

OpenDriveVLA: Towards End-to-end Autonomous Driving with Large Vision Language Action Model

Xingcheng Zhou^{1†}, Xuyuan Han¹, Feng Yang¹, Yunpu Ma², Volker Tresp², Alois Knoll¹

¹Technical University of Munich ²Ludwig Maximilian University of Munich

<https://drivevla.github.io>

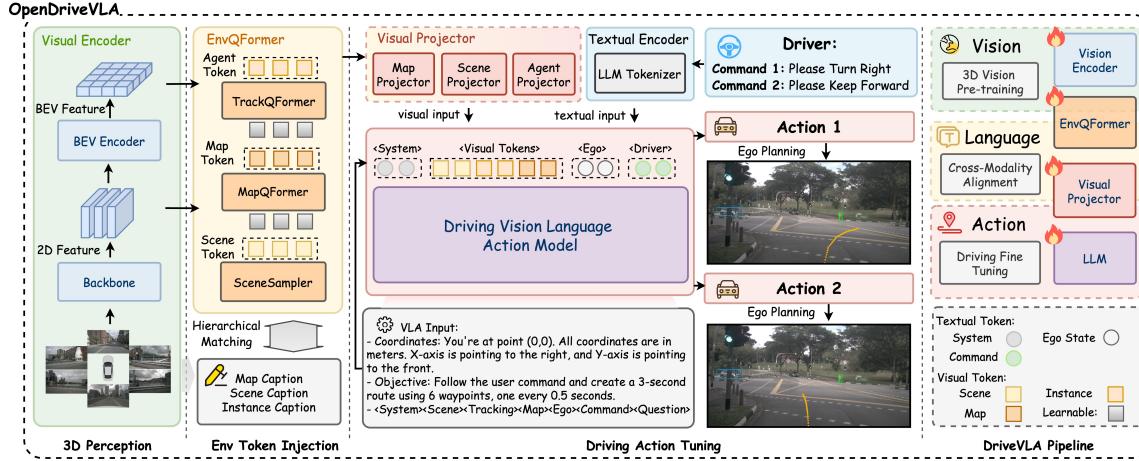


Fig. 1: OpenDriveVLA leverages open-source pre-trained large vision-language-action models to generate driving actions conditioned on 3D environmental perception, ego-vehicle states, and driver commands. It achieves strong performance in both open-loop planning and driving-related question answering, demonstrating its proficiency in scene understanding and driving action tuning.

Abstract—We present OpenDriveVLA, a Vision-Language Action (VLA) model designed for end-to-end autonomous driving, built upon open-source large language models. OpenDriveVLA generates spatially-grounded driving actions by leveraging multimodal inputs, including both 2D and 3D instance-aware visual representations, ego vehicle states, and language commands. To bridge the modality gap between driving visual representations and language embeddings, we introduce a hierarchical vision-language alignment process, projecting both 2D and 3D structured visual tokens into a unified semantic space. Furthermore,

we incorporate structured agent–environment–ego interaction modeling into the autoregressive decoding process, enabling the model to capture fine-grained spatial dependencies and behavior-aware dynamics critical for reliable trajectory planning. Extensive experiments on the nuScenes dataset demonstrate that OpenDriveVLA achieves state-of-the-art results across open-loop trajectory planning and driving-related question-answering tasks. Qualitative analyses further illustrate its superior capability to follow high-level driving commands and generate trajectories under challenging scenarios, highlighting its potential for next-generation end-to-end autonomous driving.

[†]Corresponding author: xingcheng.zhou@tum.de

I. INTRODUCTION

End-to-end learning frameworks have emerged as a promising paradigm in autonomous driving, enabling perception, prediction, and planning to be jointly optimized within a unified neural network [1]. They learn policies directly from sensor inputs and generalize well across varied scenarios. Despite notable progress, existing approaches still face critical challenges, including limited long-tail generalization, poor complex semantics understanding, and rigid task reasoning [2]. Meanwhile, large language models (LLMs) and vision-language models (VLMs) exhibit strong in-context reasoning, commonsense understanding, and zero-shot generalization abilities. These capabilities are promising for driving, where robust scene understanding is crucial [3], [4]. However, directly leveraging existing VLMs for autonomous driving poses fundamental challenges. Firstly, current VLMs are predominantly optimized for static, 2D image-language tasks, leading to poor spatial reasoning performance in dynamic 3D driving environments [5]. Besides, instance-agnostic VLMs [6] are prone to hallucinations, often yielding incorrect yet overconfident outputs, posing safety risks in autonomous driving. Motivated by these limitations, our work answers a central question: **How can we harness the emergent capabilities of large VLMs to produce safe spatially-grounded driving actions in dynamic 3D environments, while balancing inference speed and planning effectiveness?**

To enhance spatial-awareness and safety in LLM-based vision-language action model, we introduce two key designs. First, we structure the driving environment using instance-aware, hierarchical 2D and 3D visual representations to reduce the risk of instance hallucinations. Second, we incorporate agent–environment–ego interaction modeling, which is originally explicitly modeled in traditional end-to-end driving systems, as an auxiliary objective into the autoregressive LLM training pipeline. It enables the model to internalize physical feasibility and dynamic multi-agent interactions, improving robustness in safety-critical scenarios.

Built upon open-source large language models, OpenDriveVLA tightly integrates spatially-

grounded multimodal reasoning and driving trajectory generation within a unified autoregressive framework. Unlike prior VLM-based methods, OpenDriveVLA leverages structured 2D and 3D instance-aware representations, ego vehicle states, and high-level commands to directly produce reliable driving actions. Extensive experiments on nuScenes benchmark demonstrate that OpenDriveVLA achieves state-of-the-art performance in both open-loop planning and vision-language reasoning tasks. Our key contributions are:

- We present OpenDriveVLA, a 3D vision-language action model for end-to-end autonomous driving that generates reliable driving trajectories by integrating hierarchical visual input, ego state, and high-level language commands.
- We develop a multi-stage training strategy that aligns structured 2D and 3D visual features into a unified semantic space, enabling naive VLMs to generate spatially-grounded actions in complex driving scenarios.
- We introduce implicit agent–environment–ego interaction modeling into autoregressive LLM-based VLA training as an auxiliary task, enabling the model to learn behaviorally grounded and safety-aware driving actions.

II. RELATED WORK

A. End-to-End Autonomous Driving

Autonomous driving (AD) evolves through two distinct stages. Traditional approaches rely on a modular design, decomposing the system into perception [7], prediction [8], and planning [9] components. While this structure ensures interpretability and allows for independent optimization, they suffer from cascading errors between stages and are not globally optimized for the final planning objective. In contrast, end-to-end autonomous driving frameworks [10] address this by jointly optimizing perception, prediction, and planning within a unified neural network. These models learn driving policies directly from raw sensor inputs, which improves the model’s adaptability to diverse driving conditions. More recent approaches introduce diffusion models [11] and unified scene representations [12] to

further enhance the effectiveness and robustness. However, existing end-to-end methods still face semantic reasoning bottlenecks, as they struggle to fully comprehend high-level scene semantics, infer complex agent interactions, and adapt to dynamic task requirements. Moreover, their decision-making processes remain opaque, making it difficult to diagnose failure cases, especially in long-tail or unseen scenarios.

B. Large Vision Language Models

Large Language Models demonstrated strong emergent capabilities in in-context learning, instruction following, and reasoning [13], [14]. By training on vast amounts of Internet-scale data, these models acquire extensive world knowledge and exhibit strong adaptability across diverse tasks. Their success has also driven the rise of large VLMs, which extend these capabilities into cross-modal reasoning by integrating vision encoders with language models. State-of-the-art VLMs such as GPT-4V [15], LLaVA [16], and Qwen-VL [17] demonstrate strong visual understanding and multimodal reasoning in open-domain tasks. However, these models are primarily trained on static 2D images or videos and exhibit limited spatial reasoning in dynamic 3D driving environments. Moreover, VLMs are prone to hallucinations and generally over-confident but incorrect descriptions, which pose serious risks in safety-critical planning scenarios. Recently, Vision-Language Action models have emerged to directly predict actions from visual inputs, demonstrating strong performance in robotic manipulation tasks [18]. Currently, the application of such language-conditioned end-to-end action generation in autonomous driving remains underexplored. Yet, these methods are mostly limited to static setups and lack driving-specific 3D spatial design.

C. Vision Language Models in Autonomous Driving

VLMs have been applied to various autonomous driving tasks, including perception, scene description, synthetic data generation, and high-level decision-making [1]. These efforts aim to enhance interpretability, data efficiency, and instruction-following capabilities in driving models. We categorize recent works into 4 paradigms, as illustrated

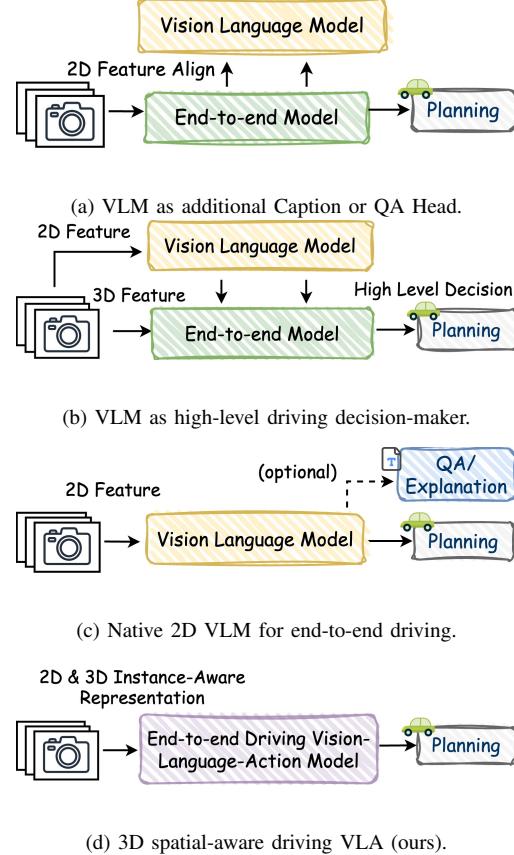


Fig. 2: Taxonomy of vision-language model applications in end-to-end autonomous driving.

in Figure 2. One line of research in Fig.2 (a) integrates language heads, such as captioning or question-answering modules, into driving models to enhance the interpretability [19]. The second category in Fig.2 (b) employs vision language models to generate high-level driving instructions, such as directional commands or abstract maneuvers, which are subsequently interpreted by separate planning modules into low-level controls [20]–[22]. It's also usually formed as a fast-slow dual system. This design allows VLMs to make independent semantic reasoning, but retains a separate module for end-to-end driving planning, making joint optimization challenging. The third line in Fig.2 (c) applies native VLMs with 2D visual tokens to produce

driving actions, and optionally scene captions or QA responses [23], [24]. These methods [25]–[27] process 2D images without explicit modeling of the instance, 3D spatial layout, and inter-agent interactions in the driving scene. It limits their spatial reasoning ability and understanding of agent dynamics in complex traffic environments. Recent studies [28] further indicate that such instance-agnostic approaches are more prone to hallucinate, often producing overconfident or semantically inconsistent text. In this work, we investigate how to extend 2D VLMs by explicitly modeling 3D instance-aware and spatial-aware scene representations into an end-to-end autonomous driving framework, as shown in Fig.2(d). Notably, we focus on fully differentiable end-to-end models in this work, while LLM-based agentic driving systems, such as [29], [30], fall outside the scope of our study.

III. OPENDRIVEVLA

The overall architecture of OpenDriveVLA is shown in Figure 1, with its multi-stage training process further detailed in Figure 3. OpenDriveVLA uses a pre-trained vision encoder to extract tokenized environmental representations from multi-view images. These visual tokens are then aligned into the textual domain through cross-modal learning. After alignment, it undergoes driving instruction tuning, followed by agent-ego-environment interaction modeling. Finally, OpenDriveVLA is trained end-to-end to predict the ego vehicle’s future trajectory, guided by the aligned visual-language tokens and driving instructions.

