Supplementary Material:
Distilling Multi-modal Large Language Models for Autonomous Driving

A. Overview

We present the supplementary material for the paper "Distilling Multi-modal Large Language Models for Autonomous Driving". In Section B, we provide details on the surrogate tasks module of DiMA . Section C has the details on the training setup and the procedure for generating additional text annotations for the nuScenes dataset. In Section D, we present additional qualitative results of DiMA's vision-based planning branch as well as visual question-answering results from the MLLM branch. We provide an additional quantitative evaluation of DiMA in Section E. In Section F, we provide an additional set of ablation experiments to examine the role of each surrogate task in planning performance.

B. Surrogate tasks overview

An important component of training the MLLM branch of DiMA is the surrogate tasks module. In addition to being trained for planning and visual question answering, the MLLM is trained to perform the following surrogate tasks: masked reconstruction, future prediction, and scene editing. We design these tasks to enrich the **b**ird's-eye-view, **e**go, **a**gent, and **m**ap ($BEAM$) scene representations. Surrogate tasks module takes hidden token embedding of penultimate layer of the LLM model. An illustration of the module can be seen in Figure 1. Each decoder head in surrogate module is consists of 3 Linear layers with a ReLU activation layer. Below, we provide some additional details on the scene editing tasks well as overviews on the other two tasks.

Masked reconstruction. Inspired by masked auto encoders [4], we formulate a masked reconstruction task of BEV token embeddings, to enrich the visual features learned by the scene encoder. The MLLM trained to infer masked regions based on the context provided by the visible BEV tokens and the rest of the multi-modal input. A reconstruction head takes the latent representations from the penultimate layer of LLM and predicts reconstructed BEV token embeddings $\hat{B}$. This decoder head is supervised using $\mathcal{L}_{recon}$ (refer equation 1 main paper).

Future prediction. We formulate future prediction of BEV token embeddings, combined with VQAs to ground the scene encoder and MLLM model spatio-temporally, and learn robust spatio-temporal $BEAM$ scene representations. These spatio-temporal $BEAM$ cues benefits planning performance (Table 4 of main paper). A future prediction head takes the latent representations from the penultimate layer of LLM and predicts future BEV token embedding at times $t+1$, $t+2$. We supervise these future BEV token embeddings using $\mathcal{L}_{future}$ (refer equation 2 main paper). Motivated by the planning objective of vehicle trajectory prediction over a 3-second horizon, we predict BEV tokens 3 seconds into the future. To achieve this 3-second prediction, the input BEV tokens are duplicated and concatenated; the first half of the future prediction head's output is then interpreted as the prediction for $(t+1)$ and the latter for $(t+2)$.

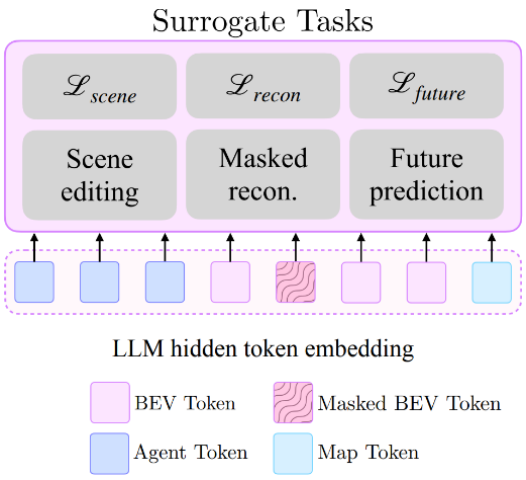


Figure 1. Overview of Surrogate tasks. Here hidden token embeddings are latent representations from the penultimate layer of LLM. These hidden token embeddings corresponding to **b**ird's-eye-view, **e**go, **a**gent, and **m**ap ($BEAM$) token embeddings are used as input as surrogate task decoder heads to perform masked reconstruction, future prediction and scene editing.

