OmniDrive: A Holistic LLM-Agent Framework for Autonomous Driving with 3D Perception, Reasoning and Planning

Shihao Wang[1]*, Zhiding Yu[2]†, Xiaohui Jiang[1], Shiyi Lan[2], Min Shi[3]*, Nadine Chang[2], Jan Kautz[2], Ying Li[1], and Jose M. Alvarez[2]

[1] Beijing Inst of Tech, [2] NVIDIA, [3] Huazhong Univ of Sci and Tech
https://github.com/NVlabs/OmniDrive

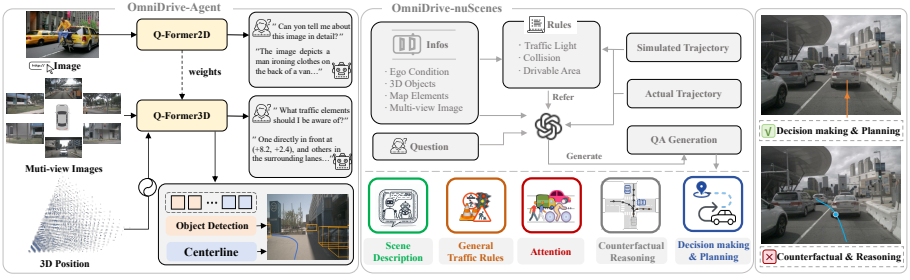

Fig. 1: We present OmniDrive, a novel framework for end-to-end autonomous driving with large language model (LLM) agents. Our main contributions involve novel solutions in both model (OmniDrive-Agent) and benchmark (OmniDrive-nuScenes). The former features a novel 3D vision-language model design, whereas the latter is constituted of comprehensive VQA tasks for reasoning and planning.

Abstract. The advances in multimodal large language models (MLLMs) have led to growing interests in LLM-based autonomous driving to leverage their strong reasoning capabilities. However, capitalizing on MLLMs' strong reasoning capabilities for improved planning behavior is challenging since it requires full 3D situational awareness beyond 2D reasoning. To address this challenge, our work proposes OmniDrive, a holistic framework for strong alignment between agent models and 3D driving tasks. Our framework starts with a novel 3D MLLM architecture that uses sparse queries to lift and compress visual representations into 3D before feeding them into an LLM. This query-based representation allows us to jointly encode dynamic objects and static map elements (e.g., traffic lanes), providing a condensed world model for perception-action alignment in 3D. We further propose a new benchmark with comprehensive visual question-answering (VQA) tasks, including scene description, traffic regulation, 3D grounding, counterfactual reasoning, decision making and planning. Extensive studies show the excellent reasoning and planning capabilities of OmniDrive in complex 3D scenes.

Keywords: Autonomous driving · Large language model · Planning

* Work done during an internship at NVIDIA.
† Corresponding author: zhidingy@nvidia.com.

1 Introduction

The recent rapid development of multimodal LLMs (MLLMs) [1,24,31] and their excellent reasoning capabilities have led to a stream of applications in end-to-end autonomous driving [6,39,46,52,58]. However, the challenge to extend the capabilities from 2D understanding to the intricacies of 3D space is a crucial hurdle to overcome to fully unlock its potential in real-world applications. Understanding and navigating through 3D space is indispensable for autonomous vehicles (AVs) because they directly impact an AV's ability to make informed decisions, anticipate future states, and interact safely with their environment. Although previous works [38,47] have demonstrated successful applications of LLM-agents in autonomous driving, a holistic and principled approach is still needed to fully extend the 2D understanding and reasoning capabilities of MLLMs into complex 3D scenes for understanding the 3D geometry and spatial relations.

Another open issue is the need to address multi-view high resolution video input. On one hand, many current popular 2D MLLM architectures, such as LLaVA-1.5 [30,31], can only take 336×336 image input due to the limited vision encoder resolution and LLM token sequence length. Increasing the limitations is not trivial, as it requires significantly more compute and memories. On the other hand, dealing with high resolution video input, oftentimes even multi-view, is a fundamental requirement for long-range AV perception and safe decision making. However, unlike many cloud-based services, real-world industrial autonomous driving applications are mostly on-device and compute-bound. As such, there is a need to design an efficient MLLM architectures with compressed 3D visual representations before feeding to the LLM.

Our answer to the above challenges is a novel Q-Former-styled [24] 3D MLLM architecture as shown in Fig. 1. Unlike LLaVA which adopts a self-attention design, the cross-attention decoder in Q-Former makes it more scalable to higher resolution input by compressing the visual information into sparse queries. Interestingly, we notice that the Q-Former architecture shares considerable similarity with the family of perspective-view models, such as DETR3D [53], PETR(v2) [32, 33], StreamPETR [50] and Far3D [21]. Using sparse 3D queries, these models have demonstrated considerable advantages over dense bird's-eye view (BEV) representation with leading performance [21,50], long-range perception [21] and capability to jointly model map elements [55]. The similarity in query-based decoder architecture enables us to align both worlds by appending 3D position encoding to the queries, lifting them to 3D, and attending the multi-view input, as shown in the left portion of Fig. 1. This process allows the MLLM to gain 3D spatial understanding with minimal efforts and changes, while leveraging the pre-trained knowledge from the abundant images in 2D.

Besides model architectures, recent drive LLM-agent works also feature the importance of benchmarks [10,38,39,43,46,47]. Many of them are presented as question-answering (QA) datasets to train and benchmark the LLM-agent for either reasoning or planning. Despite the various QA setups, benchmarks that involve planning [10,46,47] still resort to an open-loop setting on real-world sessions (e.g., nuScenes) where expert trajectories are used. Recent studies [26,60],

however, reveal limitations of open-loop evaluation with implicit biases towards ego status, overly simple planning scenarios, and easy overfit to the expert trajectory. In light of the issue, the proposed benchmark OmniDrive-nuScenes features a counterfactual reasoning benchmark setting where simulated decision and trajectories are leveraged to reason potential consequences. Our benchmark also include other challenging tasks that require full 3D spatial understanding and long-horizon reasoning, as shown in Fig. 1 (right).

