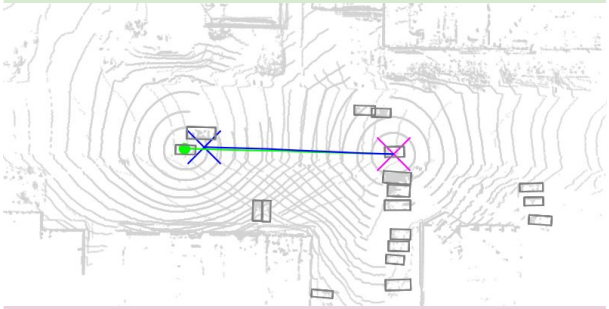
Q4: All Notable Objects

Q: I am CAV_EGO at (0.0,0.0). Is there anything I need to be aware of if my planned future trajectory is [(5.5,0.2),(10.9,0.3),(16.4,0.4),(21.9,0.6),(27.4,0.8),(33.0,0.9)]?
Context: There is no notable object visible to you. There is a car at (33.3,-1.8) invisible to you. There is a car at (39.1,-2.0) invisible to you.

V2V-GoT: There is a car at (33.3,-1.8). There is a car at (39.1,-2.0).
GT: There is a car at (35.4,-1.3). There is a car at (40.2,-1.7).

Figure 5. Qualitative testing sample result of V2V-GoT on Q4. Overall Notable Objects (Figure 2d). The context information is from the testing inference output of parent question Q3. Invisible Notable Objects (Figure 2c) and Q1. Visible Notable Objects (Figure 2a). Magenta ×: current location of the asking CAV. Magenta curve: reference trajectory in the question. Yellow ×: model output. Green ○: ground-truth answer.

Q9: Suggested Trajectory

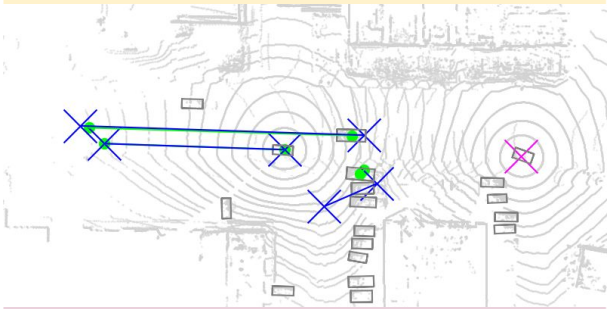


Q: I am CAV_1 at (-44.6,-1.0). What is the suggested future trajectory to avoid collision with nearby objects?
Context: What are the suggested speed and steering settings to avoid collision with nearby objects? The suggested speed setting is: moderate. The suggested steering setting is: straight.

V2V-GoT: The suggested future trajectory is [(-37.9,-0.8),(-31.3,-0.5),(-24.0,-0.2),(-17.8,0.1),(-10.8,0.3),(-3.9,0.5)].
GT: The suggested future trajectory is [(-36.4,-0.8),(-28.3,-0.6),(-20.2,-0.3),(-13.1,-0.2),(-5.8,0.0),(0.3,0.2)].

Figure 6. Qualitative testing sample result of V2V-GoT on Q9. Suggested Trajectory (Figure 2i). The context information is from the testing inference output of parent question Q8. Suggested Action Classification (Figure 2h). Magenta ×: current location of the asking CAV. Blue curve and ×: model output of the suggested future trajectory and the ending waypoint. Green curve and ○: ground-truth answer.

Q7: Overall Prediction



Q: I am CAV_1 at (-50.9,-1.4). Where might those notable objects move in the future?
Context: There is a car at (0.0,0.0) moving forward. The predicted future trajectory is [(50.5,1.9)]. There is a car at (-17.0,3.1) moving forward. The predicted future trajectory is [(43.9,5.1)]. There is a car at (-19.8,-7.3) moving forward. The predicted future trajectory is [(-8.5,-12.2)]. There is a car at (-49.2,-1.3) moving forward. The predicted future trajectory CAV_EGO is at (0.0,0.0) moving forward. Its planned future trajectory is [(38.8,1.3)]. CAV_EGO is a notable object.

V2V-GoT: There is a car at (0.0,0.0) moving forward. The predicted future trajectory is [(38.8,1.3)]. There is a car at (-17.0,3.1) moving forward. The predicted future trajectory is [(43.9,5.1)]. There is a car at (-19.8,-7.3) moving forward. The predicted future trajectory is [(-8.5,-12.2)].
GT: There is a car at (0.0,0.0) moving forward. The predicted future trajectory is [(38.8,1.3)]. There is a car at (-14.4,3.0) moving forward. The predicted future trajectory is [(42.1,4.7)]. There is a car at (-16.3,-5.2) moving forward. The predicted future trajectory is [(-17.1,-4.3)].