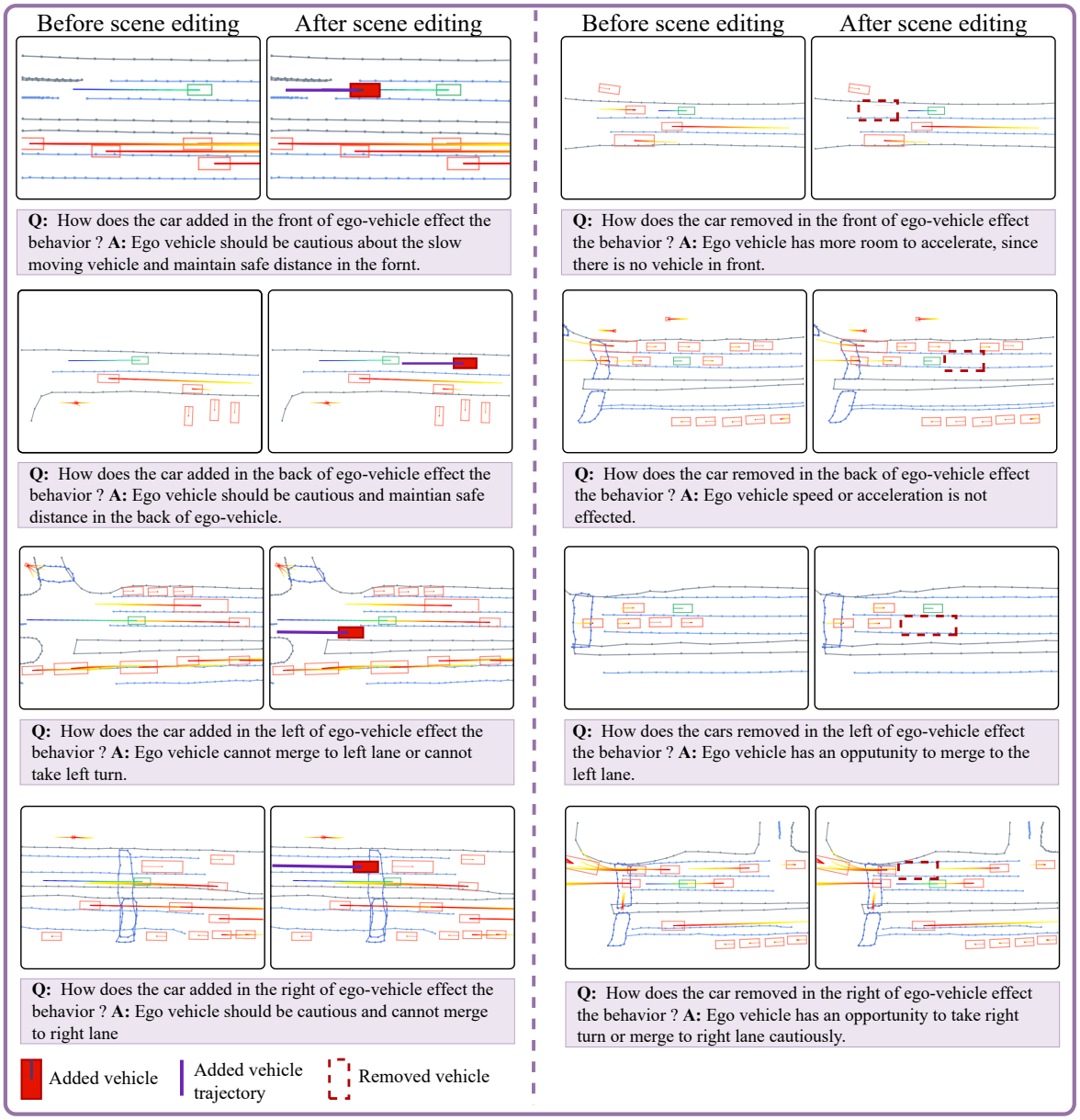


Figure 2. Examples of addition and deletion in scene editing. In the left column, a car (solid red box) is added in the premises of the ego-vehicle (green box). In the right column, a car (dashed red box) is removed from the premises of the ego-vehicle. A corresponding question-answer pair is created to characterize the edit.

Scene editing. We propose a novel scene editing task in which we augment scenes by removing or adding new agents. Along with this, we construct a question-answer pair related to the edit. We show examples of this in Figure 2. For scene addition, given the map constraints, predicted map, the ego bounding box location and the trajectories of predicted agents obtained from perception, and prediction tasks of scene encoding (refer [6, 7], using camera meta data we project all these to two-dimensional space as shown in 2. Using this, we identify possible locations in the premises of ego-vehicle location, where a new object can be added and randomly choose the location for the new agent, which is of size maximum $2\times$ of size of ego-vehicle. Given the location for the new agent and map constraints, we create a way-point trajectory for a new agent of category "car" or "truck". A new agent token embedding is then created using a linear layer. This new agent token embedding, the corresponding text prompt, and the rest of the $BEAM$ token embeddings are passed as input to the LLM. The

hidden latent LLM features corresponding to agent token embeddings are then fed into a dedicated scene editing decoder head that performs way-point prediction of the ego vehicle. The output way-point prediction of the ego vehicle from scene editing head is supervised using $\mathcal{L}_{scene}$. $\mathcal{L}_{scene}$ is ego-agent collision constraint loss with updated agents incorporating either new added agent or removed agent in scene editing task. For ego-agent collision constraint loss refer [7]. The language prediction head performs question-answering on the new QA pair. This task thus contributes to the existing planning constraint loss and VQA loss of the MLLM. The QA pairs are generated according to a template in which the possible movements of the ego vehicle are described.

C. Experimental Setup

C.1. Training details

We train DiMA-VAD-tiny, DiMA-VAD-Base, and DiMA-UniAD, using training setups adapted from [7] and [6]. Our training process follows a two-stage approach. First, we pre-train the vision planner only under the perception, prediction, and planning constraints for 60 epochs in order to learn informative latent scene representations. Second, we perform joint training of the vision planner and the MLLM for an additional 30 epochs, incorporating all proposed tasks and losses detailed in Section B. In the second stage, the language model of the MLLM is fine-tuned using LoRA [5]. Question-answer pairs from the augmented DriveLM dataset [8] are input along with the multi-view image sequence in the second stage. For each input sample, we randomly select one QA-pair from each category in every iteration to send as text prompt input to the network. This ensures diversity in questions while avoiding redundant visual inputs. For both stages, we employ the AdamW optimizer with a cosine annealing scheduler, a weight decay of 0.01, and an learning rate of 2×10^{-4}. In all our experiments we set the random masking ratio as $[0.2, 0.4]$.

