# 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

In this paper, we propose a new graph-of-thoughts reasoning framework for MLLM-based cooperative autonomous driving, as illustrated in Figure 1. In this problem setting, all CAVs share their perception features with a MLLM. Any CAV can ask the MLLM to answer a driving-related question. The MLLM fuses the perception features from all CAVs and performs inference to provide the answer. To verify the effectiveness of our proposed graph-of-thoughts, we first curate the training and testing QA data samples of our proposed V2V-GoT-QA dataset, which is built on top of the V2V4Real [8] cooperative driving dataset. Our V2V-GoT-QA includes 9 types of different QA samples, as illustrated in Figure 2. More specifically, our newly proposed **occlusion-aware perception** questions (Q1 - Q4) consider visible, occluding, and invisible objects. Our newly 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 proposed 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.

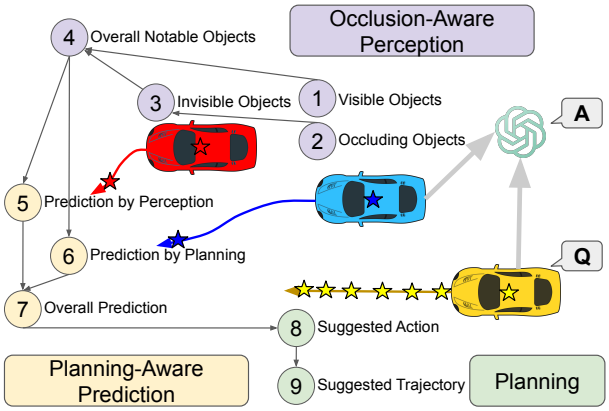
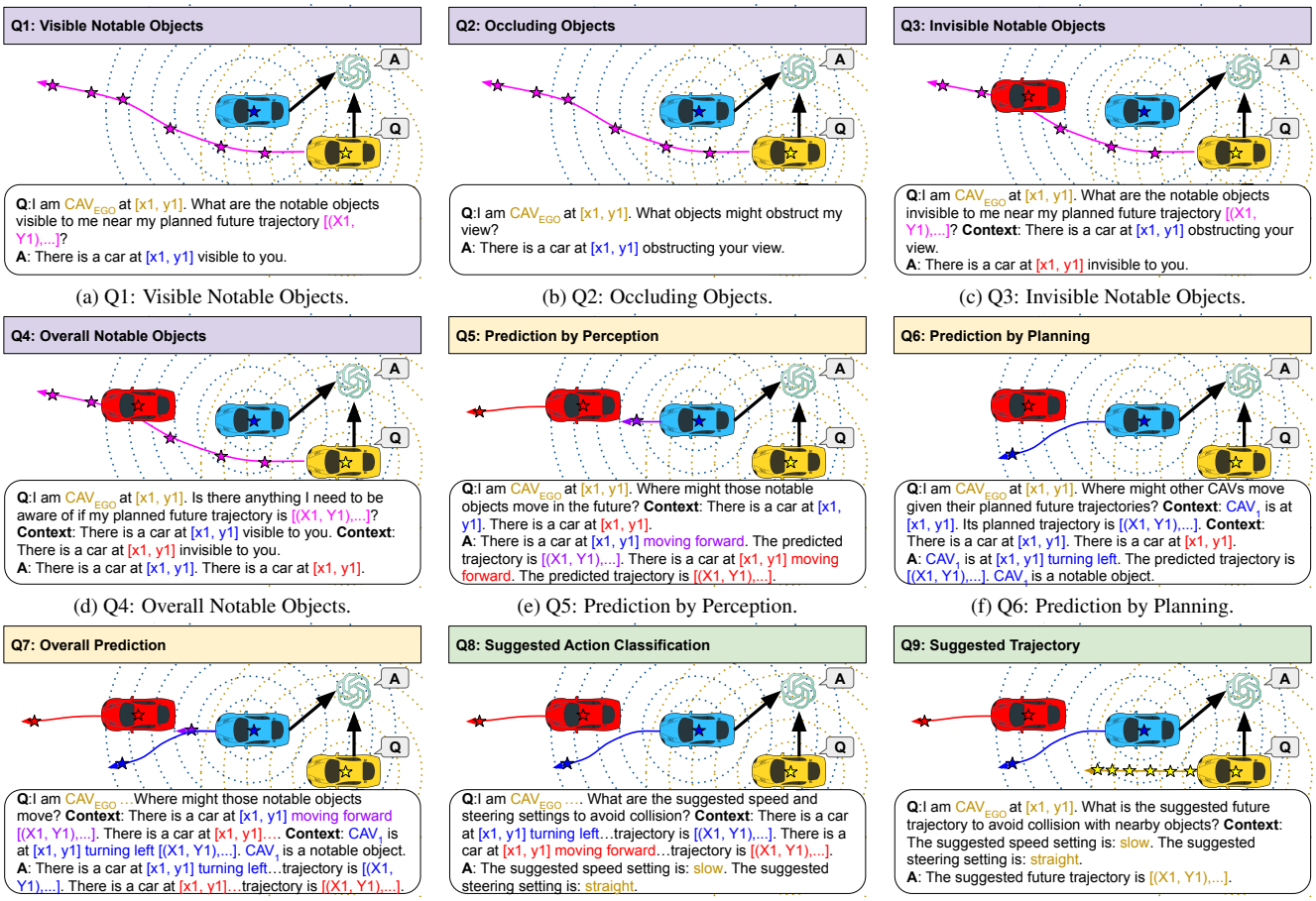