A. 3D Visual Environmental Perception

Recent VLM-based autonomous driving methods typically rely on pretrained 2D visual encoders [5], where visual token selection and attention are indirectly guided through language supervision. While effective in open-domain vision-language applications, this design lacks explicit 3D spatial grounding and structured instance-level attention, which can lead to severe hallucinations in safety-critical driving scenarios [31]. To mitigate this, OpenDriveVLA adopts a visual-centric query module, where the model first learns to focus on driving

relevant objects and map tokens through 3D vision tasks, ensuring reliable visual token proposal.

Specifically, given a set of multi-view images $I = \{I^i\}_{i=1}^N$, the visual module first extracts multi-scale 2D features from each image using a shared 2D backbone, denoted as f_{2D} . These 2D features are then aggregated across views and lifted into BEV space, producing the BEV feature f_{bev} . To obtain structured environmental representations, we adopt three visual query modules: Global Scene Sampler $\mathcal{Q}_{\text{scene}}$, Agent QueryTransformer $\mathcal{Q}_{\text{agent}}$, and Map QueryTransformer \mathcal{Q}_{map} . Each module extracts tokens focusing on a specific semantic aspect of the driving environment. Global Scene Sampler encodes the surrounding driving scene context from multi-view 2D features, producing the scene token $v_{\text{scene}} = \mathcal{Q}_{\text{scene}}(f_{2D})$. Agent QueryTransformer detects and tracks dynamic agents within the scene, extracting agent-centric tokens $\{v_{\text{agent}}^i\}_{i=1}^{N_a} = \mathcal{Q}_{\text{agent}}(f_{bev})$, where N_a denotes the number of detected agents. In parallel, Map QueryTransformer extracts static structural information, such as lane boundaries and drivable areas, forming the map token $v_{\text{map}} = \mathcal{Q}_{\text{map}}(f_{bev})$. Through vision-centric perception tasks, including 3D detection, tracking, and segmentation, the visual encoder produces structured environmental tokens that capture both dynamic agent behaviors and static map structures in a spatially grounded manner. The output tokens, denoted as $\mathbf{V}_{\text{env}} = \{v_{\text{scene}}, v_{\text{agent}}, v_{\text{map}}\}$, serve as visual environment representation of the subsequent stages.

B. Stage 1 - Hierarchical Vision-Language Alignment

To bridge the modality gap between the extracted visual tokens and the word embedding space of a pre-trained LLM, we adopt a hierarchical vision-language feature alignment strategy. Given the visual tokens extracted from the 3D visual perception module, we introduce three token-specific projectors $\{\Phi_{\text{scene}}, \Phi_{\text{agent}}, \Phi_{\text{map}}\}$. During training, each active agent query from the 3D detection and tracking task denoted as v_{agent}^i , is also matched to its corresponding ground-truth caption $\mathbf{X}_{\text{agent}}^i$. These captions provide detailed descriptions, including 2D

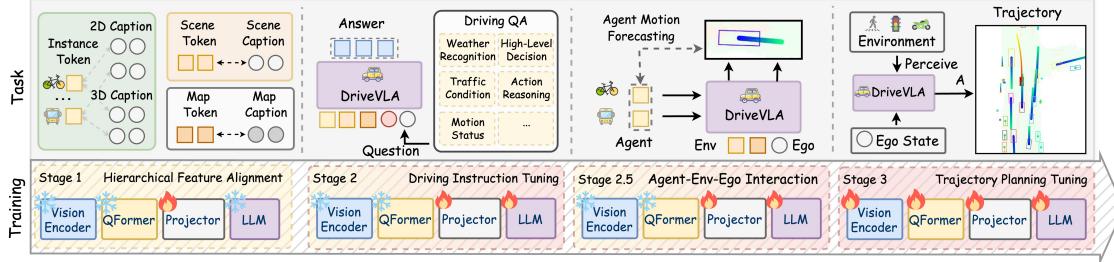


Fig. 3: Illustration of main training stages on OpenDriveVLA. Stage 1: Hierarchical Feature Alignment. Stage 2: Driving Instruction Tuning. Stage 2.5: Agent-Env-Ego Interaction Modeling. Stage 3: Trajectory Planning Tuning.

appearance descriptions and 3D spatial positions. For scene and map tokens, which encode holistic spatial context and static structural properties, a sample-wise alignment is applied, where each token is matched to a scene-level caption $\mathbf{X}_{\text{scene}}$ or \mathbf{X}_{map} . The scene token v_{scene} captures the global 2D environmental context, while the map token v_{map} encodes structural elements such as lane topology, road boundaries, and drivable areas. Each of these tokens is aligned to its corresponding caption, denoted as $\mathbf{X}_{\text{scene}}$ and \mathbf{X}_{map} . During this stage, both the visual encoder and LLM remain frozen to preserve pretrained semantics, with only the token-specific projectors being trainable. The forward alignment step is formulated as follows:

$$\hat{\mathbf{X}}_k = \text{LLM}(\Phi_k(v_k)), \quad k \in \{\text{scene, map}\} \quad (1)$$

$$\hat{\mathbf{X}}_{\text{agent}}^i = \text{LLM}(\Phi_{\text{agent}}(v_{\text{agent}}^i)), \quad i = 1, \dots, N_a \quad (2)$$

C. Stage 2 - Driving Instruction Tuning

We distill high-level driving knowledge into the model via supervised instruction tuning, enabling it to internalize semantic reasoning patterns during training. This avoids costly chain-of-thought (CoT) reasoning at inference time and balances planning efficacy with runtime efficiency.

During the tuning process, driving knowledge from the language domain is injected into the model using a curated driving instruction QA dataset. The dataset covers a wide range of driving-related reasoning, including perception understanding, motion prediction, attention allocation, action reasoning, and high-level decision-

making. By training on this diverse set of driving queries, OpenDriveVLA learns to contextualize the driving scene, follow commands, and generate semantically and behaviorally grounded planning decisions. We formulate the tuning data as instruction-response pairs $\{\mathbf{X}_{\text{input}}, \mathbf{X}_{\text{answer}}\}$, where $\mathbf{X}_{\text{input}} = (\mathbf{V}_{\text{env}}, \mathbf{S}_{\text{ego}}, \mathbf{X}_{\text{query}})$. Here, $\mathbf{X}_{\text{query}}$ denotes the driving-related question, and \mathbf{S}_{ego} encodes the textual ego vehicle state. Given this multimodal input, the LLM autoregressively learns to generate the target response. During instruction tuning, the visual encoder remains frozen while the token-specific projectors and the LLM are set to be trainable. The instruction prediction process is as:

$$\hat{\mathbf{X}}_{\text{answer}} = \text{LLM}(\mathbf{V}_{\text{env}}, \mathbf{S}_{\text{ego}}, \mathbf{X}_{\text{query}}) \quad (3)$$

D. Stage 2.5 - Agent Environment Ego Interaction

Reliable trajectory planning in autonomous driving necessitates a spatially grounded 3D representation of the environment. Beyond perception, it must also understand dynamic interactions between the ego vehicle and surrounding agents. Effective interaction modeling is essential to ensure that planned trajectories are both feasible and collision-free under real-world driving constraints. However, existing pre-trained LLMs lack an inherent inductive bias for spatial reasoning in 3D driving scenes, as they are predominantly trained on 2D vision-language and text-based datasets. We introduce a conditional agent trajectory forecasting task as an auxiliary objective, encouraging the model to

learn spatially grounded interaction priors. During this stage, OpenDriveVLA captures the underlying structure of multi-agent dynamics, enhancing its capability for scene-aware trajectory generation and improving decision-making in complex traffic scenarios.

Given scene and map tokens, as well as the ego vehicle state \mathbf{S}_{ego} , the LLM predicts the future motion of each detected agent based on its projected visual embedding $\Phi_{agent}(v_{agent}^i)$. The future motion of agent a_i is represented as a sequence of waypoints \mathcal{W}_a^i . The predicted trajectory is conditioned on the scene context, map structure, and ego vehicle state, enabling OpenDriveVLA to infer interaction-aware and spatially grounded motion sequences. The learning objective for the i -th agent is formulated as:

$$\max \prod_{t=1}^T p(w_t^i | w_{1:t-1}^i, \mathbf{V}_{env}, \mathbf{S}_{ego}, \Phi_{agent}(v_{agent}^i)) \quad (4)$$

This provides OpenDriveVLA with essential spatial priors, enabling it to bridge the gap between high-level semantic reasoning and physically grounded motion planning.

E. Stage 3 - End-to-end Trajectory Planning Tuning

In this stage, OpenDriveVLA predicts ego trajectories as discrete waypoint sequences within a short horizon, denoted as $\mathcal{W}_{ego} = \{w_1, w_2, \dots, w_T\}$. Each waypoint w_t represents the 2D coordinates (x_t, y_t) of the ego vehicle at time step t . The waypoints are tokenized into a sequence of discrete textual tokens for autoregressive generation in the LLM: $\mathcal{T}_{traj} = \text{Tokenizer}(\mathcal{W}_{ego})$. The generation process is then cast as a causal sequence prediction task, where each token is predicted in a causal manner, conditioned on the visual perception tokens \mathbf{V}_{env} , the ego state \mathbf{S}_{ego} , and the driving command \mathbf{X}_{dri} .

$$\hat{\mathcal{T}}_{traj} = \text{argmax}_{\mathbf{T}_{traj}} \prod_{t=1}^T p(w_t | w_{1:t-1}, \mathbf{V}_{env}, \mathbf{S}_{ego}, \mathbf{X}_{dri}) \quad (5)$$

The entire pipeline, including the 3D visual encoder, cross-modality projectors, and LLM, is

jointly optimized end-to-end during training, with the 2D encoder kept frozen. At inference, the model autoregressively generates the tokenized trajectory $\hat{\mathcal{T}}_{traj}$, which is then decoded back into numerical waypoints:

$$\hat{\mathcal{W}}_{ego} = \text{Decoder}(\hat{\mathcal{T}}_{traj}) \quad (6)$$

IV. EXPERIMENTS

A. Training Datasets

We curate the training data of OpenDriveVLA based on its distinct training phases, drawing from: TOD3Cap [43], nuCaption [44], nuScenesQA [45], nuX [19], and GPT-Driver [38]. We conduct experiments on nuScenes [46], following standard data split into training and validation sets. OpenDriveVLA is trained using the training set paired with corresponding QA captions, while the validation set is exclusively used for performance evaluation to ensure fair comparisons with prior works. The details of training data can be found in supplementary materials.

Hierarchical Vision-Language Alignment. For agent-level caption, we post-process data from [43], which provides the 2D visual description of individual objects. To further enhance spatial grounding, each object caption is augmented with its corresponding BEV coordinates, enabling the model to associate object attributes with precise spatial locations. For scene tokens, we process multi-view scene descriptions from [44], merging them into unified summaries that describe the driving environment across all camera views. For map tokens, structured language descriptions are derived from ground-truth annotations, translating map elements such as lane dividers, crosswalks, and road boundaries into descriptive text.