Table 1. Comparison of L2 trajectory error and collision rate on nuScenes [2] using standardized evaluation [10]. Models are evaluated on the general validation split as well as a "targeted" split of challenging samples from [10]. The performance of the DiMA model variants are in shades of purple. We summarize results by averaging over at $t = \{1, 2, 3\}s$ as well as at all time steps.

Method	Using Ego status	Traj L2 (m) ↓					Collision (%) ↓
		1s	2s	3s	$\text{Ave}_{1,2,3s}$	Ave_{all}	Ave_{all}
Full validation split							
UniAD[6]	✗	0.48	0.89	1.47	0.95	0.83	0.40
PARA-Drive[10]	✗	0.26	0.59	1.12	0.66	0.56	0.17
TOKEN[9]	✗	0.26	0.70	1.46	0.81	0.68	0.15
DiMA (UniAD)	✗	0.19	0.50	1.08	0.59	0.50	0.06
Targeted validation split							
UniAD[6]	✗	0.47	1.09	1.92	1.16	0.99	0.15
PARA-Drive[10]	✗	0.38	0.97	1.88	1.08	0.91	0.14
DiMA (UniAD)	✗	0.30	0.82	1.63	0.92	0.77	0.06
3-point turn (zero-shot)							
UniAD[6]	✗	0.68	1.55	2.90	1.71	1.43	0.00
PARA-Drive [10]	✗	0.50	1.38	2.76	1.55	1.29	5.33
TOKEN [9]	✗	0.39	1.29	2.60	1.43	1.18	4.00
DiMA (UniAD)	✗	0.28	0.94	2.16	1.13	0.90	0.00
Resume from stop							
UniAD[6]	✗	1.09	1.66	3.06	1.94	1.73	0.00
PARA-Drive	✗	0.14	0.79	2.30	1.08	0.85	0.00
TOKEN	✗	0.13	0.70	1.58	0.80	0.65	0.00
DiMA (UniAD)	✗	0.38	0.83	1.49	0.90	0.84	0.00
Overtake							
UniAD[6]	✗	0.60	1.39	2.38	1.45	1.27	0.98
PARA-Drive	✗	0.27	0.89	1.94	1.03	0.85	2.30
TOKEN	✗	0.29	0.77	1.63	0.90	0.74	0.00
DiMA (UniAD)	✗	0.28	0.75	1.55	0.86	0.78	0.41

C.2. Generation of text annotations

We augment the existing text annotations of the Drive-LM [8] by generating question-answer (QA) pairs for samples in the nuScenes dataset [2]. First, we parse the numerical annotations of each object in the scene, such as the ego-vehicle and the surrounding objects. We denote each object as an agent and assign attributes such as the camera in which it is visible, the name of the object, and the vehicle speed. Additionally, we use rule-based algorithms to assign brief text descriptions of the future movement, the direction of movement relative to the ego-vehicle, the future speed, the type of interaction with the ego vehicle, and the probability of collision with the ego-vehicle. Using this annotation along with a few in-context examples, we prompt a Llama 3-70B model [3] to generate 5 Drive-LM-like QA pairs for each category of question. The input system prompt can be seen in Figure 5. An example of a textual description of the numerical annotations can be seen in Figure 6. Examples of generated QA pairs can be seen in Figure 7.

D. Additional Qualitative Results

We present extensive qualitative results of DiMA . In Section D.1, we present a comparison of planning performance of the vision-based planner on nuScenes. In Section D.2, we provide numerous visual question-answering results on various subsets

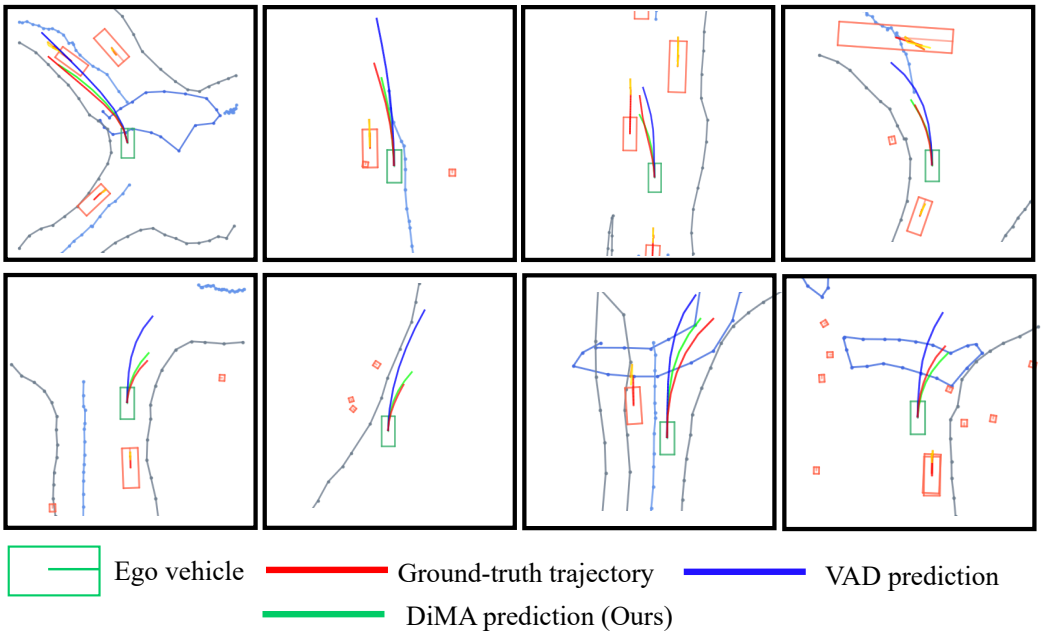


Figure 4. Visual comparison of the planning performance of DiMA (VAD-Tiny) with VAD-Tiny [7]. Samples are from the "targeted" subset of the nuScenes validation split.

of the nuScenes dataset.