Figure 1. Illustration of our proposed graph-of-thoughts reasoning framework for cooperative autonomous driving.

In our experiments, we compare the final planning performance of V2V-GoT and baselines from prior work V2V-LLM [1], including *no fusion*, *early fusion*, and *intermediate fusion*. Our experimental results show that our proposed V2V-GoT outperforms all other baselines in the planning tasks, achieving the lowest collision rates and L2 errors between the ground-truth trajectories and model outputs.




(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.

# 2. V2V-GoT-QA Dataset

## 2.1. Dataset Details

Following V2V-LLM [1], we use V2V4Real [8] as the base dataset and curate various QA pairs. V2V4Real [8] has 7105 training frames and 1993 testing frames. For each frame that is 3 seconds before the end of each driving sequence in V2V4Real [8], 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.

## 2.2. Dataset Curation

We use the ground-truth bounding box annotations, each CAV and object's future trajectory, and their geometric relationships in V2V4Real [8] 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).

## 2.3. Perception QAs

The main goal of the perception QAs is to identify and locate 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 **Q1. Visible Notable Objects** (Figure 2a): locating notable objects that are visible to the ego CAV, **Q2. Occluding Objects** (Figure 2b): locating objects occluding the ego CAV's field-of-view, and **Q3. Invisible Notable Objects** (Figure 2c): locating notable objects that are invisible to the ego CAV. Then we merge the answers in **Q4. Overall Notable Objects** (Figure 2d) to generate the final perception result. 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.

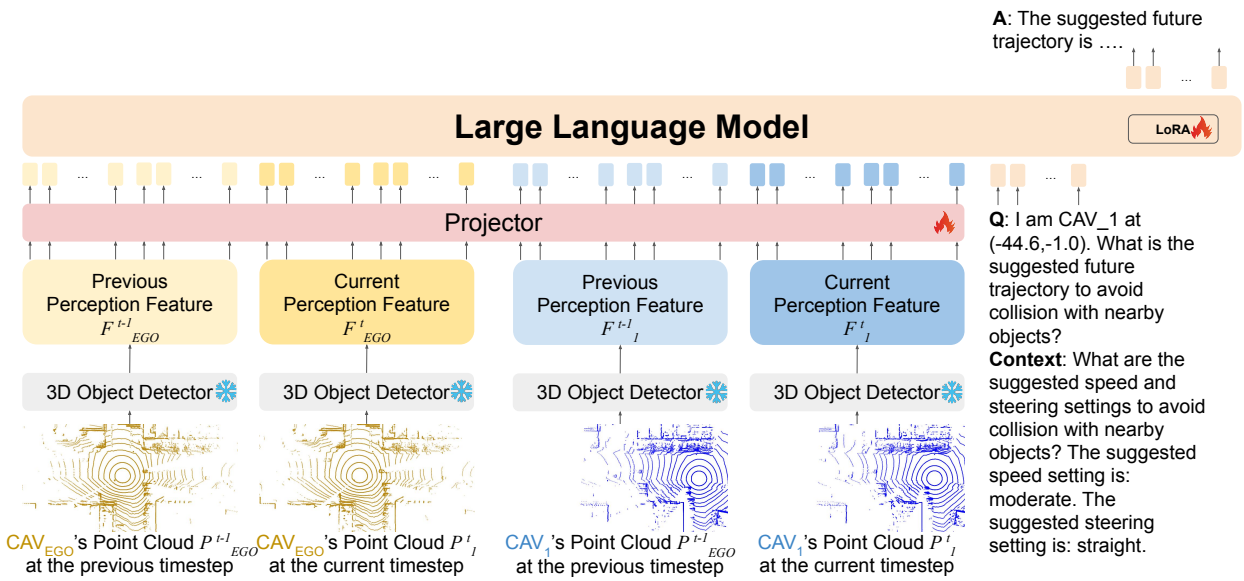



Figure 3. Model architecture of V2V-GoT.