Driving Instruction Tuning. We adopt multiple instruction-oriented datasets derived from nuScenes to inject driving-specific knowledge into OpenDriveVLA. We unify several datasets into a standardized instruction-based QA format, including driving-related question-answer pairs collected from nuCaption [44], nuScenesQA [45], and nuX [19] dataset. Each QA pair is conditioned on structured environmental visual tokens and the ego vehicle

Method	ST-P3 metrics								UniAD metrics								LLM	Input		
	L2 (m) ↓				Collision (%) ↓				L2 (m) ↓				Collision (%) ↓							
	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.	1s	2s	3s	Avg.				
None-Autoregressive Methods																				
ST-P3 [32]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71	-	-	-	-	-	-	-	-	-	Visual		
VAD [33]	0.17	0.34	0.60	0.37	0.07	0.10	0.24	0.14	-	-	-	-	-	-	-	-	-	Visual		
Ego-MLP [34]	0.46	0.76	1.12	0.78	0.21	0.35	0.58	0.38	-	-	-	-	-	-	-	-	-	Ego		
UniAD [10]	0.44	0.67	0.96	0.69	0.04	0.08	0.23	0.12	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31	-	Visual		
InsightDrive [35]	0.23	0.41	0.68	0.44	0.09	0.10	0.27	0.15	0.30	0.72	1.41	0.81	0.08	0.15	0.84	0.36	-	Visual		
FF [9]	-	-	-	-	-	-	-	-	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43	-	LiDAR		
EO [36]	-	-	-	-	-	-	-	-	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33	-	LiDAR		
Autoregressive Methods																				
GPVL [37]	0.21	0.39	0.69	0.43	0.07	0.09	0.27	0.14	-	-	-	-	-	-	-	-	BERT	Textual		
DriveVLM [21]	0.18	0.34	0.68	0.40	0.10	0.22	0.45	0.27	-	-	-	-	-	-	-	-	Qwen-VL-7B	Visual		
GPT-Driver [38]	0.20	0.40	0.70	0.44	0.04	0.12	0.36	0.17	0.27	0.74	1.52	0.84	0.07	0.15	1.10	0.44	GPT-3.5	Textual		
RDA-Driver [39]	0.17	0.37	0.69	0.40	0.01	0.05	0.26	0.10	0.23	0.73	1.54	0.80	0.00	0.13	0.83	0.32	LLaVa-7B	Visual		
OminiDrive [29]	0.14	0.29	0.55	0.33	0.00	0.13	0.78	0.30	-	-	-	-	-	-	-	-	LLaVA-7B	Visual		
EMMA [40]	0.14	0.29	0.54	0.32	-	-	-	-	-	-	-	-	-	-	-	-	Gemini	Visual		
OpenEMMA [41]	1.45	3.21	3.76	2.81	-	-	-	-	-	-	-	-	-	-	-	-	Qwen-VL-7B	Visual		
DME-Driver [42]	-	-	-	-	-	-	-	-	0.45	0.91	1.58	0.98	0.05	0.28	0.55	0.29	LLaVa-7B	Visual		
OpenDriveVLA-0.5B (Ours)	0.15	0.32	0.57	0.35	0.01	0.06	0.20	0.09	0.21	0.60	1.22	0.68	0.00	0.15	0.63	0.26	Qwen2.5-0.5B	Visual		
OpenDriveVLA-3B (Ours)	0.14	0.30	0.55	0.33	0.02	0.07	0.22	0.10	0.19	0.58	1.24	0.67	0.02	0.18	0.70	0.30	Qwen2.5-3B	Visual		
OpenDriveVLA-7B (Ours)	0.15	0.31	0.55	0.33	0.01	0.08	0.21	0.10	0.20	0.58	1.21	0.66	0.00	0.22	0.55	0.25	Qwen2.5-7B	Visual		

TABLE I: Open-Loop planning performance comparison of different driving models, including both autoregressive methods and non-autoregressive methods. OpenDriveVLA shows powerful planning ability and achieves best-in-class results among open-source models, even with the 0.5B version. We refer to the result summary from [35], [37]–[39].

state, ensuring consistency across different data sources. This multimodal instruction tuning process allows OpenDriveVLA to effectively ground language understanding into both environmental perception and scene understanding, bridging perception, reasoning, and action within the language space.

Motion Forecasting and Trajectory Prediction. We formulate both agent motion forecasting and ego trajectory planning in the ego system, where the model directly predicts future displacements within each entity’s local coordinate frame relative to the ego vehicle for planning and relative to each agent for forecasting. This formulation captures motion dynamics in a spatially consistent manner across all entities. Following [38], the ego vehicle state is encoded as textual input to ensure ego awareness throughout the training process. Both tasks predict 3-second future trajectories, sampled at 0.5-second intervals, resulting in 6 waypoints per trajectory.

B. Evaluations

We evaluate OpenDriveVLA on the open-loop planning task of nuScenes benchmark, where the model is reported under both ST-P3 [32] and

UniAD [10] settings. The evaluation metrics include L2 displacement errors at 1, 2, and 3 seconds, along with the average collision rate over the prediction horizon. To further assess the scene understanding ability of OpenDriveVLA, we report its QA prediction performance on three driving visual question answering (VQA) datasets directly after the driving instruction tuning stage, i.e., [44], nuScenesQA [45], and nuX [19]. The VQA evaluation results adopt standard NLG metrics, including BLEU, METEOR, CIDEr, BERT-Score, etc.

C. Implementation Details

The 3D visual perception module in OpenDriveVLA follows the vision-centric design from [10], using a ResNet-101 backbone for 2D feature extraction. The perception backbone is pre-trained via multi-task learning on 3D object detection, object tracking, and map segmentation. The resulting BEV feature map has a spatial resolution of 200×200 . To construct a unified scene representation, the global SceneSampler applies 2D adaptive pooling to each camera view, subsequently concatenating the pooled multi-view features into a global scene token. Agent and map tokens are extracted from

Method	nu-Caption					nuScenes-QA							
	BL-1	BL-2	BL-3	BL-4	BERT-S	Ext	Cnt	Obj	Sts	Cmp	H0	H1	Acc
Mini-GPT4 [47]	15.0	6.8	3.7	2.6	84.4	-	-	-	-	-	-	-	-
Instruct-BLIP [48]	18.7	13.4	7.4	5.2	85.9	-	-	-	-	-	-	-	-
LLaMA-AdapV2 [49]	30.2	17.3	10.4	7.5	86.5	19.3	2.7	7.6	10.8	1.6	15.1	4.8	9.6
LLaVA1.5 [50]	20.0	12.1	8.6	5.4	85.0	45.8	7.7	7.8	9.0	52.1	25.7	41.5	26.2
LiDAR-LLM [44]	41.0	30.0	23.4	19.3	91.3	74.5	15.0	37.8	45.9	57.8	-	-	48.6
BEVDet+BTUD [45]	-	-	-	-	-	83.7	20.9	48.8	52.0	67.7	-	-	57.0
OpenDriveVLA-0.5B (Ours)	47.2	35.8	29.4	25.2	91.9	83.9	22.0	50.2	57.0	68.4	62.3	56.5	58.4
OpenDriveVLA-3B (Ours)	48.3	36.9	30.3	26.1	92.0	84.0	22.3	50.3	56.9	68.5	62.6	56.5	58.5
OpenDriveVLA-7B (Ours)	49.6	38.3	31.9	27.6	92.2	84.2	22.7	49.6	54.5	68.8	62.4	56.1	58.2

TABLE II: Performance on nu-Caption [44] and nuScenes-QA [45]. BL-1/2/3/4: BLEU scores. QA metrics report accuracy on five question types: Existence, Counting, Object, Status, and Comparison.

the final layer of their respective QueryTransformer modules. Each token type is then mapped into the language space using a separate two-layer MLP with GeLU activation. We adopt Qwen 2.5-Instruct [14] as the pre-trained LLM, which undergoes full parameter tuning during training. Training is performed on 4 NVIDIA H100 GPUs with a batch size of 1, completed in approximately two days. We freeze the 2D backbone during stage 3. During inference, we set the decoding temperature to 0 to ensure deterministic trajectory generation. See supplementary material for detailed training configurations.

D. Main Results

Open Loop Trajectory Planning. We evaluate OpenDriveVLA on the open-loop trajectory planning task using both ST-P3 and UniAD metrics, ensuring comprehensive performance assessment across spatial accuracy and collision avoidance. As shown in Table I, OpenDriveVLA achieves state-of-the-art performance across both settings. Specifically, both 3B and 7B version models achieve an average L2 error of 0.33m under ST-P3 metrics, outperforming prior autoregressive language models [21], [38]. On the UniAD metrics, OpenDriveVLA-7B also achieves great performance with an average L2 error of 0.66m. Notably, despite significantly fewer parameters, the 0.5B version still outperforms prior models obviously.

Driving Question Answering. We access OpenDriveVLA on the driving VQA task across three nuScenes-based datasets (Table II, Table III), reporting results after the second stage of training. Open-

TABLE III: Performance comparison of OpenDriveVLA on the Nu-X dataset [19].

Models	CIDER	BL-4	METEOR	ROUGE-L
Hint-UniAD [19]	21.7	4.2	12.7	27.0
Hint-VAD [19]	22.4	4.2	13.2	27.6
GPT-4o [24]	19.0	4.0	10.3	24.9
Gemini 1.5 [51]	17.6	3.4	9.3	23.4
Vote2CapDETR [52]	15.3	2.6	10.9	24.2
TOD ³ Cap [43]	14.5	2.5	10.5	23.5
OpenDriveVLA				
0.5B (Ours)	32.3	5.4	12.5	27.9
3B (Ours)	25.5	4.3	12.8	27.8
7B (Ours)	26.2	4.5	12.8	27.4

DriveVLA reaches best-in-class performance across all three datasets, consistently outperforming previous language-enhanced driving models and general-purpose multimodal baselines among most metrics. On nuCaption dataset, it achieves the best captioning performance among all evaluated models, outperforming both general VLMs LLaVA1.5 [50] and Mini-GPT4 [47], as well as autonomous driving-specific models such as LiDAR-LLM [44]. For nuScenesQA dataset, OpenDriveVLA also achieves strong performance. Compared to models that directly fuse BEV features with language models such as BEVDet+BTUD [45], it demonstrates clear advantages in object and status-related questions, which highlights the benefit of its spatially grounded visual-language alignment. Notably, the 0.5B version outperforms even the larger 7B on the Nu-X dataset, which shows its powerful scene-understanding ability even with lightweight LLMs.

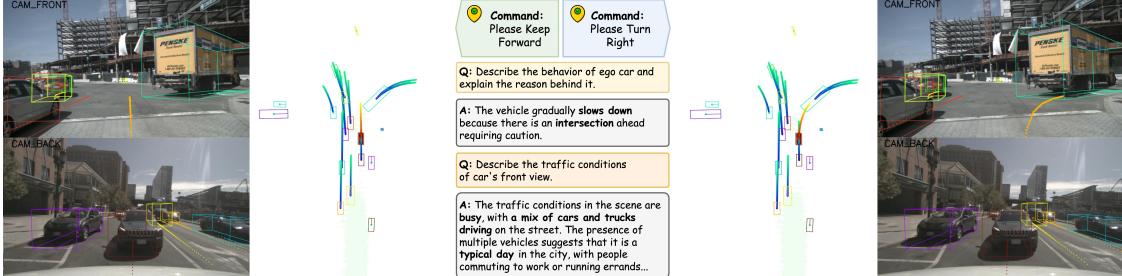


Fig. 4: Visualization of OpenDriveVLA-7B planning actions under original dataset instruction to keep forward (left) and modified instruction to turn right (right). The QA prediction showcases (middle) are from results reported in Table II and Table III. The agent motion prediction results are visualized after the agent-env-ego interaction stage.

E. Ablation Study

We conduct ablation studies to evaluate the impact of input modalities and our multi-stage training strategy on OpenDriveVLA’s performance. Additionally, we qualitatively assess the model’s ability to follow diverse driving commands.

Visu	Ego	Hist	Cmd	Avg. Collision (%) ↓		Avg. L2 (m) ↓	
				UniAD	ST-P3	UniAD	ST-P3
✓		✓	✓	0.77	0.24	1.34	0.75
✓	✓		✓	1.14	0.49	1.30	0.75
✓	✓	✓	✓	0.29	0.10	0.77	0.39
✓	✓	✓		0.33	0.13	0.80	0.40
✓	✓	✓	✓	0.26	0.09	0.68	0.35

TABLE IV: Ablation study on the effect of different input combinations on OpenDriveVLA-0.5B.