D.1. nuScenes planning

We present qualitative planning results of DiMA compared to that of VAD in Figure 4. We evaluate on difficult "targeted" samples from nuScenes. These are samples where the ego-vehicle is performing right and left turns. As seen in the figure, training with DiMA ensures safe trajectory prediction, avoiding collisions with vehicles when turning around corners (see row 1, columns 1 and 4) as well as avoiding lane departures (row 1 column 2). DiMA also results in more precise turns (see row 2).

In Fig. 1 of the main paper, we show a 3-point turn emphasizing complex maneuver, involving a sharp left turn, backward movement, and a second left turn, typically takes ∼1 minute to complete. In this supplemental material, we include a complete turn playable video visualization in Fig. 3 above.

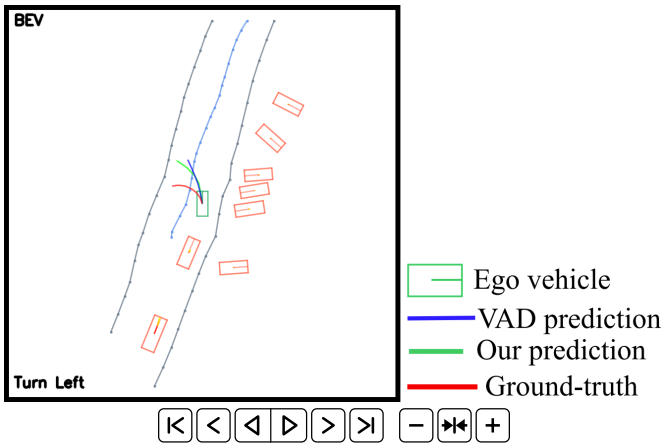


Figure 3. Visualization of three-point turn results. Press center buttons to play with Adobe Reader.

D.2. Visual question-answering

We present numerous qualitative examples of planning and VQA performance of the DiMA MLLM branch. For Drive-LM test samples that have ground-truth annotations, we compare the generated text response with the ground-truth answer in Figure 8. We also plot the predicted future trajectory. In Figure 9 we show two scenarios in which DiMA provides incorrect VQA results.

For a more extensive qualitative analysis, we compare the performance of DiMA -MLLM with GPT-4 [1] in Figures 10, 11, and 12. We present common reasoning questions along with the DiMA -MLLM response and the GPT-4 response. The

Table 2. $Avg_{all}(\downarrow)$ L2 trajectory error on nuScenes dataset using standardized evaluation ParaDrive

Method	Surrogate			targeted	long-tail		
	Masked recon.	Future pred.	Scene editing		3point	overtake	resume
VAD-Tiny	✗	✗	✗	1.37	1.83	1.75	1.32
DiMA (VAD-Tiny)	✓	✓	✓	0.97	1.43	1.23	0.83
	✓	✗	✗	1.22	1.67	1.51	1.07
	✗	✓	✗	1.16	1.56	1.43	0.98
	✗	✗	✓	1.09	1.51	1.39	0.96

input to GPT-4 is the text prompt and a stitched image of the multi-view image set. We also show the planning performance plotted in the image as well as in a diagrammatic form on the right side of each row. As observed in these examples, DiMA is able to focus on objects important for navigation and planning. As can be seen in Figure 10 row 4, our model correctly predicts the future right turn to be taken by the ego-vehicle, while GPT-4 suggests the ego-vehicle should move straight. A similar problem is observed in Figure 12 row 4, where the prediction by GPT-4 is much more vague than that of DiMA .

E. Additional Quantitative Results

We present the performance of DiMA-UniAD performance evaluated on the nuScenes dataset using standardized evaluation [10] in Table 1. We compare the performance of both DiMA-UniAD and UniAD[6] on the general validation split as well as a "targeted" split of challenging samples from [10] and on long-tail scenarios. We observe consistent improvement across all metrics, resulting in significantly reduced L2 trajectory error and collision rate. This model version also out-performs state-of-the-art methods PARA-Drive [10] and TOKEN [9] in almost all cases.

F. Additional ablation study

We include additional ablation studies to analyze the effect of surrogate tasks across targeted and long-tail scenarios in nuScenes in Tab. 2 of this supplementary material. Our experiments demonstrate that closer the objective of the surrogate task is to the planning task, the more it boosts performance. For example, masked reconstruction enhances the visual representation. We observe a stronger boost in planning by training for future prediction, which encourages temporal consistency. The task of scene editing is most beneficial, as it encodes agent behaviors and reasoning patterns, further improving planning performance.