Figure 7. Qualitative testing sample result of V2V-GoT on Q7. Overall Prediction (Figure 2g). The context information is from the testing inference output of parent question Q5. Prediction by Perception (Figure 2e) and Q6. Prediction by Planning (Figure 2f). Magenta ×: current location of the asking CAV. Blue line and ×: model output of the predicted future trajectories, starting, and ending waypoints of notable objects. Green line and ○: ground-truth answer.

diction, which are designed to take advantage of V2V information sharing in cooperative driving scenarios and the multimodal understanding ability of MLLMs. To verify the effectiveness of our proposed ideas, we curate the V2V-GoT-QA dataset and develop the V2V-GoT model. Our experimental results show that our proposed method outperforms all other baseline methods. Moreover, the ablation study further indicates that our innovative designs in the graph-of-thoughts improve overall cooperative perception, prediction, and planning performance. To facilitate open-source research, we have publicly released our V2V-GoT-QA dataset and V2V-GoT code.

References

[1] Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *ACM/IEEE Symposium on Edge Computing (SEC)*, 2019. 2

[2] Hsu-kuang Chiu and Stephen F. Smith. Selective communication for cooperative perception in end-to-end autonomous driving. In *IEEE International Conference on Robotics and Automation (ICRA) Workshop*, 2023. 1

[3] Hsu-kuang Chiu, Chien-Yi Wang, Min-Hung Chen, and Stephen F. Smith. Probabilistic 3d multi-object cooperative tracking for autonomous driving via differentiable multi-sensor kalman filter. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 1, 2

[4] Hsu-kuang Chiu, Ryo Hachiuma, Chien-Yi Wang, Stephen F. Smith, Yu-Chiang Frank Wang, and Min-Hung Chen. V2v-llm: Vehicle-to-vehicle cooperative autonomous driving with multimodal large language models. *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. 2, 3, 5, 6, 7

[5] Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[6] Shiyu Fang, Jiaqi Liu, Mingyu Ding, Yiming Cui, Chen Lv, Peng Hang, and Jian Sun. Towards interactive and learnable cooperative driving automation: a large language model-driven decision-making framework. *IEEE Transactions on Vehicular Technology*, pages 1–12, 2025. 2, 3

[7] Xiangbo Gao, Yuheng Wu, Rujia Wang, Chenxi Liu, Yang Zhou, and Zhengzhong Tu. Langcoop: Collaborative driving with language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4235–4246, 2025. 2, 3

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 5

[9] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[10] Changxing Liu, Genjia Liu, Zijun Wang, Jinchang Yang, and Siheng Chen. Colmdriver: Llm-based negotiation benefits cooperative autonomous driving. *arXiv preprint arXiv:2503.08683*, 2025. 2, 3

[11] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 5

[12] Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. In *Advances in Neural Information Processing Systems (NeurIPS) Workshop (Foundation Models for Decision Making)*, 2023. 2

[13] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. In *Conference On Language Modeling (COLM)*, 2024. 2

[14] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision (ECCV)*, 2024. 2

[15] 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 *AAAI Conference on Artificial Intelligence (AAAI)*, 2024. 2

[16] 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. In *Europian Conference on Computer Vision (ECCV)*, 2024. 1, 2, 5

[17] Hao Su, Shin'ichi Arakawa, and Masayuki Murata. Cooperative 3d multi-object tracking for connected and automated vehicles with complementary data association. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 285–291, 2024. 2

[18] Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In *Conference on Robot Learning (CoRL)*, 2024. 1

[19] Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M. Alvarez. OmniDrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *CVPR*, 2025. 1

[20] Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, James Tu, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[21] Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. *arXiv preprint*, 2023. 2

[22] Hao Xiang, Runsheng Xu, and Jiaqi Ma. Hm-vit: Heteromodal vehicle-to-vehicle cooperative perception with vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[23] Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning (CoRL)*, 2022. 2, 5, 7

[24] Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European Conference on Computer Vision (ECCV)*, 2022. 5, 7

[25] Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 1, 2, 5, 7

[26] Runsheng Xu, Xin Xia, Jinlong Li, Hanzhao Li, Shuo Zhang, Zhengzhong Tu, Zonglin Meng, Hao Xiang, Xiaoyu Dong, Rui Song, Hongkai Yu, Bolei Zhou, and Jiaqi Ma. V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 6

[27] Zewei Zhou, Seth Z. Zhao, Tianhui Cai, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Turbotrain: Towards efficient and balanced multi-task learning for multi-agent perception and prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 1