In summary, OmniDrive aims to provide a holistic framework for end-to-end autonomous driving with LLM-agents. We propose a principled model design excellent in 3D reasoning and planning, as well as a more challenging benchmark going beyond single expert trajectories.

2 OmniDrive-Agent

As a recap, we aim for a unified 3D MLLM design to: 1) leverage the 2D MLLM pre-training knowledge, and 2) addressing the high-resolution multi-view input in autonomous driving. We propose a Q-Former-styled architecture by compressing the visual features into a fixed number of queries before feeding to an LLM [24]. Noticing the similarity between Q-Former and query-based 3D perception frameworks, we align our MLLM architecture with StreamPETR [50], where use queries to encode both dynamic objects and static map elements. These queries, together with additional carrier tokens, serve as a condensed world model to align perception with reasoning and planning.

2.1 Preliminaries

The Q-Former based MLLMs are composed of a general visual encoder to extract single-view image features $F_s \in \mathbb{R}^{C\times H\times W}$, a projector (Q-Former) that serves as visual-language alignment module, and a large language model for text generation. The architecture of the projector is stacked transformer decoder layers. The projection process from image features to the textual embedding can be represented as:

$$\tilde{Q}_t = f_q(Q_t, F_s) \tag{1}$$

where Q_t is the initialized text embedding. $\tilde{Q}_t$ is the refined text embedding, which will be sent to the language model to generate the final text output.

The query-based 3D perception models [21, 28, 29, 55] consist of a shared visual encoder to extract multi-view image features, and a detection head f_d. It is based on the PETR [32] and utilizes transformer decoder architecture to efficiently convert multi-view image features $F_m \in \mathbb{R}^{N\times C\times H\times W}$ into detection queries $\tilde{Q}_d$, which can be formulated as:

$$\tilde{Q}_d = f_d(Q_d, F_m + P_m) \tag{2}$$

where P_m is the 3D position encoding that effectively capturing the geometric relationship between the image view and 3D domains. Q_d is the initialized detection queries to gather the multi-view image features.

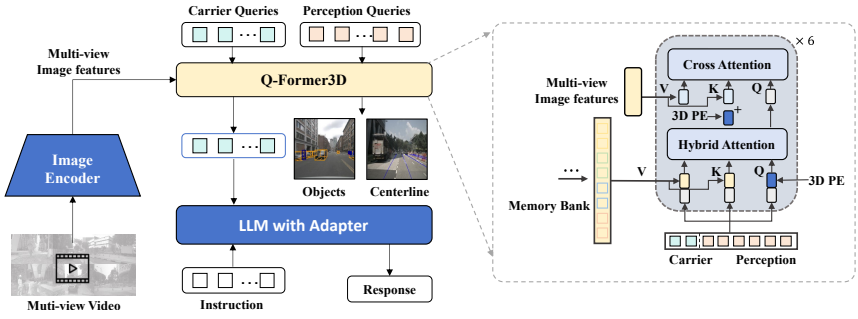

Fig. 2: Overall pipeline of OmniDrive-Agent. The left diagram illustrates the overall framework of the model. We employ a 3D perception task to guide Q-Former's learning. The right diagram depicts the specific structure of Q-Former3D, which is consist of six transformer decoder layers. The attention weights are initialized from 2D pre-pretrain. The input are multi-view image features. Additionally, 3D position encoding is added in the attention operation. Furthermore, we introduce temporal modeling through a memory bank.

It can be observed that the Transformer decoder in Q-Former and the sparse query-based 3D perception models, represented by StreamPETR [50], share highly similar architecture designs. To enhance the localization abilities of the MLLMs, we consider introducing the design of 3D position encoding and the supervision of the query-based perception models.

2.2 Overall Architecture

As show in Fig. 2, Omnidrive first uses a shared visual encoder to extract multi-view image features $F_m \in \mathbb{R}^{N\times C\times H\times W}$. The extracted features with the position encoding P_m are fed into the Q-Former3D. In Q-Former3D, we initialize the detection queries and carrier queries and perform self-attention to exchange their information, which can be summarized by the following formula:

$$\begin{aligned}(Q,K,V) &= ([Q_c,Q_d],[Q_c,Q_d],[Q_c,Q_d]),\\ \tilde{Q} &= \text{Multi-head Attention}(Q,K,V)\end{aligned} \tag{3}$$

$[\cdot]$ is the concatenation operation. For simplicity, we omit the position encoding. Then these queries collect information from multi-view images:

$$\begin{aligned}(Q,K,V) &= ([Q_c,Q_d],P_m+F_m,F_m),\\ \tilde{Q} &= \text{Multi-head Attention}(Q,K,V)\end{aligned} \tag{4}$$

After that, the perception queries are used to predict the categories and coordinates of the foreground elements. The carrier queries are sent to a single layer MLP to align with the dimension of LLM tokens (e.g., 4096 dimensions in LLaMA [48]) and further used for text generation following LLaVA [31].

In our model, the carrier queries play the role of the visual-language alignment. Additionally, this design enables carrier queries to leverage the geometric priors provided by the 3D position encoding, while also allowing them to leverage query-based representations acquired through the 3D perception tasks.

2.3 Multi-task and Temporal Modeling

Our approach benefits from multi-task learning and temporal modeling [25,33]. In multi-task learning, we can integrate task-specific Q-Former3D modules for each perception task, employing a uniform initialization strategy (please refer to Sec. 2.4). In different tasks, carrier queries can gather information of different traffic elements. In our implementation, we cover tasks such as center-line construction and 3D object detection. During the training and inference phases, both heads share the same 3D position encoding. Regarding temporal modeling, we store the perception queries with top-k classification scores into a memory bank and propagate them frame by frame [28, 59]. The propagated queries interact with the perception and carrier queries from the current frame through cross-attention, expanding the capabilities of our model to effectively process video input.