QA pair generation with Llama-3-70B | System prompt

```
You are a system designed to generate high quality question answer pairs in the
scenario of autonomous driving, from the point of view of an ego-vehicle
viewing a 360 degree scene around you. Questions-answer pairs may be of three
types : Perception, Prediction, and Planning. Perception questions relate to
the nature of the agents/objects around the ego-vehicle. Planning questions
relate to questions about the future actions of the ego-vehicle. Prediction
questions are detailed questions about the  agents/objects around the ego
vehicle. Here are some examples of each type of question.

Perception
 1. Q: 'What are objects to the front left of the ego car?'
    A: 'There is one truck and one car to the front left of the ego car.'
 2. Q: 'Are there moving pedestrians to the front right of the ego car?'
    A: 'Yes.'

Prediction
 1. Q: 'Is <c1,CAM_FRONT> a traffic sign or a road barrier?'
    A: 'No'
 2. Q: 'What object should the ego vehicle notice first when the ego
    vehicle is getting to the next possible location? .... [TRUNCATED]
    A: 'Firstly, notice <c6,CAM_FRONT,1074.8,336.5>. It is a traffic sign, so
       the ego vehicle should stop. Secondly, notice
       <c1,CAM_FRONT,1413.3,534.2>. It is stationary, so .... [TRUNCATED]

Planning
 1. Q: 'What is the probability of colliding with <c1,CAM_FRONT> after the ego
    vehicle steps on the brakes?'
    A: 'Low'
 2. Q: 'In this scenario, what are safe actions to take for the ego vehicle?'
    A: 'Brake gently to a stop, slightly offset to the right.'

For the given scene, generate five of each type of questions based on
the attributes provided of the ego vehicle and each agent surrounding the ego
vehicle. Make sure that the answer is in the format

    {
        'Perception':
            {'Q':'Question','A':'Answer'},
        'Prediction':
            {'Q':'Question','A':'Answer'},
        'Planning':
            {'Q':'Question','A':'Answer'}
    }

Do not return any other text than the QA pairs in a correct python dictionary
format. Make sure the answers are as descriptive as possible. Avoid one word
answers or yes/no questions.
```

Figure 5. The system prompt given to Llama-3 to generate question-answer pairs.

QA pair generation with Llama-3-70B | Sample numerical annotation


```
"token": "f9878012c3f6412184c294c13ba4bac3",
"scene_description": "Car overtaking, parking lot, pedestrians, pedestrian
exiting car, objects on the ground",
"agent_attributes": {
    "c0": {
        "category": "truck",
        "speed": -0.00784316331860773,
        "assigned_cameras": [
            "CAM_BACK_LEFT"
        ],
        "future movement": "stopped",
        "future speed": "not moving",
        "direction": "towards from ego vehicle",
        "interaction_with_ego_vehicle_type": "none",
        "probability_of_collision_with_ego_vehicle": "low"
    },
    "c1": {
        "category": "pedestrian",
        "speed": 0.0,
        "assigned_cameras": [
            "CAM_FRONT",
            "CAM_FRONT_LEFT"
        ],
        "future movement": "stopped",
        "future speed": "not moving",
        "direction": "away from ego vehicle",
        "interaction_with_ego_vehicle_type": "none",
        "probability_of_collision_with_ego_vehicle": "low"
    },

    ... [TRUNCATED] ...

    "c24": {
        "category": "car",
        "speed": 0.0,
        "assigned_cameras": [
            "CAM_BACK"
        ],
        "future movement": "stopped",
        "future speed": "not moving",
        "direction": "away from ego vehicle",
        "interaction_with_ego_vehicle_type": "none",
        "probability_of_collision_with_ego_vehicle": "low"
    }
```


Figure 6. An example of the text-description of the numerical annotations of a scene from nuScenes. This JSON file is created using rule-based algorithms. This is appended to the system prompt.

QA pair generation with Llama-3-70B | Generated QA pairs

Perception

```
{
    "Q": "What are objects to the back right of the ego car?",
    "A": "There is one car and one bicycle to the back right of the ego
         car."
},
{
    "Q": "Are there moving agents to the back right of the ego car?",
    "A": "Yes, there is one moving car to the back right of the ego car."
},
... [TRUNCATED] ...
```

Prediction

```
{
    "Q": "What are the future movements of the agents to the back right
         of the ego car?",
    "A": "The car will slightly steer to the right and the bicycle will
         remain stopped."
},
{
    "Q": "What are the future speeds of the agents to the back right of
         the ego car?",
    "A": "The car will be driving fast and the bicycle will not be
         moving."
},
... [TRUNCATED] ...
```

Planning

```
{
    "Q": "What is the probability of colliding with the car after the
         ego vehicle steps on the brakes?",
    "A": "Medium"
},
{
    "Q": "What actions taken by the ego vehicle can lead to a collision
         with the bicycle?",
    "A": "No action taken by the ego vehicle will lead to a collision
         with the bicycle."
},
... [TRUNCATED] ...
```

Behavior

```
    "Q": "Predict the behavior of the ego vehicle.",
    "A": "The ego vehicle is slighlty steering to the right. The ego
         vehicle is moving at a moderate speed."
```

Figure 7. Some examples of generated QA pairs. The perception, prediction, and planning pairs are generated with Llama-3. The behavior QA is created using the future motion of the ego vehicle.