### 2.4. Prediction QAs

We propose the idea of **planning-aware prediction** to the graph-of-thoughts reasoning. If a notable object is also a CAV, it can directly share its planned future trajectory with other CAVs. In our graph-of-thoughts, we divide the prediction task into two parts: **Q5. Prediction by Perception** (Figure 2e) and **Q6. Prediction by Planning** (Figure 2f). Then we merge their results in **Q7. Overall Prediction** (Figure 2g) to generate the final prediction output.

### 2.5. Planning QAs

We first classify the suggested action's speed and steering settings in **Q8. Suggested Action Classification** (Figure 2h), and then use the answer as the input context to generate the final suggested future trajectory in **Q9. Suggested Trajectory** (Figure 2i). For more details of each QA types in our proposed V2V-GoT-QA, please refer to our supplementary materials.

## 3. V2V-GoT Model

We build our V2V-GoT model, as shown in Figure 3. LLaVA [3] is used as the base MLLM architecture. We apply a LiDAR-based 3D object detector, PointPillars [2], to extract perception features from each individual CAV's point cloud. 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. During training, we only train the projector layers and the LoRA parts of the model and freeze the remaining parts.

## 4. Experiment

### 4.1. Baseline Methods

We adopt V2V-LLM [1] 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 In *early fusion*, the union of the point cloud of both CAVs is used to extract perception features. In *intermediate fusion*, cooperative 3D object detectors are used to extract perception features as MLLM's visual input.

### 4.2. 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.

We also perform ablation study to further verify the impacts of our newly proposed ideas of **occlusion-aware perception** and **planning-aware prediction**. We build the *simplified perception graph* by removing Q1, Q2, and Q3 from the proposed graph-of-thoughts. And we build the *simplified prediction graph*, by removing Q6 and Q7 from the proposed graph-of-thoughts. The ablation results are given in Table 2, showing that both **occlusion-aware perception** and **planning-aware prediction** are effective to improve the overall performance.

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 [1]. 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 [7] | 3.65 | 6.21 | 8.75 | 6.20 | 1.19 | 4.41 | 6.38 | 3.99 | 0.4008 |
| V2X-ViT [6] | 3.46 | 5.80 | 8.19 | 5.81 | 1.45 | 4.24 | 6.59 | 4.09 | 0.4008 |
| CoBEVT [5] | 3.38 | 5.42 | 7.46 | 5.42 | 1.31 | 4.41 | 5.75 | 3.82 | 0.4008 |
| *LLM Fusion* | | | | | | | | | |
| V2V-LLM [1] | 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 |

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 |

### 4.3. Qualitative Results

Figure 4 shows the qualitative testing results of our proposed V2V-GoT in Q9. Suggested Trajectory (Figure 2i). The input context information comes from the model inference output of the associated parent question Q8. Suggested Action Classification (Figure 2h). We can observe that our method generates a planning output close to ground-truth answer. More qualitative results in the perception and prediction stages can be found in the supplementary material.

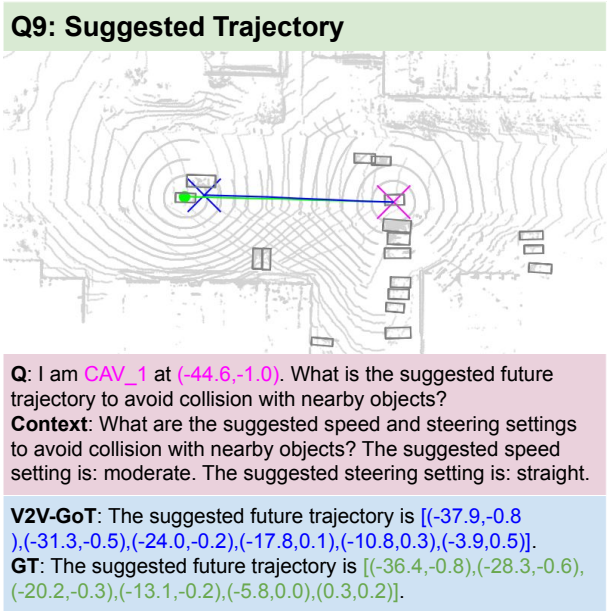



Figure 4. Qualitative testing sample result of V2V-GoT on Q9. Suggested Trajectory (Figure 2i). The context information is from the testing inference output of parent question Q8. Suggested Action Classification (Figure 2h). Magenta ×: current location of the asking CAV. Blue curve and ×: model output of the suggested future trajectory and the ending waypoint. Green curve and ○: ground-truth answer.