Effect of Input Modalities. We investigate how individual input components contribute to trajectory planning. Table IV presents the results of ablating visual perception, ego state, historical trajectory, and high-level language commands. The inclusion of visual inputs significantly boosts overall performance. Adding textual commands and historical information further improves the predictions, emphasizing the value of semantic intent and temporal context. Notably, ego-state features play a critical role in nuScenes open-loop benchmark, consistent with prior findings [53].

Effect of Multi-Stage Training Strategy. We evaluate the contribution of each training phase in our staged pipeline incrementally. As shown in Table V, each additional stage consistently improves performance, with the most notable reductions in collision rate observed after Hierarchical Vision-Language

Alignment and Agent-Environment-Ego Interaction Modeling. These improvements highlight the effectiveness of cross-modal grounding and interaction-aware reasoning in enhancing safety-critical planning behavior.

Training Stage	Avg. Collision (%) ↓			Avg. L2 (m) ↓		
	1	2	2.5	3	UniAD	ST-P3
✓				✓	0.37	0.13
✓		✓		✓	0.32	0.12
✓	✓			✓	0.31	0.11
✓	✓	✓	✓	✓	0.26	0.09

TABLE V: Ablation study on the effect of multi-stage training of 0.5B model. Stage 1, 2, 2.5, and 3 correspond to hierarchical feature alignment, driving instruction tuning, Agent-Env-Ego modeling, and trajectory tuning, respectively.

Effect of Driving Command. Figure 4 presents the qualitative comparison at an intersection under two different driver instructions: keep forward and turn right, with the right turn as the ground truth. OpenDriveVLA accurately adapts its plan to the given command while maintaining context-aware and environment-consistent behavior, demonstrating robust command-following and generalization in complex scenes. In addition, we visualize the QA predictions for the same scene, showcasing the model’s ability to reason over decision-making and traffic scene understanding.

V. CONCLUSION

In this work, we present OpenDriveVLA, a scalable vision-language action model designed for end-to-end autonomous driving. Built upon pre-trained large language models, OpenDriveVLA generates

3D spatially grounded and semantically consistent driving actions from multimodal inputs. We introduce a hierarchical vision-language feature alignment module and realize agent-env-ego interaction in LLM to enable fine-grained spatial reasoning and dynamic scene understanding. Through multi-stage training paradigm, OpenDriveVLA achieves state-of-the-art performance in open-loop planning and driving-related question answering. Extensive evaluations on nuScenes dataset show its superior trajectory planning capability compared to existing approaches. Our work demonstrates the feasibility of a scalable vision-language-driven approach for autonomous driving and highlights the potential of large language models as a foundation for end-to-end driving action systems.

REFERENCES

- [1] X. Zhou, M. Liu, E. Yurtsever, B. L. Zagar, W. Zimmer, H. Cao, and A. C. Knoll, "Vision language models in autonomous driving: A survey and outlook," *IEEE Transactions on Intelligent Vehicles*, pp. 1–20, 2024.
- [2] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [3] M. Liu, E. Yurtsever, J. Fossaert, X. Zhou, W. Zimmer, Y. Cui, B. L. Zagar, and A. C. Knoll, "A survey on autonomous driving datasets: Statistics, annotation quality, and a future outlook," *IEEE Transactions on Intelligent Vehicles*, pp. 1–29, 2024.
- [4] X. Zhou and A. C. Knoll, "Gpt-4v as traffic assistant: An in-depth look at vision language model on complex traffic events," 2024. [Online]. Available: <https://arxiv.org/abs/2402.02205>
- [5] X. Zhai, B. Mustafa, A. Kolesnikov, and L. B. et al., "Sigmoid loss for language image pre-training," 2023. [Online]. Available: <https://arxiv.org/abs/2303.15343>
- [6] H. Liu, W. Xue, Y. Chen, D. Chen, X. Zhao, K. Wang, L. Hou, R. Li, and W. Peng, "A survey on hallucination in large vision-language models," 2024. [Online]. Available: <https://arxiv.org/abs/2402.00253>
- [7] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from nnbsp;multi-camera images vinnbsp;spatiotemporal transformers," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 1–18. [Online]. Available: https://doi.org/10.1007/978-3-031-20077-9_1
- [8] S. Zhang, Y. Zhai, J. Mei, and Y. Hu, "Fusionocc: Multi-modal fusion for 3d occupancy prediction," in *Proceedings of the 32nd ACM International Conference on Multimedia*, ser. MM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 787–796. [Online]. Available: <https://doi.org/10.1145/3664647.3681293>
- [9] P. Hu, A. Huang, J. Dolan, D. Held, and D. Ramanan, "Safe local motion planning with self-supervised freespace forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 732–12 741.
- [10] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.
- [11] B. Liao, S. Chen, H. Yin, B. Jiang, C. Wang, S. Yan, X. Zhang, X. Li, Y. Zhang, Q. Zhang, and X. Wang, "Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving," *arXiv preprint arXiv:2411.15139*, 2024.
- [12] X. Jia, J. You, Z. Zhang, and J. Yan, "Drivetransformer: Unified transformer for scalable end-to-end autonomous driving," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=M42KR4W9P5>
- [13] H. Touvron, T. Lavril, G. Izacard, and e. a. Xavier Martinet, "Llama: Open and efficient foundation language models," 2023. [Online]. Available: <https://arxiv.org/abs/2302.13971>
- [14] A. Yang, B. Yang, and B. Z. et al., "Qwen2.5 technical report," *arXiv preprint arXiv:2412.15115*, 2024.
- [15] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, and J. A. et al., "Gpt-4 technical report," 2024. [Online]. Available: <https://arxiv.org/abs/2303.08774>
- [16] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," January 2024. [Online]. Available: <https://llava-vl.github.io/blog/2024-01-30-llava-next/>
- [17] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond," *arXiv preprint arXiv:2308.12966*, 2023.
- [18] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Santek, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.
- [19] K. Ding, B. Chen, Y. Su, H.-a. Gao, B. Jin, C. Sima, X. Li, W. Zhang, P. Barsch, and H. e. a. Li, "Hint-ad: Holistically aligned interpretability in end-to-end autonomous driving," in *8th Annual Conference on Robot Learning*, 2024.
- [20] B. Jiang, S. Chen, B. Liao, X. Zhang, W. Yin, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Senna: Bridging large vision-language models and end-to-end autonomous driving," 2024. [Online]. Available: <https://arxiv.org/abs/2410.22313>
- [21] X. Tian, J. Gu, B. Li, Y. Liu, Y. Wang, Z. Zhao, K. Zhan, P. Jia, X. Lang, and H. Zhao, "DriveVLM: The convergence of autonomous driving and large vision-language models," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: <https://openreview.net/forum?id=928V4Umlys>
- [22] W. Wang, J. Xie, C. Hu, H. Zou, J. Fan, W. Tong, Y. Wen, S. Wu, H. Deng, Z. Li et al., "Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving," *arXiv preprint arXiv:2312.09245*, 2023.
- [23] B. Jin, X. Liu, Y. Zheng, P. Li, and H. Z. et al., "Adapt: Action-aware driving caption transformer," 2023. [Online]. Available: <https://arxiv.org/abs/2302.00673>
- [24] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8186–8193, 2024.
- [25] J. Mei, Y. Ma, X. Yang, L. Wen, X. Cai, X. Li, D. Fu, B. Zhang, P. Cai, M. Dou, B. Shi, L. He, Y. Liu, and Y. Qiao, "Continuously learning, adapting, and improving: A dual-process approach to autonomous driving," 2024. [Online]. Available: <https://arxiv.org/abs/2405.15324>
- [26] J. Zhang, Z. Huang, A. Ray, and E. Ohn-Bar, "Feedback-guided autonomous driving," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 15 000–15 011.

- [27] H. Fu, D. Zhang, Z. Zhao, J. Cui, D. Liang, C. Zhang, D. Zhang, H. Xie, B. Wang, and X. Bai, “Orion: A holistic end-to-end autonomous driving framework by vision-language instructed action generation,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.19755>
- [28] A. Favero, L. Zancato, M. Trager, S. Choudhary, P. Perera, A. Achille, A. Swaminathan, and S. Soatto, “Multi-modal hallucination control by visual information grounding,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.14003>
- [29] S. Wang, Z. Yu, X. Jiang, S. Lan, M. Shi, N. Chang, J. Kautz, Y. Li, and J. M. Alvarez, “OmniDrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning,” *arXiv:2405.01533*, 2024.
- [30] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, P. Luo, A. Geiger, and H. Li, “Drivelm: Driving with graph visual question answering,” *arXiv preprint arXiv:2312.14150*, 2023.
- [31] S. Xie, L. Kong, Y. Dong, C. Sima, and W. Z. et al., “Are vlms ready for autonomous driving? an empirical study from the reliability, data, and metric perspectives,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.04003>
- [32] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, “St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning,” in *European Conference on Computer Vision (ECCV)*, 2022.
- [33] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, “Vad: Vectorized scene representation for efficient autonomous driving,” *ICCV*, 2023.
- [34] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, “Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.10430>
- [35] R. Song, X. Guo, H. Wu, Q. Wei, and L. Chen, “Insightdrive: Insight scene representation for end-to-end autonomous driving,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.13047>
- [36] T. Khurana, P. Hu, A. Dave, J. Ziglar, D. Held, and D. Ramanan, “Differentiable raycasting for self-supervised occupancy forecasting,” in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 353–369. [Online]. Available: https://doi.org/10.1007/978-3-031-19839-7_21
- [37] T. Li, H. Wang, X. Li, W. Liao, T. He, and P. Peng, “Generative planning with 3d-vision language pre-training for end-to-end autonomous driving,” 2025. [Online]. Available: <https://arxiv.org/abs/2501.08861>
- [38] J. Mao, Y. Qian, J. Ye, H. Zhao, and Y. Wang, “Gpt-driver: Learning to drive with gpt,” 2023. [Online]. Available: <https://arxiv.org/abs/2310.01415>
- [39] Z. Huang, T. Tang, S. Chen, S. Lin, Z. Jie, L. Ma, G. Wang, and X. Liang, “Making large language models better planners with reasoning-decision alignment,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.13890>
- [40] J.-J. Hwang, R. Xu, H. Lin, W.-C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, Y. Zhou, J. Guo, D. Anguelov, and M. Tan, “Emma: End-to-end multimodal model for autonomous driving,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.23262>
- [41] S. Xing, C. Qian, Y. Wang, H. Hua, K. Tian, Y. Zhou, and Z. Tu, “Openemma: Open-source multimodal model for end-to-end autonomous driving,” 2025. [Online]. Available: <https://arxiv.org/abs/2412.15208>
- [42] W. Han, D. Guo, C.-Z. Xu, and J. Shen, “Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving,” 2024. [Online]. Available: <https://arxiv.org/abs/2401.03641>
- [43] B. Jin, Y. Zheng, P. Li, W. Li, Y. Zheng, and S. e. a. Hu, “Tod3cap: Towards 3d dense captioning,” in *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29 – October 4, 2024, Proceedings, Part XVIII*. Berlin, Heidelberg: Springer-Verlag, 2024, p. 367–384. [Online]. Available: https://doi.org/10.1007/978-3-031-72649-1_21
- [44] S. Yang, J. Liu, R. Zhang, M. Pan, Z. Guo, X. Li, Z. Chen, P. Gao, Y. Guo, and S. Zhang, “Lidar-llm: Exploring the potential of large language models for 3d lidar understanding,” 2023. [Online]. Available: <https://arxiv.org/abs/2312.14074>
- [45] T. Qian, J. Chen, L. Zhuo, Y. Jiao, and Y.-G. Jiang, “Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario,” *arXiv preprint arXiv:2305.14836*, 2023.
- [46] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [47] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhosseiny, “MiniGPT-4: Enhancing vision-language understanding with advanced large language models,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=1tZbq88f27>
- [48] W. Dai and J. L. et al., “InstructBLIP: Towards general-purpose vision-language models with instruction tuning,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: <https://openreview.net/forum?id=vvoWPYqZIA>
- [49] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, and P. e. a. Lu, “Llama-adapter v2: Parameter-efficient visual instruction model,” *arXiv preprint arXiv:2304.15010*, 2023.
- [50] H. Liu, C. Li, Y. Li, and Y. J. Lee, “Improved baselines with visual instruction tuning,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.03744>
- [51] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, S. Mariooryad, Y. Ding, X. Geng, F. Alcober, R. Frostig, M. Omernick, and L. W. et al., “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.05530>
- [52] S. Chen, H. Zhu, X. Chen, Y. Lei, G. Yu, and T. Chen, “End-to-end 3d dense captioning with vote2cap-detr,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11124–11133.
- [53] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, “Is ego status all you need for open-loop end-to-end autonomous driving?” in *2024 IEEE/CVF Conference*