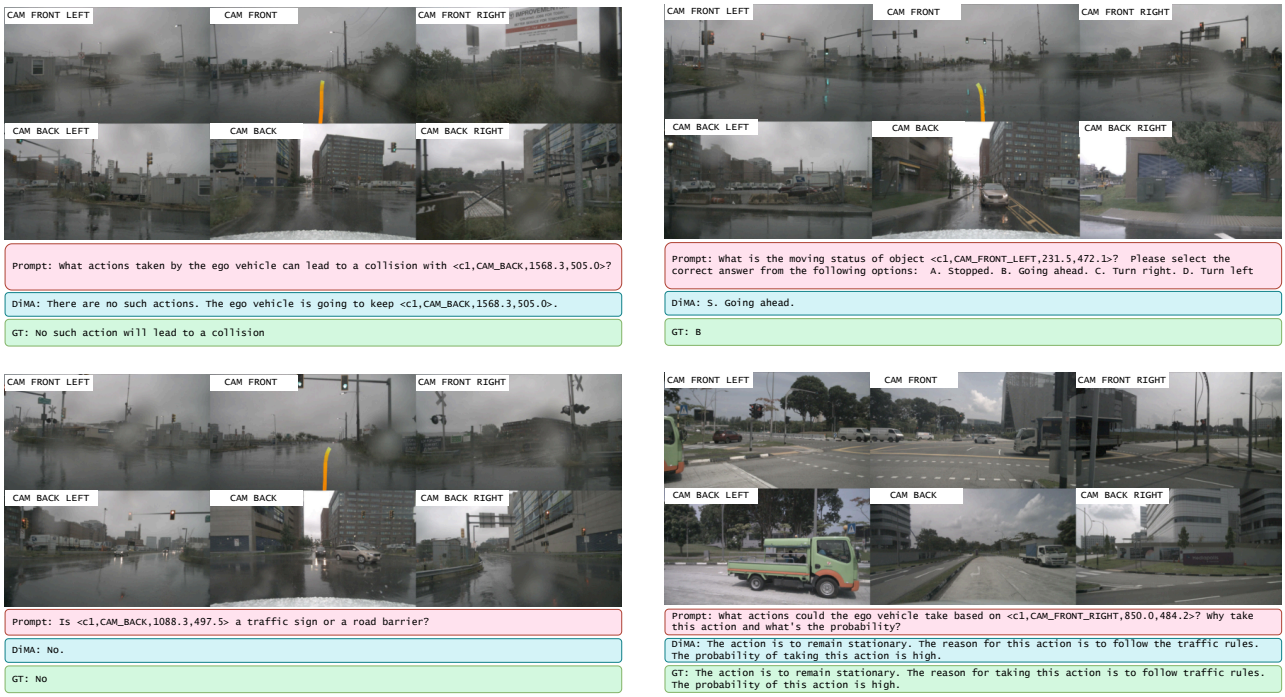


Figure 8. Visualization of VQA and planning prediction by the MLLM branch of DiMA-VAD-Tiny. We plot the predicted trajectory (orange-yellow) and show an example response of the language model branch to a question from the DriveLM test dataset [8].

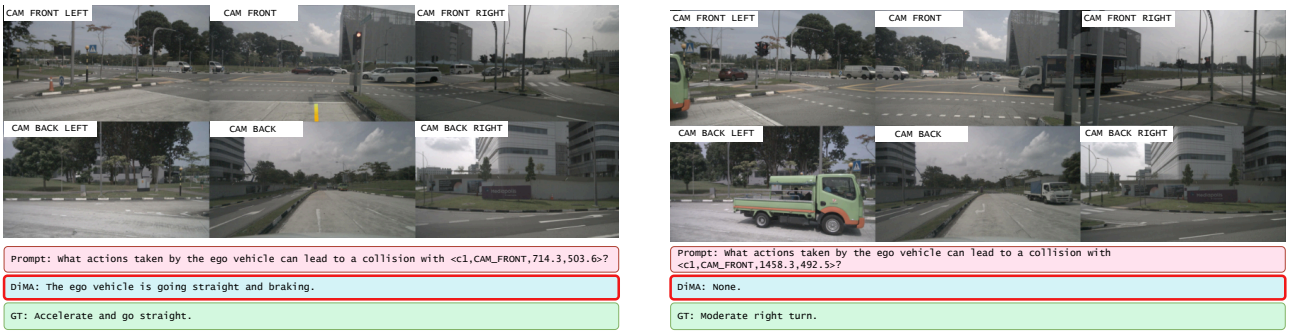
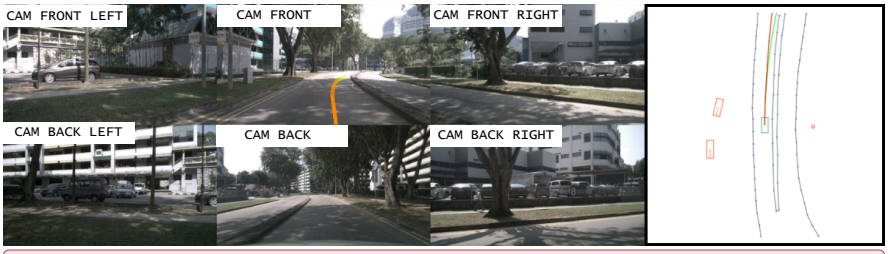