2.4 Training Strategy

The training of OmniDrive-agent comprises two stages: 2D-Pretraining and 3D-Finetuning. In the initial stage, we pretrain the MLLMs on 2D image tasks to initialize the Q-Former and carrier queries. Following this, the model is fine-tuned on 3D-related driving tasks (e.g., motion planning, 3D grounding, etc.). In both stages, we calculate the text generation loss without considering contrasting learning and matching loss for in BLIP-2 [24].

2D-Pretraining. The 2D-Pretraining stage aims to pretrain the carrier queries and the Q-Former, and achieve better alignment between image features and the large language model. When removing the detection queries, the OmniDrive model can be viewed as a standard vision language model capable of generating text conditioned on images. Therefore, we adopt the training recipe and data from LLaVA v1.5 [30] to pretrain OmniDrive on 2D images. The MLLMs is first trained on 558K image-text pairs, during which all parameters except the Q-Former are frozen. Subsequently, we fine-tune the MLLMs using the instruction tuning dataset from LLaVA v1.5. During this fine-tuning step, only the image encoder is frozen, while all other parameters are trainable.

3D-Finetuning. During the 3D finetuning phase, we aim to enhance the model's 3D localization capability while retaining its 2D semantic understanding as much as possible. We have augmented the original Q-Former with 3D position encoding and temporal modules. In this phase, we fine-tune the visual encoder and the large language model with Lora [16] using a small learning rate, and train Q-Former3D with a relatively larger learning rate.

3 OmniDrive-nuScenes

To benchmark drive LLM-agents, we propose OmniDrive-nuScenes, a novel benchmark built on nuScenes [4] with high quality visual question-answering (QA) pairs covering perception, reasoning and planning in 3D domain.

OmniDrive-nuScenes features a fully-automated procedural QA generation pipeline using GPT4. Similar to LLaVA [31], the proposed pipeline takes the 3D perception ground truths as context information via prompt input. Traffic rules and planning simulations are further leveraged as additional inputs, thereby easing the challenges GPT-4V faces in comprehending 3D environments. Our benchmark asks long-horizon questions in the forms of attention, counterfactual reasoning, and open-loop planning. These questions challenge the true spatial understanding and planning capabilities in 3D space as they require planning simulations in the next few seconds to obtain the correct answers.

Besides using the above pipeline to curate the offline question-answering sessions for OmniDrive-nuScenes, we further propose a pipeline to online generate various types of grounding questions. This part can also be viewed as certain form of implicit data augmentation to enhance the 3D spatial understanding and reasoning capabilities of the models.

3.1 Offline Question-Answering.

Tab. 1 shows an example of the proposed offline data generation pipeline, where in-context information is used to generate the QA pairs on nuScenes. We begin with related details on how different types of the prompt information is obtained:
Caption. When both the image and lengthy scene information are fed into GPT-4V simultaneously, GPT-4V tends to overlook details in the image. So we first prompt GPT-4V to produce the scene description based on the multi-view input only. As shown in the top block of Tab. 1, we stitch the three frontal views and three rear views into two separate images, and feed them into GPT-4V, as shown in the top of the Tab. 1. We also prompt GPT-4V to include the following details: 1) mentions weather, time of day, scene type and other image contents; 2) understands the general direction of each view (e.g., the first frontal view being front-left); 3) avoids mentioning the contents from each view independently and replaces with positions relative to the ego vehicle.
Lane-object association. For GPT-4V, understanding the relative spatial relationships of traffic elements (such as objects, lane lines, etc.) in the 3D world is a highly challenging task. Directly inputting the 3D object coordinates and the curve representation of lane lines to GPT-4V does not enable effective reasoning. Therefore, we represent the relationships between objects and lane lines in the form of a file tree, and convert the information of objects into a natural language description based on their 3D bounding boxes.
Simulated trajectories. Trajectories are sampled for counterfactual reasoning in two ways: 1) We select the initial lane based on three driving intentions: lane keeping, left lane change, and right lane change. Then we use the Depth-First Search (DFS) algorithm to link the lane centerline and get all possible vehicle

Table 1: An example to illustrate the instruction-following data generation for offline QAs. The top block shows contexts such as captions and boxes used to prompt GPT4, and the bottom block shows the four types of responses. Note that the visual image is not used to prompt GPT4 except for caption generation.

Prompt type 1: Caption
The images depict a daytime setting in a controlled-access area, likely a parking lot or a service entrance of a commercial or industrial facility. On the left, there's a grassy area with trees and a building with blue accents. Moving towards the center, we see a security checkpoint with a raised barrier arm, indicating an entrance or exit point, and a security guard booth...<omitted>
Prompt type 2: Lane-object association
|— your current straight lane [(-2.6, +0.5), (+1.2, +0.7), (+5.0, +0.9), (+8.8, +1.0)]
| |— your own car
| |— movable_object.trafficcone in the front location: (+8.2, +2.4)...<omitted>
Prompt type 3: Simulated decision and trajectory
Simulated decision: Moderate Speed, Left Turn
Simulated trajectory: [PT, (+4.85, -0.08), (+9.71, -0.22), ..., (+27.42, -0.93)].
Out of the drivable area
Prompt type 4: Expert decision and trajectory
Expert decision: Moving Slowly, Lane Keeping, Go Straight
Expert trajectory: [PT, (+0.76, +0.02), (+1.45, +0.03), ..., (+3.44, +0.12)].
Objects near your path: vehicle.moving at (-6.0, +0.3), human.pedestrian.moving at (+4.4, -7.2).