- on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 864–14 873.
- [54] L. Xiao, X. Yang, X. Lan, Y. Wang, and C. Xu, “Towards visual grounding: A survey,” 2024. [Online]. Available: <https://arxiv.org/abs/2412.20206>
 - [55] S. Wang, D. Kim, A. Taalimi, C. Sun, and W. Kuo, “Learning visual grounding from generative vision and language model,” in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 8057–8067.
 - [56] X. Zhou, K. Larintzakis, H. Guo, W. Zimmer, M. Liu, H. Cao, J. Zhang, V. Lakshminarasimhan, L. Strand, and A. Knoll, “TUMTraf videoQA: Dataset and benchmark for unified spatio-temporal video understanding in traffic scenes,” in *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. [Online]. Available: <https://openreview.net/forum?id=Yfoi5O68rf>
 - [57] U. contributors, “Planning-oriented autonomous driving,” <https://github.com/OpenDriveLab/UniAD>, 2023.
 - [58] Z. Li, W. Wang, E. Xie, Z. Yu, A. Anandkumar, J. M. Alvarez, P. Luo, and T. Lu, “Panoptic segformer: Delving deeper into panoptic segmentation with transformers,” 2022. [Online]. Available: <https://arxiv.org/abs/2109.03814>
 - [59] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles,” 2022. [Online]. Available: <https://arxiv.org/abs/2106.11810>
 - [60] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan, “Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving,” in *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.
 - [61] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, A. Geiger, and K. Chitta, “Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.15349>

Appendix-OpenDriveVLA: Towards End-to-End Autonomous Driving with Large Vision Language Action Model

VI. IMPLEMENTATION DETAILS

A. Model Details

1) *Vision Encoder*: To obtain accurate instance-level token representations, one option is to adopt language-guided visual grounding tasks [54], [55], where visual regions are aligned with textual descriptions with cross-modality supervision. However, such supervision is often ambiguous and imprecise, especially in complex traffic environments where spatial accuracy is essential. This ambiguity arises from the inherent subjectivity of textual annotations and the weak spatial constraints in general vision-language datasets [56]. Moreover, they typically lack consistent object definitions and fail to capture structured scene semantics, making them suboptimal for autonomous driving applications that demand precise object localization and spatial understanding.

Motivated by these, we adopt a vision-centric pretraining visual encoder and instance token decoder based on high-quality autonomous driving datasets. The overall architecture of the vision encoder and token extraction modules is illustrated in Figure 5.

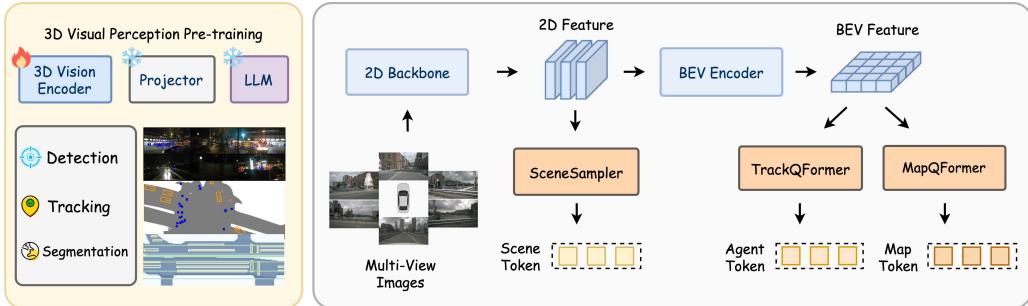


Fig. 5: Overview of the visual encoder pretraining stage. The 3D vision-centric pretraining tasks include 3D object detection, tracking, and BEV panoptic segmentation. The extracted features are projected into scene, agent, and map tokens for downstream vision-language alignment and planning.

We adopt the perception training stage introduced in UniAD [10], [57] to produce a robust visual encoder for the traffic environment. It is trained via vision-centric 3D perception tasks, including 3D object detection, tracking, and BEV panoptic segmentation. Compared to grounding via general vision-language grounding tasks, vision-centric tasks offer more structured and semantically instance-level supervision, which is critical for autonomous driving.

The visual encoder follows a multi-view, query-based architecture, where multi-view images are processed by a shared ResNet-101 backbone with FPN to extract multi-scale 2D features, which are aggregated into BEV representation using BEVFormer [7]. A detailed summary of the visual module setting in OpenDriveVLA is provided in Table VI.

The vision-centric task is supervised using a combination of detection, tracking and segmentation losses following original setting:

$$\mathcal{L}_{\text{vis}} = \mathcal{L}_{\text{track}} + \mathcal{L}_{\text{map}} \quad (13)$$

Module	Input	Output	Architecture	Configuration
2D Backbone	Multi-view images	2D features	ResNet-101 + FPN	Output strides: {1/8, 1/16, 1/32}
BEV Encoder	2D features	BEV feature map	BEVFormer (6-layer encoder)	Hidden dim: 256; BEV: 200 × 200
Scene Sampler	2D features	Global scene token	Adaptive max pooling	Output size: (6, 3, 5)
Agent Query Transformer	BEV features	Agent tokens	TrackQFormer (6-layer decoder)	900 queries
Map Query Transformer	BEV features	Map tokens	MapQFormer (6-layer decoder)	300 queries (3 thing + 1 stuff)

TABLE VI: Summary of components in the 3D visual perception module.

The tracking loss $\mathcal{L}_{\text{track}}$ combines focal classification loss and L1 bounding box regression, optimized via a Hungarian matching algorithm. The map segmentation loss \mathcal{L}_{map} includes classification, bounding box, mask, and IoU losses over both thing and stuff classes. Map supervision follows Panoptic SegFormer [58]. This vision-centric supervision provides dense and structured grounding signals, enabling the encoder to learn spatially precise and semantically aligned features.

2) *Structural Token Extraction*: Based on multi-scale 2D features \mathbf{F}_{2D} from multi-view images and aggregated BEV representation \mathbf{F}_{bev} , OpenDriveVLA retains the structured perception architecture of UniAD and extracts semantically meaningful tokens through three query-based modules. These modules encode complementary aspects of the driving scene and serve as a compact interface for downstream alignment and planning.

Agent QueryTransformer: TrackQFormer decodes dynamic object-level semantics from the BEV feature map $f_{bev} \in \mathbb{R}^{200 \times 200 \times D}$ using learnable queries. Each token encodes an individual agent’s spatial location, category, and motion trajectory. We retain the top- N_a tokens filtered by confidence threshold: $\{v_{\text{agent}}^i\}_{i=1}^{N_a} = \mathcal{Q}_{\text{agent}}(f_{bev})$. To improve efficiency and robustness, we filter out low-confidence agent predictions based on detection scores. This selective mechanism reduces the number of input visual tokens, leading to faster downstream processing and a more focused representation. Moreover, it helps mitigate hallucination by suppressing uncertain or noisy detections from the vision input source.

Map QueryTransformer: MapQFormer focuses on extracting static structural elements from the BEV features via separate decoder heads for thing and stuff categories. It produces up to N_m map tokens, each encoding elements such as lane dividers, road boundaries, and drivable areas: $\{v_{\text{map}}^j\}_{j=1}^{N_m} = \mathcal{Q}_{\text{map}}(f_{bev})$.

Global Scene Sampler: To capture holistic scene-level context, we apply adaptive max pooling over the 2D features $\mathbf{F}_{2D} \in \mathbb{R}^{6 \times 256 \times H \times W}$ from six camera views, compressing them into (3×5) spatial grids per view. These tokens encode global contextual information, including weather, lighting, scene layout, and traffic flow, which are difficult to infer solely from BEV-based modules. By abstracting high-level semantics from raw visual features, the scene tokens serve as a complementary source of information to the agent and map tokens, providing redundancy and enhancing robustness for downstream alignment and planning: $v_{\text{scene}} = \mathcal{Q}_{\text{scene}}(\mathbf{f}_{2D}) \in \mathbb{R}^{90 \times D}$.

Together, three token extractors provide a compact and structural encoding of the traffic environment, covering dynamic agents, static infrastructure, and global context: $V_{\text{env}} = \{V_{\text{scene}}, V_{\text{agent}}, V_{\text{map}}\}$.

3) *Large Language Action Module*: We adopt Qwen2.5-Instruct [14] as the pre-trained LLM backbone for generating structured driving actions. To balance model capacity and computational efficiency, we evaluate three variants with 0.5B, 3B, and 7B parameters. Our implementation builds upon the LLava NeXT framework [16], which enables the language model to perform structured cross-modality reasoning and autoregressive action generation.

To support structured multimodal OpenDriveVLA input, we extend the Qwen2.5 tokenizer with a set of special tokens and token indices, as summarized in Table VII. These tokens formulate inputs into discrete semantic segments, allowing the LLM to differentiate between scene-level context, dynamic agents, static

Token / Index	Description
<i>Placeholder Token Indices</i>	
IMAGE_TOKEN_INDEX	Placeholder for global image token
SCENE_TOKEN_INDEX	Placeholder for scene-level visual token
TRACK_TOKEN_INDEX	Placeholder for agent-level track token
MAP_TOKEN_INDEX	Placeholder for map-level token
OBJECT_TOKEN_INDEX	Reserved placeholder for object token
<i>High-Level Markers</i>	
<SCENE>, <TRACK>, <MAP>	Denote visual token segments
<EGO>	Textualized ego-vehicle state
<COMMAND>	Driving command or query string
<trajectory>	Start of autoregressive trajectory output
<i>Token Delimiters (Start / End Wrappers)</i>	
<scene_start>, <scene_end>	Scene token span delimiters
<track_start>, <track_end>	Track token span delimiters
<map_start>, <map_end>	Map token span delimiters
<ego_start>, <ego_end>	Ego state string delimiters
<command_start>, <command_end>	Driving command delimiters
<traj_start>, <traj_end>	Generated trajectory delimiters
<i>Optional QA Format Tokens</i>	
<question_start>, <question_end>	Input question delimiters
<answer_start>, <answer_end>	Output answer delimiters

TABLE VII: Extended special tokens and placeholder indices used in OpenDriveVLA’s LLM tokenizer.