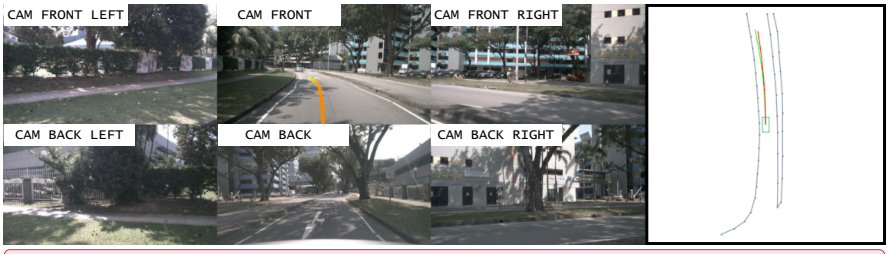
Figure 9. Visualization of failure cases of DiMA-VAD-Tiny. We plot the predicted trajectory (orange-yellow) and show an example response of the language model branch to a question from the DriveLM test dataset [8].



Prompt: What are the important objects in the scene?

DiMA: There are two cars and one pedestrian in the scene. The pedestrian is standing. The cars are parked.

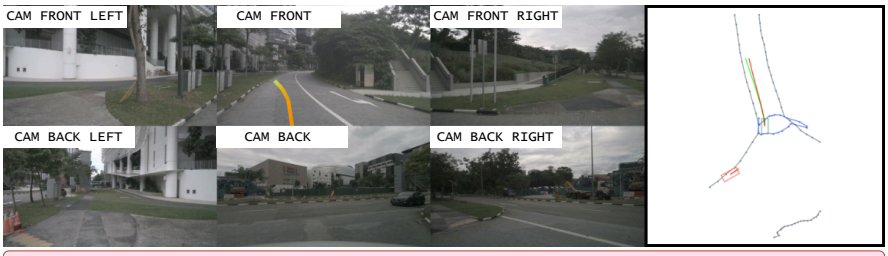
GPT4: The important objects in the scene include a multi-story building, parked cars, grassy areas, and surrounding trees.



Prompt: What are the important objects in the scene?

DiMA: There are many traffic cones in the scene. The road is relatively empty.

GPT4: The important objects in the scene include a roadway, trees, a parking area, a food court signage, and parts of nearby buildings.



Prompt: Predict the behavior of the ego vehicle.

DiMA: The ego vehicle is slightly steering to the left. The ego vehicle is driving with normal speed.

GPT4: The ego-vehicle is likely moving forward along the road while navigating around obstacles, with potential changes in direction due to the street layout.

Prompt: Predict the behavior of the ego vehicle.

DiMA: The ego vehicle is slightly steering to the right. The ego vehicle is driving fast.

GPT4: The ego-vehicle is likely moving straight along the road while encountering light rain and moderate visibility conditions.

Figure 10. Visualization of visual question-answering on the targeted subset of the nuScenes dataset. On the image, we plot the predicted trajectory (orange-yellow) The red line is the ground-truth trajectory. In the diagram, the green line is the predicted trajectory.