Conversation type 1: Attention
Question: What traffic elements should I be aware of while driving in this area?
Answer: There are movable traffic cones at various points along your path: one directly in front at (+8.2, +2.4), and others in the surrounding lanes, such as those at (+4.7, -6.3)....<omitted>
Conversation type 2: Counterfactual reasoning
Question: If I decide to accelerate and make a left turn, what could be the consequences?
Answer: It appears that you would be turning into the lane designated for allowing from right to left driving. This maneuver could potentially lead to a collision with oncoming traffic...<omitted>
Conversation type 3: Decision making and planning
Question: What should be my next action given the current driving situation, and why?
Answer: The most suitable trajectory to follow would be similar to the one provided as safe, which is [PT, (+0.76, +0.02), (+1.45, +0.03),...,(+3.44, +0.12)]. This trajectory ensures that you remain in your current straight lane [(-2.6, +0.5), (+1.2, +0.7), ...], avoid any obstacles...<omitted>
Other Conversations:
...<omitted>

trajectory paths. Then selecting different completion rates and speed targets for various lanes (acceleration, deceleration, and speed maintenance) to create simulated trajectories. 2) Generating driving trajectories based solely on lane centerlines makes it difficult to simulate scenarios that are 'out of the drivable'. Therefore, we also performed clustering on the entire nuScenes dataset's ego trajectories, selecting representative driving paths each time.

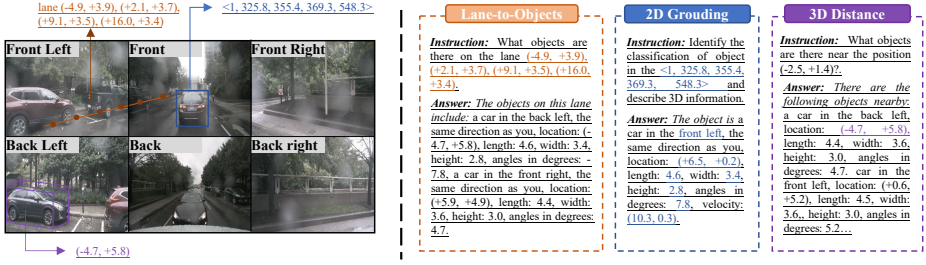

Fig. 3: Online QA generation.

Expert trajectory. This is the log replay trajectory from nuScenes. The expert trajectories are classified into different types for high-level decision making. We also identify an object as "close", if its minimum distance to the trajectory is smaller than 10 meters in the next 3 seconds. The close objects are then listed below the expert trajectory.

In the bottom block of the Tab. 1, we describe the different types of QA responses obtained by using the above context information:

Scene description. We directly take caption (prompt type 1 in Tab. 1) as the answer of scene description.

Attention. Given the simulated and expert trajectories, run simulation to identify close objects. At the same time, we also allowed GPT4 to use its own common sense to identify threatening traffic elements.

Counterfactual reasoning. Given the simulated trajectories, we simulate to check if the trajectories violate the traffic rules, such as run a red light, collision to other objects or the road boundary.

Decision making and planning. We present the high-level decision making as well as the expert trajectory and use GPT-4V to reason why this trajectory is safe, given the previous prompt and response information as context.

General Conversation. We also prompt GPT-4 with generating multi-turn dialogues based on caption information and image content, involving the object countings, color, relative position, and OCR-type tasks. We found that this approach helps improve the model's recognition of long-tail objects.

3.2 Online Question-Answering.

To fully leverage the 3D perception labels in the autonomous driving dataset, we generate numerous grounding-like tasks during the training process in an online manner. Specifically, the following tasks (in Fig. 3) are designed:

2D-to-3D Grounding. Given a 2D bounding box on a specific camera, e.g., $< \text{FRONT}, 0.45, 0.56, 0.72, 0.87 >$, the model is required to provide the 3D properties of the corresponding object, including its 3D categories, location, size, orientation, and velocity.

3D Distance. Based on the randomly generated 3D coordinate, identify the traffic elements near the corresponding location and provide the 3D properties of the traffic elements.
Lane-to-objects. Based on the randomly selected lane center-line, list the objects present on this lane and their 3D properties.

3.3 Metrics

The proposed OmniDrive dataset involves captioning, open-loop planning and counterfactual reasoning tasks. Each of these tasks has distinct emphasis, and it's challenging to evaluate them with a single metric. In this section, we will elaborate on how we assess the performance of models on our dataset.

For caption-related tasks, such as scene description and the selection of attention objects, we utilize the commonly employed language-based metrics to evaluate the sentence similarity at the word level, including **METEOR** [3], **ROUGE** [27], and **CIDEr** [49]. Following BEV-Planner [26], **Collision Rate** and **Intersection Rate** with the road boundary are adopted to evaluate the performance of the Open-loop planning. To evaluate the performance of the counterfactual reasoning, we ask GPT-3.5 to extract keywords based on the predictions. The keywords include 'safety,' 'collision,', 'running a red light,' and 'out of the drivable area.' Then we compare extracted keywords with the ground truth to calculate the **Precision** and **Recall** for each category of the accident.

4 Experiment

4.1 Implementation Details

Our model uses EVA-02-L [14] as the vision encoder. It applies masked image modeling to distill CLIP [44], which can extract language-aligned vision features.

During the 2D pre-training stage, the training data and strategies, including batchsize, learning rate, and optimizer are the same as LLaVA v1.5's [30]. In the finetuning stage, the model is trained by AdamW [34] optimizer with a batch size of 16. The learning rate for the projector is 4e-4, while the visual encoder and the LLM's learning rates are 5e-4. The cosine annealing policy is employed for the training stability. The models in the ablation study are trained for 6 epochs unless specified otherwise. The number of object queries, lane queries and carrier queries is set to 900, 300 and 256 respectively.

We also explore alternative architectures. Q-Former2D is initialized with 2D pre-trained weights. It processes the image features individually in the projector and fuses them in the LLM. The Dense BEV approach uses LSS method [40,41] to transform perspective features into a BEV feature map. We implement temporal modeling following SOLOFusion [40]. The BEV features will be consecutively fed into a MLP projector and a LLM.

Table 2: Ablation study on planning related tasks. P and R represent Precision and Recall respectively. "No online" means removing the online traing data. "No temporal" means removing the temporal modules. Freeze means no gradients are applied to the backbone layers. "No Object" and "No Lane" indicate no corresponding 3D perception supervision. In this table, **none of the models** in the planner are using high-level command and ego status.