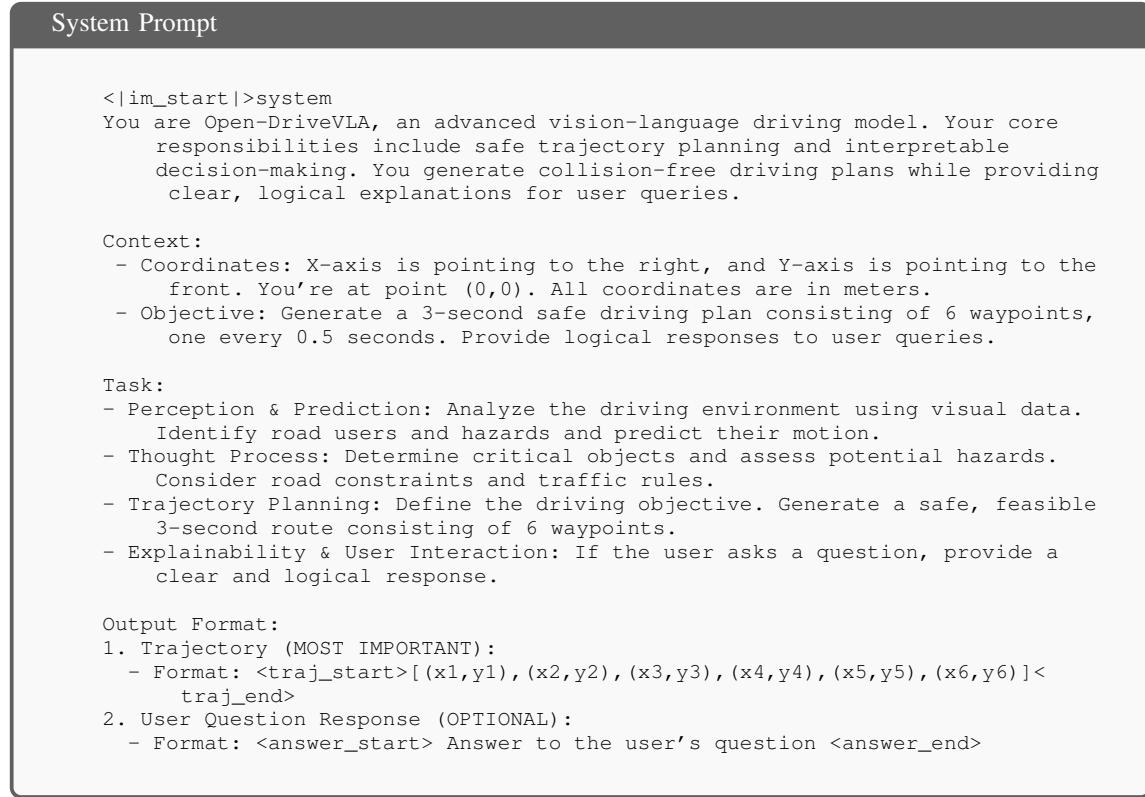
maps, ego vehicle state, and high-level driving instructions. The input sequence passed to the LLM is structured as:

```
<SYSTEM><SCENE><TRACK><MAP><EGO><COMMAND>
```

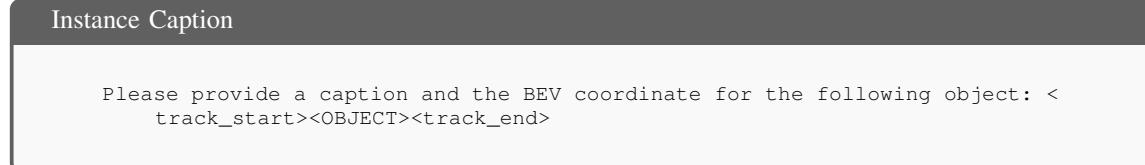
During input construction, <SCENE>, <TRACK>, and <MAP> are replaced with projected visual tokens, while <EGO> and <COMMAND> are filled with formatted textual strings.

B. Prompting Techniques

1) *System Prompt:* The system prompt follows the BEV coordinate conventions introduced in GPT Driver [38], while specifying the model role, driving objectives, and output format. It guides the LLM to perform perception, reasoning, and trajectory generation in a unified manner, with optional user interaction. The system prompt is prepended to all inputs during both training and inference to ensure consistent task framing and instruction following.



2) *Prompts for Hierarchical Feature Alignment:* Each type of visual token is associated with a separate captioning prompt. These prompts instruct the LLM to generate textual descriptions based on the corresponding visual input segment.



Map Caption

```
Please provide a caption for the following map: <map_start><MAP><map_end>
```

Scene Caption

```
Please provide a caption for the following scene: <scene_start><SCENE><scene_end>
```

3) *Prompts for Driving Question Answering:* All VQA datasets adopt a consistent structured prompt format, integrating scene, agent, map, ego state, and historical trajectory components. Question Placeholders are replaced with dataset-specific content during training and evaluation.

Driving Question Answering Prompt

```
Scene information: <scene_start><SCENE><scene_end>\nObject-wise tracking  
information: <track_start><TRACK><track_end>\nMap information: <  
map_start><MAP><map_end>\nEgo states: - Velocity (vx,vy): <Velocity  
Placeholder> - Heading Angular Velocity (v_yaw): <Angular Velocity  
Placeholder> - Acceleration (ax,ay): <Acceleration Placeholder> - Can  
Bus: <Can Bus Placeholder> - Heading Speed: <Speed Placeholder> -  
Steering: <Steering Placeholder>\nHistorical trajectory (last 2 seconds)  
: <Trajectory Placeholder> \nPlease answer the following question: <  
Question Placeholder>
```

4) *Prompts for Agent-Env-Ego Interaction:* This prompt is used in Stage 2.5-modeling agent-environment-ego interactions. The input includes structured visual context, historical ego trajectory, and a queried target object, and the model is instructed to predict the future motion of the agent.

Driving Question Answering Prompt

```
<scene_start><SCENE><scene_end>\nObject-wise tracking information: <  
track_start><TRACK><track_end>\nMap information: <map_start><MAP><  
map_end>\nEgo Vehicle Token: <trajectory>\nPlease predict relative  
motion trajectory for the following object: <track_start><OBJECT><  
track_end>
```

5) *Prompts for Trajectory Planning Tuning:* This prompt is used in the training stage 3 for trajectory planning tuning, where the model is supervised to generate a 3-second driving plan based on structured multi-modal context.

Driving Question Answering Prompt

```
Scene information: <scene_start><SCENE><scene_end>\nObject-wise tracking
information: <track_start><TRACK><track_end>\nMap information: <
map_start><MAP><map_end>\nEgo states: - Velocity (vx,vy): <Velocity
Placeholder> - Heading Angular Velocity (v_yaw): <Angular Velocity
Placeholder> - Acceleration (ax,ay): <Acceleration Placeholder> - Can
Bus: <Can Bus Placeholder> - Heading Speed: <Speed Placeholder> -
Steering: <Steering Placeholder>\nHistorical trajectory (last 2 seconds)
: <Trajectory Placeholder>\nMission goal: <Command Placeholder>\
nPlanning trajectory: <trajectory>
```

C. Training and Inference Details

1) *Training Configuration*.: OpenDriveVLA is trained on 4 NVIDIA H100 GPUs with a per-GPU batch size of 1. The full training process takes approximately two days for the 0.5B variant. All training stages use mixed-precision (bf16) and gradient checkpointing to improve memory efficiency and speed. The 2D vision backbone is frozen during the final end-to-end stage. LLM parameters are fully tuned unless specified otherwise.

		Stage 1	Stage 2	Stage 2.5	Stage 3
Train	Tunable parts	projector	projector,LLM	projector,LLM	Full model (except 2D encoder)
	Trainable Params (MB)	3.1	496.9	496.9	552.6
	Per-GPU batch size	1	1	1	1
	GPUs	4	4	4	4
	LR (ψ_{vision})	-	-	-	1×10^{-5}
LR ($\{\theta_{\text{proj/LLM}}\}$)	1×10^{-4}	1×10^{-5}	1×10^{-5}		1×10^{-5}
	Epochs	1	1	1	1

TABLE VIII: Multi-stage training hyperparameters of OpenDriveVLA-0.5B.

2) *Inference Efficiency*: Table IX reports the inference performance of OpenDriveVLA across three LLM scales (0.5B, 3B, 7B) under bf16 precision on a single NVIDIA A100 GPU. Evaluation is conducted on the NuScenes trajectory validation set with 6019 samples.

Model	LLM	GPUs	Speed (Sample/s)	Latency (s)	Max VRAM (GB)
0.5B	Qwen2.5-0.5B-Instruct	1	0.74	1.36	1.56
3B	Qwen2.5-3B-Instruct	1	0.54	1.85	7.35
7B	Qwen2.5-7B-Instruct	1	0.57	1.74	17.15

TABLE IX: Inference details of OpenDriveVLA under BF16 precision on a single NVIDIA A100 GPU.

VII. DATASETS DETAILS

A. Dataset Overview

We utilize a curated set of multimodal driving datasets derived from nuScenes [46] to support the multi-stage training of OpenDriveVLA, covering object-level captioning, visual question answering, scene description, and decision reasoning. An overview of the datasets used in Stage 1 and Stage 2 of OpenDriveVLA training is provided in Table X.

Dataset	#Train	#Val	Annotation Types	Type
TOD3Cap	1.89M	410K	Object Caption: appearance, motion, relationships	Dense Captioning
nuScenes-QA	376K	83K	existence, counting, object, status, comparison	Visual Question Answering
nuCaption	348K	72K	Scene Caption: layout, agents, hazards	Scene Description
nuX	28K	6K	Driving Decision Justification	Reasoning & Narration

TABLE X: VQA Datasets in OpenDriveVLA Stage 1/2 training, with sample counts, annotation, and types.

TOD3Cap [43]: TOD3Cap introduces the task of object-centric dense captioning in 3D driving scenes. It provides 2.3M human-verified natural language descriptions for over 64K objects across 850 nuScenes [46] scenes, covering appearance, motion, context, and inter-object relations. Each caption captures fine-grained semantics including what, where, how, and why, facilitating object-level alignment between 3D perception and language.

nuScenes-QA [45]: nuScenes-QA is a large-scale visual question answering benchmark tailored for autonomous driving. It contains 460K question–answer pairs over 34K multimodal driving scenes with synchronized images and LiDAR. Questions are generated using scene graphs and structured templates, spanning reasoning types such as existence, counting, attribute queries, spatial relations, and comparisons.

nuCaption [44]: nuCaption is a 3D scene captioning dataset constructed from nuScenes. It comprises both image-text and LiDAR-text pairs, with both global and viewpoint-specific captions describing traffic layout, object interactions, and potential risks. By aligning 3D spatial representations with language, nuCaption allows 3D captioning and scene-level reasoning.

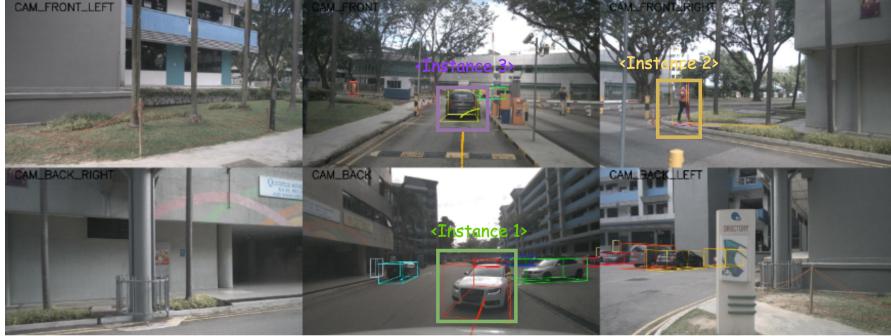
nuX [19]: nuX is a human-annotated explanation dataset designed for interpretable autonomous driving. For each keyframe, it provides natural language explanations combining factual narration (what is happening) and causal reasoning (why it is happening), grounded in the outputs of perception, prediction, and planning modules. It supports the development of driving models with aligned and transparent decision-making with textual interpretability.

B. Dataset Sample Visualization

	Stage 1	Stage 2	Stage 2.5	Stage 3
Anno	2D/3D Caption, Scene Description	Driving QA	Trajectory	Trajectory
Task	Hierarchical Feature Alignment	Driving Instruction Tuning	Agent-Env-Ego Interaction	Trajectory Planning
Source	TOD3Cap, nuScenes, nuCaption	nuCaption, nuS-QA, nuX	nuScenes	nuScenes
#Samples	536k	566k	459k	28k

TABLE XI: Overview of the Dataset for multi-stage training pipeline of OpenDriveVLA, detailing the specific tasks, annotation types, data sources, and number of training samples.

An overview of the multi-stage training data is provided in Table XI, covering task types, annotations, data sources, and sample counts. We provide visualization examples of gannotations we used during the hierarchical feature alignment training process of OpenDriveVLA, as shown in Figure 6 and Figure 7.