## 5. 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 prediction, 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] 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. 1, 2, 3, 4

[2] 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. 3

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[4] 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

[5] 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. 4

[6] 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. 4

[7] 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. 4

[8] 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. 1, 2

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

Supplementary Material

## 6. Details of V2V-GoT-QA

**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' 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.

**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.

**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 [4]. 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 exam-

ple, 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.

# 7. Ablation Study

## 7.1. 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, we perform an ablation study on two additional graph structures: a *simplified perception graph* and a *simplified prediction graph* The ablation results are given in Table 2.

### 7.1.1. Simplified Perception Graph

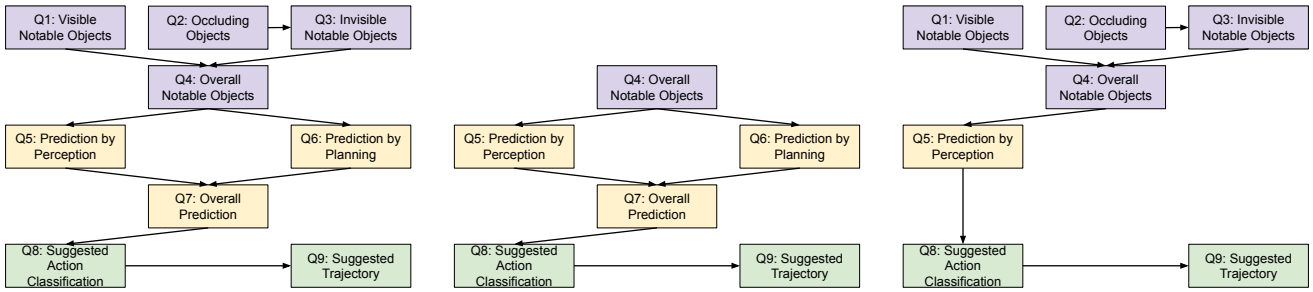
To verify the impact of **occlusion-aware perception**, we create the *simplified perception graph* reasoning procedure, where we directly ask the model to locate all notable objects without considering Q1, Q2, or Q3 first. We create additional training data samples of Q4. Overall Notable Objects (Figure 2d) but without context from the answer of Q2 and Q3. Then we train the model with newly generated data samples and the existing training dataset, and perform testing inference directly starting from Q4. Overall Notable Objects (Figure 2d). 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.

### 7.1.2. Simplified Prediction Graph

To verify the impact of **planning-aware prediction**, we create the *simplified prediction graph* reasoning procedure, where we exclude Q6. Prediction by Planning (Figure 2f) and Q7. Overall Prediction (Figure 2g) from the graph-of-thoughts. We directly use the answer of Q5. Prediction by Perception (Figure 2e) and the final prediction result during the inference. 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.

# 8. More Qualitative Results

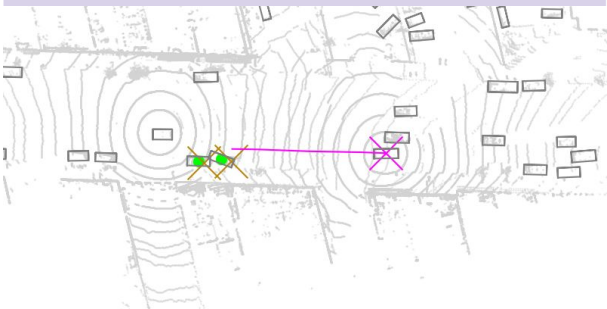
Figures 6 and 7 show the qualitative testing results of our proposed V2V-GoT in Q4. Overall Perception (Figure 2d) and Q7. Overall Prediction (Figure 2g), respectively. The input context information comes from the model inference output of the associated parent questions. We can observe that our method generates perception and prediction output close to ground-truth answers.




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

Figure 5. 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.

**Q4: All Notable Objects**

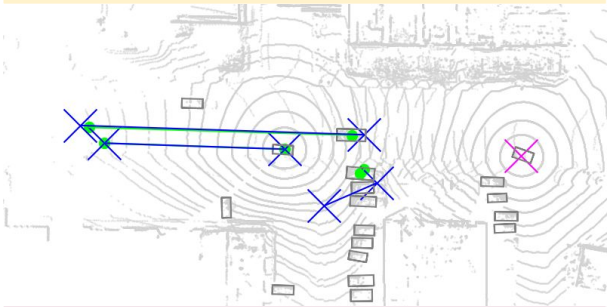


**Q**: I am CAV_EGO at (0.0,0.0). Is there anything I need to be aware of if my planned future trajectory is [(5.5,0.2),(10.9,0.3),(16.4,0.4),(21.9,0.6),(27.4,0.8),(33.0,0.9)]?
**Context**: There is no notable object visible to you. There is a car at (33.3,-1.8) invisible to you. There is a car at (39.1,-2.0) 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 6. 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.

**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.