Ablation	Exp.	Counterfactual								Open-loop	
		Safe		Red Light		Collision		Drivable Area		Col(%)	Inter(%)
		P	R	P	R	P	R	P	R	Avg.	Avg.
Full Model	Q-Former3D	70.7	49.0	57.6	58.3	**32.3**	**72.6**	**48.5**	**58.6**	**3.79**	4.59
Data	No Online	69.4	39.4	36.2	**65.6**	29.7	69.4	48.0	57.8	4.93	**4.02**
Architecture	Q-Former2D	**71.4**	39.3	**58.3**	61.1	32.0	66.7	44.4	52.8	3.98	6.03
	Dense BEV	70.2	17.3	48.7	53.6	31.1	70.4	32.4	56.6	4.43	8.56
	No Temporal	67.8	48.4	47.0	62.6	31.2	63.8	46.5	55.3	6.07	5.83
Perception Supervision	No Lane	67.7	57.3	58.1	59.6	31.0	56.7	47.9	56.8	4.65	8.71
	No Object & Lane	69.0	**57.8**	51.3	61.2	30.0	53.2	45.3	57.1	6.77	8.43

4.2 Planning with Counterfactual Reasoning

Based on our Omnidrive-nuScenes, we ablate various modifications in training recipes and model architectures. All analysis shown in Tab. 2 are conducted without using high-level commands and ego status [20, 26]. Under this setting, it can be observed that there is a certain correlation between counterfactual metrics and open-loop planning.

We found that Q-former2D performs better on 2D-related tasks, such as determining the status of traffic lights. However, Q-Former3D clearly has a greater advantage in 3D tasks such as collision detection (32.2% in Precision and 72.6% in Recall) and drivable area identification (48.5% in Precision and 58.6% in Recall). The models with center-line construction tasks (i.e., Full Model) outperforms the one without lane supervision in drivable area tasks.

4.3 Ablation Study & Analysis

Counterfactual Reasoning and Captioning. In Tab. 3, the Full Model achieves the best performance in terms of the counterfactual reasoning, with average precision of 52.3% and average recall of 59.6% .

More importantly, we show that our model benefits strongly from Q-Former3D and achieves comparable performance to that of Q-Former2D on caption tasks with 38.0% METEOR score, 68.6% CIDEr score and 32.6 ROUGE score. Furthermore, our model can process multi-view cameras simultaneously, while Q-Former2D processes each view separately and necessitating an inefficiently high number of tokens (1500+) for input to LLM. We note that the dense BEV model yields the poorest results because it fails to leverage the benefits brought by 2D pre-training. We also found that, with the same Q-Former equipped with pre-training, the performance on descriptive tasks is similar. Introducing additional 3D supervision and temporal information did not result in significant improvements.

Table 3: Analysis on our benchmark.

Abaltion	Exp.	Counterfactual AP (%)	Counterfactual AR (%)	Caption METEOR↑	Caption CIDEr ↑	Caption ROUGE↑
Full Model	Q-Former3D	**52.3**	**59.6**	38.0	68.6	32.6
Data	No Online	45.8	58.1	38.2	**69.0**	**32.7**
Architecture	Q-Former2D	51.5	55.0	**38.3**	67.1	32.5
	Dense BEV	45.6	49.5	35.6	59.5	27.8
	No Temporal	48.1	57.5	37.9	68.4	32.6
Perception Supervision	No Lane	51.2	57.6	38.0	67.8	32.6
	No Object & Lane	48.9	57.3	38.2	67.8	32.6

Comparison on NuScenes-QA. We also present results on NuScenes-QA in Tab. 4. In NuScenes-QA, most answers in NuScenes-QA are single-word and related to perception only. In the same camera modality, our model surpasses BEVDet+MCAN by 1.3% in accuracy, demonstrating the importance of pre-training. Our model's performance is comparable to the Lidar modality's models.

Model	Modality	Acc.(%)
BEVDet+BUTD [43]	C	57.0
BEVDet+MCAN [43]	C	57.9
CenterPoint+BUTD [43]	L	58.1
CenterPoint+MCAN [43]	L	**59.5**
OmniDrive	C	**59.2**

Table 4: Results on NuScenes-QA [43]. L and C represent Lidar and Camera respectively. We highlight the SoTA methods in each modality.

Table 5: Comparison on the Open-loop planning. For a fair comparison, we referred to the reproduced results in BEV-Planner [26]. †: The official implementation of ST-P3 (ID-0) utilized partial erroneous ground truth. ‡: The high-level command is not used during the training and testing phases.

Method	Ego Status BEV	Ego Status Planner	L2 (m) ↓ 1s	2s	3s	Avg.	Collision (%) ↓ 1s	2s	3s	Avg.	Intersection (%) ↓ 1s	2s	3s	Avg.
ST-P3	-	-	1.59†	2.64†	3.73†	2.65†	0.69†	3.62†	8.39†	4.23†	2.53†	8.17†	14.4†	8.37†
UniAD	-	-	0.59	1.01	1.48	1.03	0.16	0.51	1.64	0.77	0.35	1.46	3.99	1.93
UniAD	✓	✓	0.20	0.42	0.75	0.46	0.02	0.25	0.84	0.37	**0.20**	**1.33**	**3.24**	**1.59**
VAD-Base	-	-	0.69	1.22	1.83	1.25	0.06	0.68	2.52	1.09	1.02	3.44	7.00	3.82
VAD-Base	✓	✓	0.17	0.34	0.60	0.37	0.04	0.27	**0.67**	0.33	0.21	2.13	5.06	2.47
Ego-MLP	-	✓	0.15	0.32	0.59	0.35	**0.00**	0.27	0.85	0.37	0.27	2.52	6.60	2.93
BEV-Planner	-	-	0.30	0.52	0.83	0.55	0.10	0.37	1.30	0.59	0.78	3.79	8.22	4.26
BEV-Planner++	✓	✓	0.16	0.32	0.57	0.35	**0.00**	0.29	0.73	0.34	0.35	2.62	6.51	3.16
OmniDrive‡	-	-	1.15	1.96	2.84	1.98	0.80	3.12	7.46	3.79	1.66	3.86	8.26	4.59
OmniDrive	-	-	0.40	0.80	1.32	0.84	0.04	0.46	2.32	0.94	0.93	3.65	8.28	4.29
OmniDrive++	✓	✓	**0.14**	**0.29**	**0.55**	**0.33**	**0.00**	**0.13**	0.78	**0.30**	0.56	2.48	5.96	3.00