Q: Please provide a caption and the BEV coordinate for the following object:
 <track_start><Token of Instance 1><track_end>

A: A white car in the driving lane is moving slowly. It is in the back of ego car and within shouting distance. The BEV coordinate is (-1.39,-7.23).

Q: Please provide a caption and the BEV coordinate for the following object:
 <track_start><Token of Instance 2><track_end>

A: A person wearing a red shirt and black pants is moving slowly. It is in the front right of ego car and just a few steps away from ego car. The BEV coordinate is (7.60,8.88).

Q: Please provide a caption and the BEV coordinate for the following object:
 <track_start><Token of Instance 3><track_end>

A: A black, shiny, and sleek car in the driving lane is moving quickly. It is in the front of ego car and close by ego car. The BEV coordinate is (0.44,13.67).

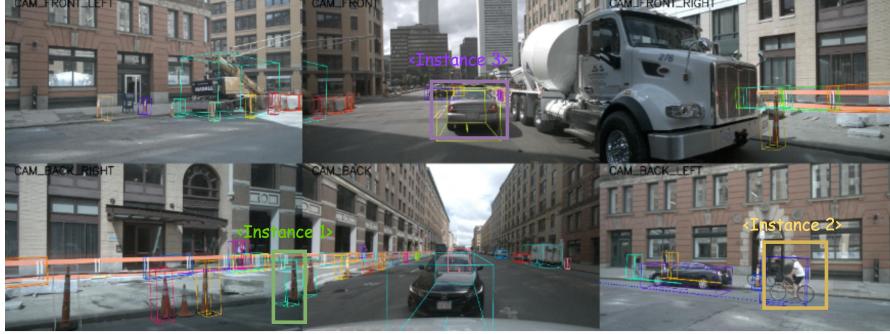
Q: Please provide a caption for the following map: <map_start><Token of Sample MAP><map_end>

A: This scene contains 10 lane-related elements:
 There are 3 instances of physical divider such as traffic cones, barriers, or guardrails.
 There are 6 instances of road boundary marking the edge of the drivable area.
 There is 1 instance of lane marker separating lanes moving in the same direction.

Q: Please provide a caption for the following scene: <scene_start><Token of Sample Scene><scene_end>

A: This is the scene caption of the **front left view**: The scene shows a tree with a rope tied around it, and a fence nearby. The scene is set in a park, with a sidewalk nearby. The weather appears to be sunny, and the time of the day is likely daytime.
 This is the scene caption of the **front view**: The scene shows a busy city street with cars and trucks driving down the road. There are buildings and parking meters in the background, indicating an urban environment. The street is likely bustling with activity, as it is a typical day in a city.
 This is the scene caption of the **front right view**: The scene shows a man walking down a sidewalk in a city, surrounded by a park-like setting....
 This is the scene caption of the **back right view**: The scene shows a parking lot with a tall building in the background....
 This is the scene caption of the **back view**: The road scene is set in a city, with a white car driving down a street....
 This is the scene caption of the **back left view**: The road scene is set in a city, with a street sign and a parking meter on the side of the road. The sign is blue and white, and the parking meter is green....

Fig. 6: Example 1 of Stage 1 dataset visualization from nuScenes Sample 90. Green boxes show instance token captions; Yellow boxes indicate map token captions; Red boxes represent scene token captions.



Q: Please provide a caption and the BEV coordinate for the following object:
<track_start><Token of Instance 1><track_end>

A: A trafficcone in the driving lane is not moving. It is in the back right of ego car and just a few steps away from ego car. The BEV coordinate is (6.73,-12.25).

Q: Please provide a caption and the BEV coordinate for the following object:
<track_start><Token of Instance 2><track_end>

A: A red bicycle in the driving lane is moving quickly. It is in the back left of ego car and just a few steps away from ego car. The BEV coordinate is (-9.74,-1.38).

Q: Please provide a caption and the BEV coordinate for the following object:
<track_start><Token of Instance 3><track_end>

A: A silver, shiny, and metallic car in the stop lane is not moving. It is in the front of ego car and close by ego car. The BEV coordinate is (0.55,9.71).

Q: Please provide a caption for the following map: <map_start><Token of Sample MAP><map_end>

A: This scene contains 21 lane-related elements:
There are 10 instances of physical divider such as traffic cones, barriers, or guardrails.
There are 6 instances of pedestrian crosswalk marked with white stripes.
There are 4 instances of road boundary marking the edge of the drivable area.
There is 1 instance of lane marker separating lanes moving in the same direction.

Q: Please provide a caption for the following scene: <scene_start><Token of Sample Scene><scene_end>

A: This is the scene caption of the front left view: The scene shows a city street with a construction site in the background. There are several cars parked along the street, and a truck is driving down the road. The street is lined with buildings, and the overall atmosphere suggests an urban environment.
This is the scene caption of the front view: The scene shows a busy city street with a mix of vehicles, including cars, a truck, and a bus....
This is the scene caption of the front right view: The scene shows a city street with a large truck parked on the side of the road. The street is lined with buildings, and there are several cars parked along the street....
This is the scene caption of the back right view: The scene shows a city street with a tall building in the background....
This is the scene caption of the back view: The scene shows a busy city street with tall buildings on both sides. The street is filled with cars, including a black car that is driving down the road....
This is the scene caption of the back left view: The scene shows a city street with a tall building in the background....

Fig. 7: Example 2 of Stage 1 dataset visualization from nuScenes Sample 3287. Green boxes show instance token captions; Yellow boxes indicate map token captions; Red boxes represent scene token captions.

VIII. RESULTS AND DISCUSSIONS

A. Results of Driving Question Answering

Figure 8 and Figure 9 show representative examples of OpenDriveVLA’s responses to diverse driving-related questions drawn from three datasets: nuScenes-QA, nuCaption, and nuX. These qualitative results highlight the model’s multi-level reasoning capabilities across perception, commonsense understanding, and contextual decision-making.

In Figure 8, the model provides a narration of the ego vehicle’s behavior while approaching and passing over a speed bump, accompanied by a causal explanation grounded in the surrounding scene. This demonstrates the model’s ability to generate coherent natural language outputs that reflect its interpretation of driving decisions. However, in the scene-level description, the model hallucinates that there are no pedestrians visible in the right-front view, despite the presence of one pedestrian in the image.

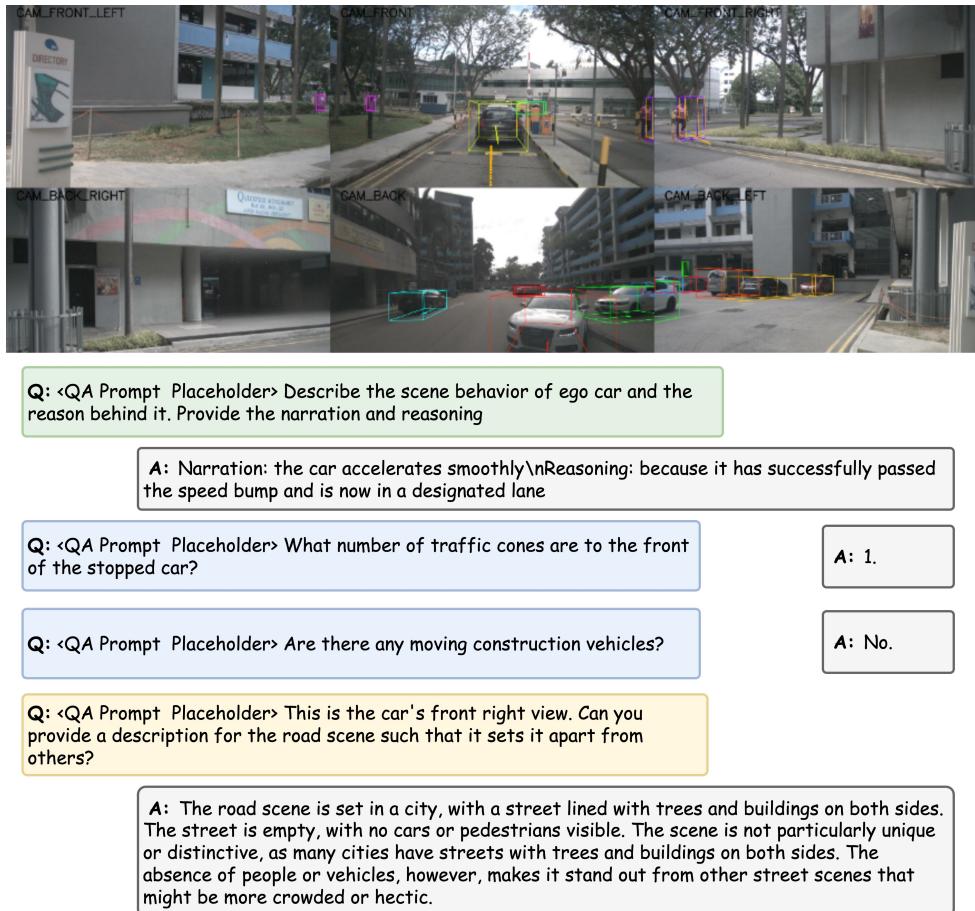


Fig. 8: Example 1 of Stage 2 dataset visualization from nuScenes Sample 6801. Green boxes show nuX dataset predictions; blue boxes are from nuScenes-QA dataset, while yellow boxes are from nuCaption.

Figure 9 depicts a more complex urban scenario involving multiple interacting agents near an intersection. The model identifies the ego vehicle's maneuver and provides a causal explanation grounded in its spatial context, correctly localizing the vehicle as it turns at the junction. In the scene-level description, the model can ground key static and dynamic elements, including the presence of a traffic light, pedestrians near a crosswalk, and parked vehicles along the roadside. However, the caption exhibits positional inaccuracy when referencing the camera view, particularly in the identification of the front view. This may be attributed to the model being trained with scene-level captions in a camera-agnostic fashion, without explicit supervision for distinguishing between different camera perspectives.

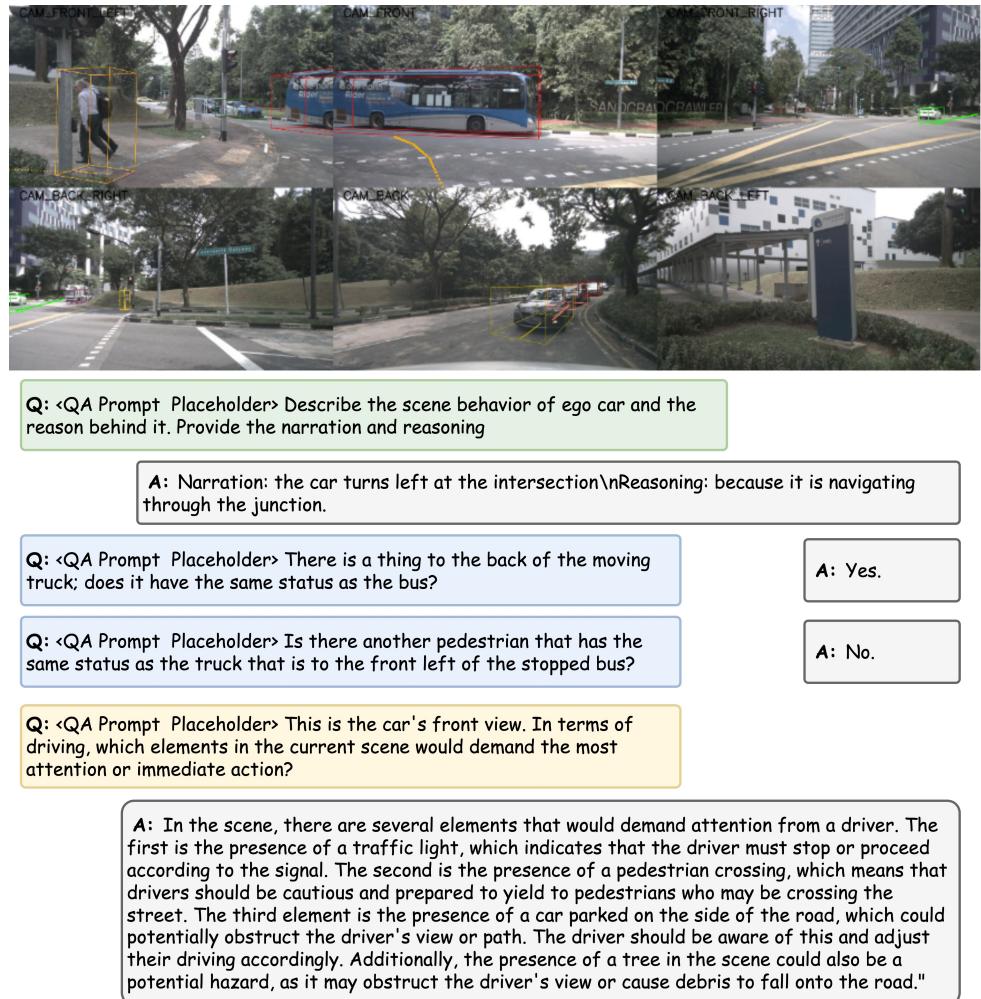


Fig. 9: Example 2 of Stage 2 dataset visualization from nuScenes Sample 6801. Green boxes show nuX dataset predictions; blue boxes are from nuScenes-QA dataset, while yellow boxes are from nuCaption.