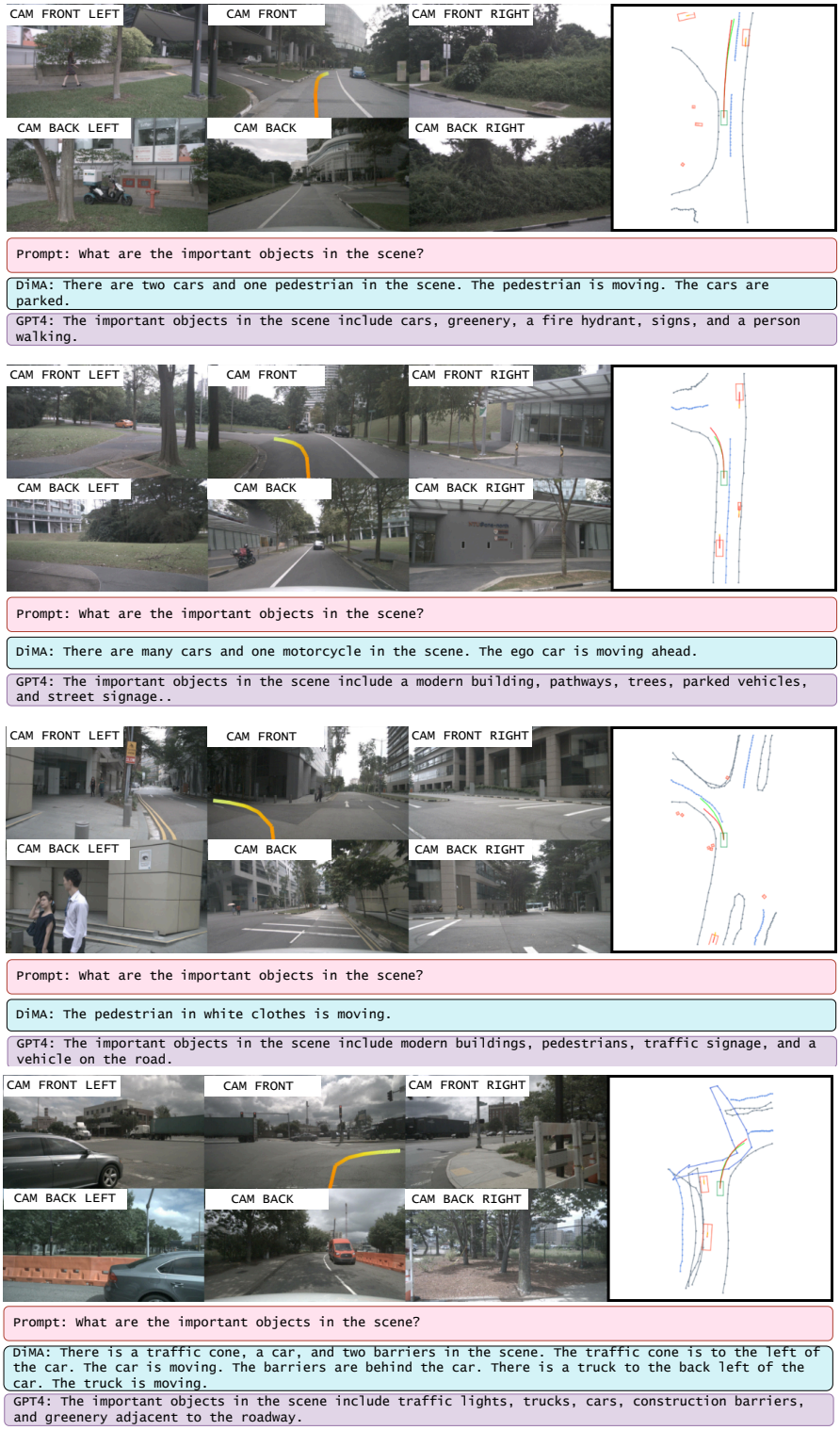
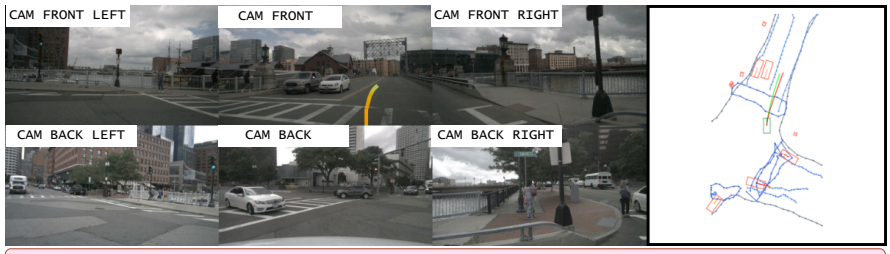


Figure 11. More visualization of visual question-answering on the targeted subset of the nuScenes dataset. On the image, we plot the predicted trajectory (orange-yellow) The red line is the ground-truth trajectory. In the diagram, the green line is the predicted trajectory.

Prompt: What are the important objects in the scene?

DiMA: There are two cars, one truck, and one pedestrian in the scene.

GPT4: The important objects in the scene include vehicles, traffic signals, road markings, and the Raymond L. Flynn Marine Park sign.



Prompt: What are the important objects in the scene?

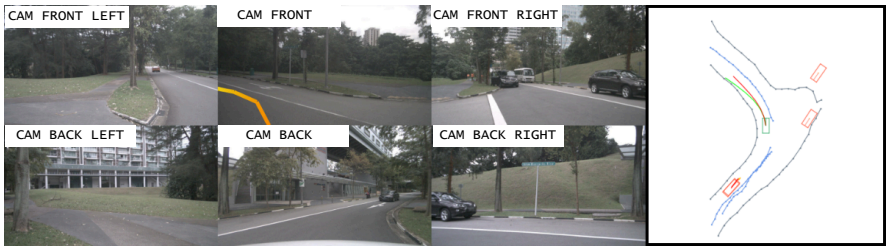
DiMA: There are two pedestrians moving.

GPT4: The important objects in the scene include cars, street signs, buildings, a river, and pedestrians.

Prompt: What are the important objects in the scene?

DiMA: There are many cars, two pedestrians, and one truck in the scene. The cars are moving. The pedestrians are standing. The truck is moving. The traffic light is green.

GPT4: The important objects in the scene include vehicles, traffic lights, road signs, and architectural structures, set in a wet urban environment.



Prompt: In this scenario, what are safe actions to take for the ego vehicle?

DiMA: Turn left.

GPT4: The ego vehicle should proceed cautiously, ensuring to yield to any approaching traffic while navigating the area.

Figure 12. More visualization of visual question-answering on the targeted subset of the nuScenes dataset. On the image, we plot the predicted trajectory (orange-yellow) The red line is the ground-truth trajectory. In the diagram, the green line is the predicted trajectory.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3

[3] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 3

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 1

[5] 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. *arXiv preprint arXiv:2106.09685*, 2021. 3

[6] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 2, 3, 5

[7] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023. 2, 3, 4

[8] 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. 3, 9

[9] 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. *arXiv preprint arXiv:2407.00959*, 2024. 3, 5

[10] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024. 3, 5