4.4 Discussion on Open-loop Planning

We compare the proposed OmniDrive with previous state-of-the-art vision-based planners in Tab. 5. The MLLM based open-loop planning can also achieve comparable performance to SoTA methods. However, as mentioned in BEV-Planner [26], it is observed that encoding the ego status significantly improves

the metrics across all methods. Additionally, we found that the high-level command also drastically reduces the collision rate and the intersection rate. Previous methods provided high-level commands based on the relative position of the ground-truth trajectory, which imposed significant constraints on the network's regression, thereby reducing the likelihood of deviating too far from the ground truth trajectory. We believe this design is also unreasonable, so in other experiments, we eliminated this setting.

Table 6: Visualization results on scene description and counterfactual reasoning.

Scene Description & Counterfactual Reasoning	
	
User	Can you summarize what the panoramic images show?
Ours	The panoramic images depict a rainy day in an urban setting with wet roads reflecting the overcast sky. On the left-front, we see a parking lot entrance and a traffic light showing green, indicating the flow of traffic is permitted. The front view reveals a multi-lane road with vehicles, including a white car turning right, and traffic lights at an intersection. The right-front shows a pedestrian crossing with a traffic light, and a person waiting to cross. The left-rear view captures a traffic light at red, a pedestrian crossing, and a black car waiting. The rear view shows a large building under construction, with a crane towering above and a few vehicles on the road. The right-rear view includes a traffic light at red, a pedestrian crossing, and a road leading to a roundabout, with a black car approaching the intersection. The overall scene is one of a bustling city despite the damp weather.
User	What would be the outcome if you decided to move slowly and make a right turn?
Ours	If I were to move slowly and make a right turn, following the trajectory [PT, (+2.44, -0.09), (+4.91, -0.36), (+7.41, -0.84), (+9.91, -1.53), (+12.37, -2.44), (+14.76, -3.55)], it would result in leaving the drivable area, which is a dangerous and incorrect action. The trajectory indicates a turn that is too sharp and not aligned with the road layout, leading to potential collisions or driving off the road. In this scenario, the correct behavior would be to maintain my lane and follow the road's curvature, ensuring I stay within the bounds of the drivable area and adhere to traffic rules.

4.5 Visualization

A challenging scene is shown in Tab. 6. OmniDrive-Agent shows impressive results on scene description and counterfactual reasoning. The model has a basic understanding of relative positions and can make counterfactual reasoning correctly based on given trajectories and safety considerations.

5 Related Works

5.1 End-to-End Autonomous Driving

The objective of end-to-end autonomous driving is to create a fully differentiable system that spans from sensor input to control signals [42, 57, 61]. This system allows for the joint optimization of the entire system, thereby mitigating the accumulation of errors. The current technical road-map is primarily divided into two paths: open-loop autonomous driving and closed-loop autonomous driving.

In the open-loop autonomous driving, the training and evaluation processes are generally conducted on log-replayed real world datasets [43]. This approach overlooks the impact of interactions between the ego vehicle and other traffic participants, leading to cumulative errors. Pioneering work UniAD [17] and VAD [20] integrate modularized design of perception tasks such as object detection, tracking, and semantic segmentation into a unified planning framework. However, Ego-MLP [60] and BEV-Planner [26] highlight the limitations of open-loop end-to-end driving benchmarks. In these benchmarks, models may overfit the ego-status information to achieve unreasonably high performance. Researchers are addressing the challenges in open-loop evaluation by introducing closed-loop benchmarks. Recent works, e.g., MILE [15], ThinkTwice [19], VADv2 [7] leverage CARLA [11] as the simulator, which enables the creation of virtual environments with feedback from other agents. Researchers urgently need a reasonable way to evaluate end-to-end autonomous driving systems in the real world. MLLM models bridge the gap between data-driven decision-making and the user. This enables us to perform interpretable analysis and conduct counterfactual reasoning based on a specific trajectory, thereby enhancing the safety redundancy of the autonomous driving system.

5.2 Multimodal Language Models (MLLMs)

Muiltimodal language models leverage LLMs and various modalities' encoders to successfully bridge the gap between language and other modalities and perform well on multimodal tasks ranging from visual question answer, captioning, and open-world detection. Some MLLMs such as CLIP [44] and ALIGN [18] utilize contrastive learning to create a similar embedding space for both language and vision. More recently, others such as BLIP-2 [24] explicitly targets multimodal tasks and takes multimodal inputs. For these models, there are two common techniques in order to align language and other input modalities: self-attention and cross-attention. LLaVa [31], PaLM-E [12], PaLI [8], and RT2 [62] utilize self-attention for alignment by interleaving or concatenating image and text tokens in fixed sequence lengths. However, self-attention based MLLMs are unable to handle high resolution inputs and are unsuitable for autonomous driving with multi-camera high solution images. Conversely, Flamingo [1], Qwen-VL [2], BLIP-2 [24], utilize cross-attention and are able to extract a fixed number of visual tokens regardless of image resolution. Because of this, our model utilizes Qformer architecture from BLIP-2 to handle our high resolution images.

5.3 Drive LLM-Agents and Benchmarks

Drive LLM-Agents. Given LLM/MLLMs' high performance and ability to align modalities with language, there is a rush to incorporate MLLMs/LLMs with autonomous driving (AD). Most AD MLLMs methods attempt to create explainable autonomous driving with end-to-end learning. DriveGPT4 leverages LLMs to generate reasons for car actions while also predicting car's next control signals [58]. Similarly, Drive Anywhere proposes a patch-aligned feature extraction for MLLMs that allow it to provide text query-able driving decisions [51]. Other works leverage MLLMs through graph-based VQA (DriveLM) [46] or chain-of-thought (CoT) design [47, 54]. They explicitly solve multiple driving tasks alongside typical MLLM tasks, such as generating scene description and analysis, prediction, and planning.

Benchmarks. To evaluate AD perception and planning, there are various datasets that capture perception, planning, steering, motion data (ONCE [37], NuPlan [5], nuScenes [4], CARLA [11], Waymo [13]). However, datasets with more comprehensive lanugage annotations are required to evaluate Drive LLM methods. Datasets focused on perception and tracking include reasoning, or descriptive like captions range from nuScenes-QA [43], NuPrompt, [56]. HAD and Talk2Car both contain human like advice to best navigate the car [9, 22], while LaMPilot contains labels meant to evaluate transition from human commands to drive action [35]. Beyond scene descriptions, DRAMA [36] and Rank2Tell [45] focus on risk object localization. Contrastly, BDD-X, Reason2Drive focus on car explainability by providing reasons behind ego car's action and behavior [23,38,39]. LingoQA [38] has introduced counterfactual questions into the autonomous driving QA dataset. We believe that the interpretability and safety redundancy of autonomous driving in the open-loop setting can be further enhanced by applying counterfactual reasoning to 3D trajectory analysis.

6 Conclusion

We address the challenges of end-to-end autonomous driving with LLM-agents by proposing OmniDrive-Agent and OmniDrive-nuScenes. OmniDrive-Agent adopts a novel Q-Former3D MLLM architecture that can efficiently handle high resolution multi-view videos. Our model design enables minimal adjustments to leverage 2D pre-trained knowledge while gaining important 3D spatial understanding. We additionally provide a novel benchmark for end-to-end autonomous driving which features counterfactual reasoning alongside 3D spatial awareness and reasoning tasks. OmniDrive-Agent demonstrates the efficacy by addressing high-resolution multi-view video input and illustrate excellent scene description and counterfactual reasoning. The model also yields compelling performance on open-loop 3D planning.

Limitations. Our method has not been validated on even larger datasets e.g., nuPlan [5]. The simulation of counterfactual outcomes, despite moving beyond single trajectories, does not yet consider reaction from other agents. This part can be further formed as a closed-loop setup, and we will leave it for future work.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. In: NeurIPs (2022) 2, 13
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-VL: A frontier large vision-language model with versatile abilities. arXiv:2308.12966 (2023) 13
3. Banerjee, S., Lavie, A.: METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In: ACL workshop (2005) 9
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 6, 14
5. Caesar, H., Kabzan, J., Tan, K.S., Fong, W.K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., Omari, S.: NuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles. arXiv:2106.11810 (2021) 14
6. Chen, L., Sinavski, O., Hünermann, J., Karnsund, A., Willmott, A.J., Birch, D., Maund, D., Shotton, J.: Driving with LLMs: Fusing object-level vector modality for explainable autonomous driving. arXiv:2310.01957 (2023) 2
7. Chen, S., Jiang, B., Gao, H., Liao, B., Xu, Q., Zhang, Q., Huang, C., Liu, W., Wang, X.: VADv2: End-to-end vectorized autonomous driving via probabilistic planning. arXiv:2402.13243 (2024) 13
8. Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al.: PaLI: A jointly-scaled multilingual language-image model. arXiv:2209.06794 (2022) 13
9. Deruyttere, T., Vandenhende, S., Grujicic, D., Van Gool, L., Moens, M.F.: Talk2Car: Taking control of your self-driving car. In: EMNLP-IJCNLP (2019) 14
10. Ding, X., Han, J., Xu, H., Liang, X., Zhang, W., Li, X.: Holistic autonomous driving understanding by bird's-eye-view injected multi-modal large models. arXiv:2401.00988 (2024) 2
11. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: CoRL (2017) 13, 14
12. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv:2303.03378 (2023) 13
13. Ettinger, S., Cheng, S., Caine, B., Liu, C., Zhao, H., Pradhan, S., Chai, Y., Sapp, B., Qi, C.R., Zhou, Y., Yang, Z., Chouard, A., Sun, P., Ngiam, J., Vasudevan, V., McCauley, A., Shlens, J., Anguelov, D.: Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In: ICCV (2021) 14
14. Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y.: EVA-02: A visual representation for neon genesis. arXiv:2303.11331 (2023) 9
15. Hu, A., Corrado, G., Griffiths, N., Murez, Z., Gurau, C., Yeo, H., Kendall, A., Cipolla, R., Shotton, J.: Model-based imitation learning for urban driving. In: NeurIPS (2022) 13
16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. arXiv:2106.09685 (2021) 5
17. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR (2023) 13
18. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021) 13

19. Jia, X., Wu, P., Chen, L., Xie, J., He, C., Yan, J., Li, H.: Think Twice before Driving: Towards scalable decoders for end-to-end autonomous driving. In: CVPR (2023) 13
20. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: VAD: Vectorized scene representation for efficient autonomous driving. arXiv:2303.12077 (2023) 10, 13
21. Jiang, X., Li, S., Liu, Y., Wang, S., Jia, F., Wang, T., Han, L., Zhang, X.: Far3d: Expanding the horizon for surround-view 3d object detection. arXiv:2308.09616 (2023) 2, 3
22. Kim, J., Misu, T., Chen, Y.T., Tawari, A., Canny, J.: Grounding human-to-vehicle advice for self-driving vehicles. In: CVPR (2019) 14
23. Kim, J., Rohrbach, A., Darrell, T., Canny, J., Akata, Z.: Textual explanations for self-driving vehicles. ECCV (2018) 14
24. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023) 2, 3, 5, 13
25. Li, Z., Deng, H., Li, T., Huang, Y., Sima, C., Geng, X., Gao, Y., Wang, W., Li, Y., Lu, L.: BEVFormer++ : Improving bevformer for 3d camera-only object detection: 1st place solution for waymo open dataset challenge 2022 (2023) 5
26. Li, Z., Yu, Z., Lan, S., Li, J., Kautz, J., Lu, T., Alvarez, J.M.: Is ego status all you need for open-loop end-to-end autonomous driving? arXiv:2312.03031 (2023) 2, 9, 10, 11, 13
27. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. pp. 74–81 (2004) 9
28. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4D: Multi-view 3d object detection with sparse spatial-temporal fusion. arXiv:2211.10581 (2022) 3, 5
29. Liu, H., Teng, Y., Lu, T., Wang, H., Wang, L.: SparseBEV: High-performance sparse 3d object detection from multi-camera videos. In: ICCV (2023) 3
30. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv:2310.03744 (2023) 2, 5, 9
31. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. NeurIPS (2023) 2, 4, 6, 13
32. Liu, Y., Wang, T., Zhang, X., Sun, J.: PETR: Position embedding transformation for multi-view 3d object detection. arXiv:2203.05625 (2022) 2, 3
33. Liu, Y., Yan, J., Jia, F., Li, S., Gao, Q., Wang, T., Zhang, X., Sun, J.: PETRv2: A unified framework for 3d perception from multi-camera images. arXiv:2206.01256 (2022) 2, 5
34. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. arXiv:1608.03983 (2016) 9
35. Ma, Y., Cui, C., Cao, X., Ye, W., Liu, P., Lu, J., Abdelraouf, A., Gupta, R., Han, K., Bera, A., et al.: LaMPilot: An open benchmark dataset for autonomous driving with language model programs. arXiv:2312.04372 (2023) 14
36. Malla, S., Choi, C., Dwivedi, I., Choi, J.H., Li, J.: DRAMA: Joint risk localization and captioning in driving. In: WACV (2023) 14
37. Mao, J., Niu, M., Jiang, C., Liang, X., Li, Y., Ye, C., Zhang, W., Li, Z., Yu, J., Xu, C., et al.: One million scenes for autonomous driving: Once dataset (2021) 14
38. Marcu, A.M., Chen, L., Hünermann, J., Karnsund, A., Hanotte, B., Chidananda, P., Nair, S., Badrinarayanan, V., Kendall, A., Shotton, J., et al.: LingoQA: Video question answering for autonomous driving. arXiv:2312.14115 (2023) 2, 14

39. Nie, M., Peng, R., Wang, C., Cai, X., Han, J., Xu, H., Zhang, L.: Reason2Drive: Towards interpretable and chain-based reasoning for autonomous driving. arXiv:2312.03661 (2023) 2, 14
40. Park, J., Xu, C., Yang, S., Keutzer, K., Kitani, K., Tomizuka, M., Zhan, W.: Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. arXiv:2210.02443 (2022) 9
41. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV (2020) 9
42. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: CVPR (2021) 13
43. Qian, T., Chen, J., Zhuo, L., Jiao, Y., Jiang, Y.G.: NuScenes-QA: A multi-modal visual question answering benchmark for autonomous driving scenario. arXiv:2305.14836 (2023) 2, 11, 13, 14
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 9, 13
45. Sachdeva, E., Agarwal, N., Chundi, S., Roelofs, S., Li, J., Kochenderfer, M., Choi, C., Dariush, B.: Rank2Tell: A multimodal driving dataset for joint importance ranking and reasoning. In: WACV (2024) 14
46. Sima, C., Renz, K., Chitta, K., Chen, L., Zhang, H., Xie, C., Luo, P., Geiger, A., Li, H.: DriveLM: Driving with graph visual question answering. arXiv:2312.14150 (2023) 2, 14
47. Tian, X., Gu, J., Li, B., Liu, Y., Hu, C., Wang, Y., Zhan, K., Jia, P., Lang, X., Zhao, H.: DriveVLM: The convergence of autonomous driving and large vision-language models. arXiv:2402.12289 (2024) 2, 14
48. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288 (2023) 4
49. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: CIDEr: Consensus-based image description evaluation. In: CVPR (2015) 9
50. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. arXiv:2303.11926 (2023) 2, 3, 4
51. Wang, T.H., Maalouf, A., Xiao, W., Ban, Y., Amini, A., Rosman, G., Karaman, S., Rus, D.: Drive Anywhere: Generalizable end-to-end autonomous driving with multi-modal foundation models. arXiv:2310.17642 (2023) 14
52. Wang, W., Xie, J., Hu, C., Zou, H., Fan, J., Tong, W., Wen, Y., Wu, S., Deng, H., Li, Z., et al.: DriveMLM: Aligning multi-modal large language models with behavioral planning states for autonomous driving. arXiv:2312.09245 (2023) 2
53. Wang, Y., Vitor Campagnolo, G., Zhang, T., Zhao, H., Solomon, J.: DETR3D: 3d object detection from multi-view images via 3d-to-2d queries. In: CoRL (2022) 2
54. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. NeurIPS (2022) 14
55. Wu, D., Chang, J., Jia, F., Liu, Y., Wang, T., Shen, J.: TopoMLP: A simple yet strong pipeline for driving topology reasoning. arXiv:2310.06753 (2023) 2, 3
56. Wu, D., Han, W., Wang, T., Liu, Y., Zhang, X., Shen, J.: Language prompt for autonomous driving. arXiv:2309.04379 (2023) 14
57. Wu, P., Jia, X., Chen, L., Yan, J., Li, H., Qiao, Y.: Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In: NeurIPS (2022) 13

58. Xu, Z., Zhang, Y., Xie, E., Zhao, Z., Guo, Y., Wong, K.K., Li, Z., Zhao, H.: DriveGPT4: Interpretable end-to-end autonomous driving via large language model. arXiv:2310.01412 (2023) 2, 14
59. Yuan, T., Liu, Y., Wang, Y., Wang, Y., Zhao, H.: StreamMapNet: Streaming mapping network for vectorized online hd map construction. arXiv:2308.12570 (2023) 5
60. Zhai, J.T., Feng, Z., Du, J., Mao, Y., Liu, J.J., Tan, Z., Zhang, Y., Ye, X., Wang, J.: Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. arXiv:2305.10430 (2023) 2, 13
61. Zhang, Z., Liniger, A., Dai, D., Yu, F., Van Gool, L.: End-to-end urban driving by imitating a reinforcement learning coach. In: ICCV (2021) 13
62. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: CoRL (2023) 13