B. Results of Agent Motion Prediction

Figure 10 and Figure 11 present qualitative results of OpenDriveVLA on the agent motion prediction task of Stage 2.5, where the model jointly reasons over agent trajectories, environment, and ego vehicle state.



Fig. 10: Example 1 of agent motion prediction result of stage 2.5 Agent-Environment-Ego interaction modeling on nuScenes validation set Sample 8587.

Figure 10 depicts a night-time urban intersection with multiple static and dynamic agents under low-light conditions. The figure highlights two turning vehicles. The predicted motion for the white sedan on the right side of the scene is consistent with the lane orientation and road semantics. In contrast, the white vehicle ahead of the ego car, the model predicts a trajectory curving toward the right. Yet, given the surrounding road geometry and lane configuration, a left turn would be the more plausible maneuver.

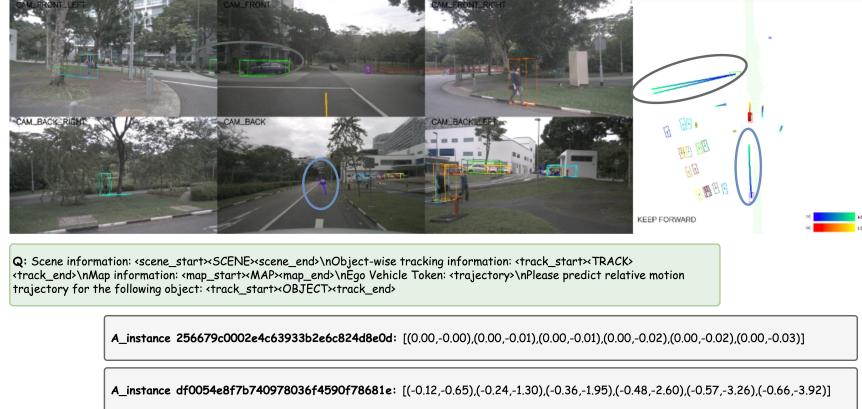


Fig. 11: Example 2 of agent motion prediction result after stage 2.5 Agent-Environment-Ego interaction modeling on nuScenes validation set Sample 32496.

In Figure 11, the model predicts motion trajectories for multiple agents in a curved road scenario during daytime. Notably, the visualization highlights two predicted trajectories: one for a gray sedan in

the front view, and another for a vehicle observed in the back view. Both predictions reflect distinct motion patterns consistent with the road layout and surrounding context. This example demonstrates the model’s capacity to capture plausible motion uncertainty and agent-specific intention under partially observable environments.

C. Planning Results

1) *Comparison with Prior Methods.*: Figure 12 presents a qualitative comparison between the open-loop planning results of OpenDriveVLA and UniAD [10] in a challenging narrow-road scenario with multiple parked vehicles.

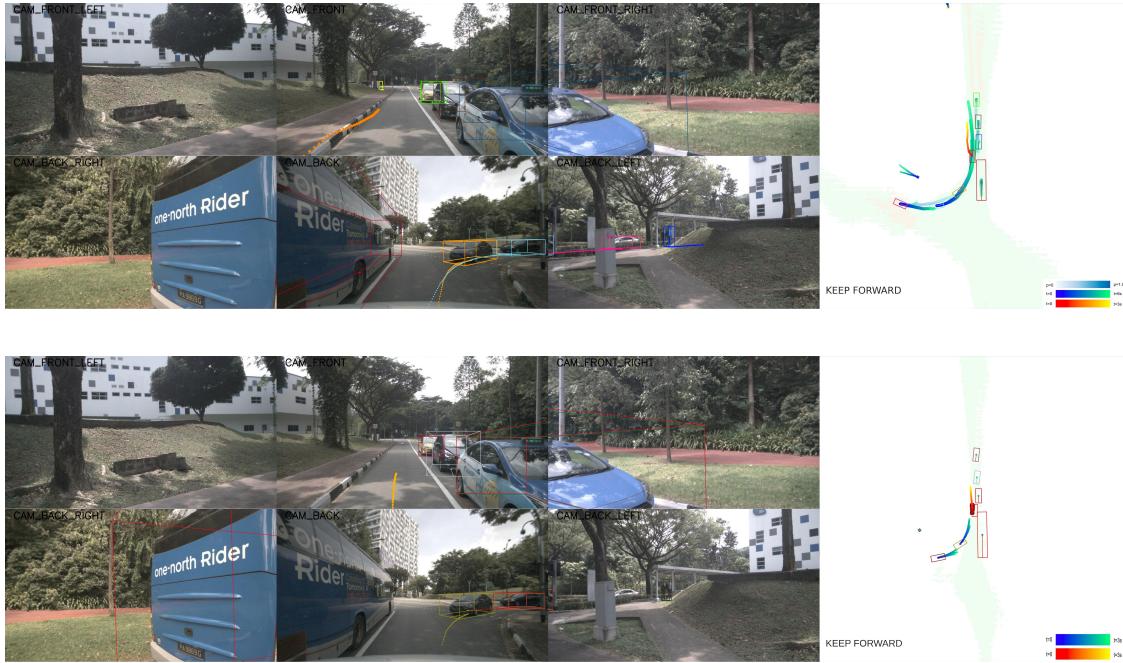


Fig. 12: Qualitative comparison of open-loop planning. Top: Planning results of UniAD [10]. Bottom: Planning results of OpenDriveVLA-7B (Ours). The agent motion prediction results are visualized after the agent-environment-ego interaction stage.

In this scenario, UniAD exhibits overly sensitive reactions to the parked vehicles on the right side of the road, resulting in unstable and zigzagging planned trajectories. In contrast, OpenDriveVLA generates smoother and more consistent motion plans that better follow the intended driving path while maintaining safe clearance from surrounding objects. This demonstrates a stronger ability to reason about scene semantics and generate robust and spatially grounded trajectories.

2) *Qualitative Results of Driving Instruction Following.*: Figure 13 and Figure 14 illustrate the trajectory planning behavior of OpenDriveVLA under different conditional driving instructions, including left turn, right turn, and keep forward.

Figure 13 illustrates a complex urban intersection where the ground-truth instruction in nuScenes dataset is going forward. The predicted paths remain safe and well-aligned with the scene context under new driving instructions, demonstrating the model’s capacity to flexibly interpret high-level commands while maintaining collision-free behavior. Similarly, Figure 14 shows another intersection scenario in a suburban

environment. Despite differences in layout and visual context, OpenDriveVLA consistently adapts its planned motion to match the given instruction safely. These results demonstrate the instruction following ability of OpenDriveVLA across various environments and its robustness in instruction-conditioned planning.

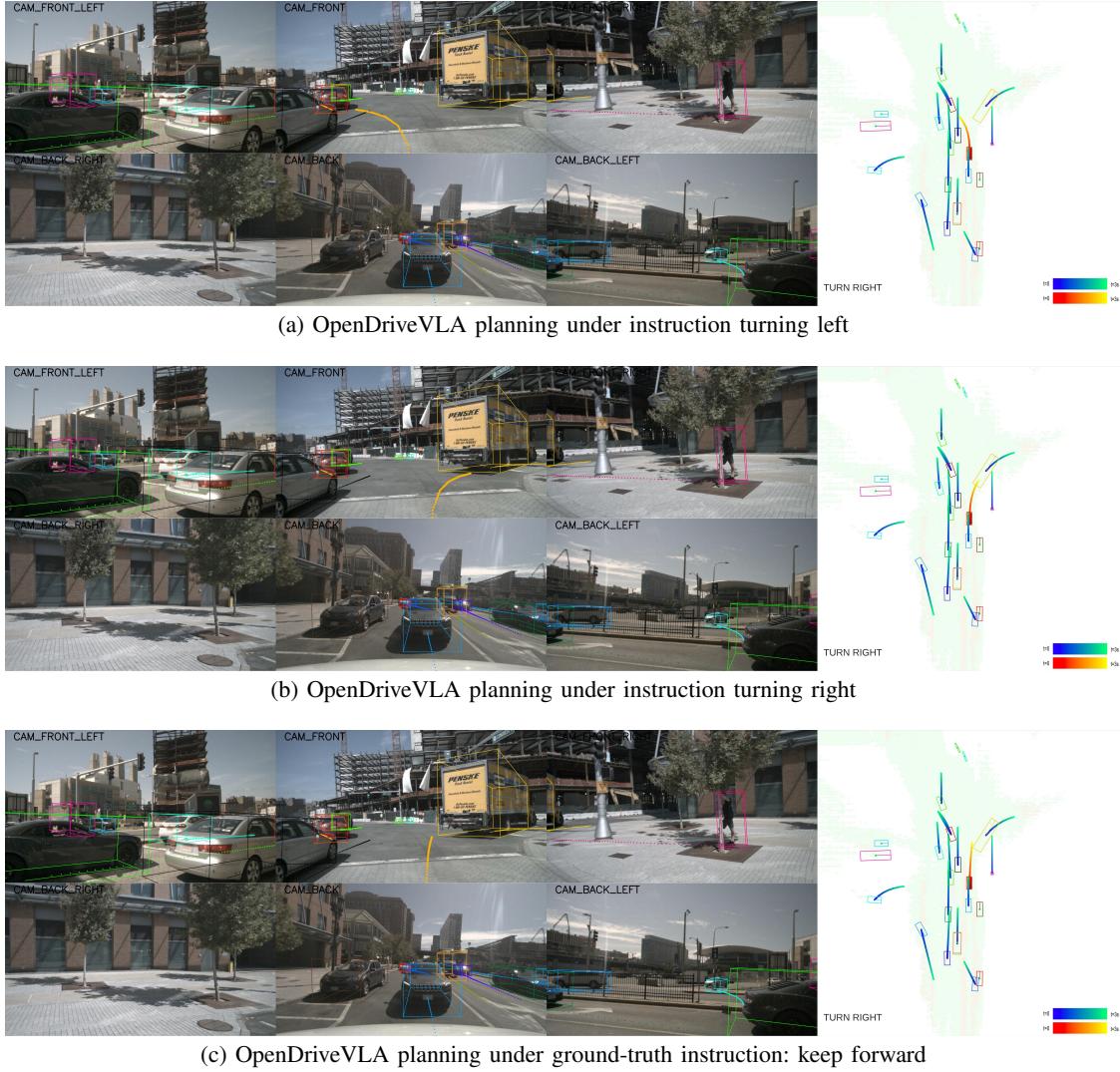


Fig. 13: Example 1 of trajectory planning results of OpenDriveVLA after stage 3 with different driving instructions.

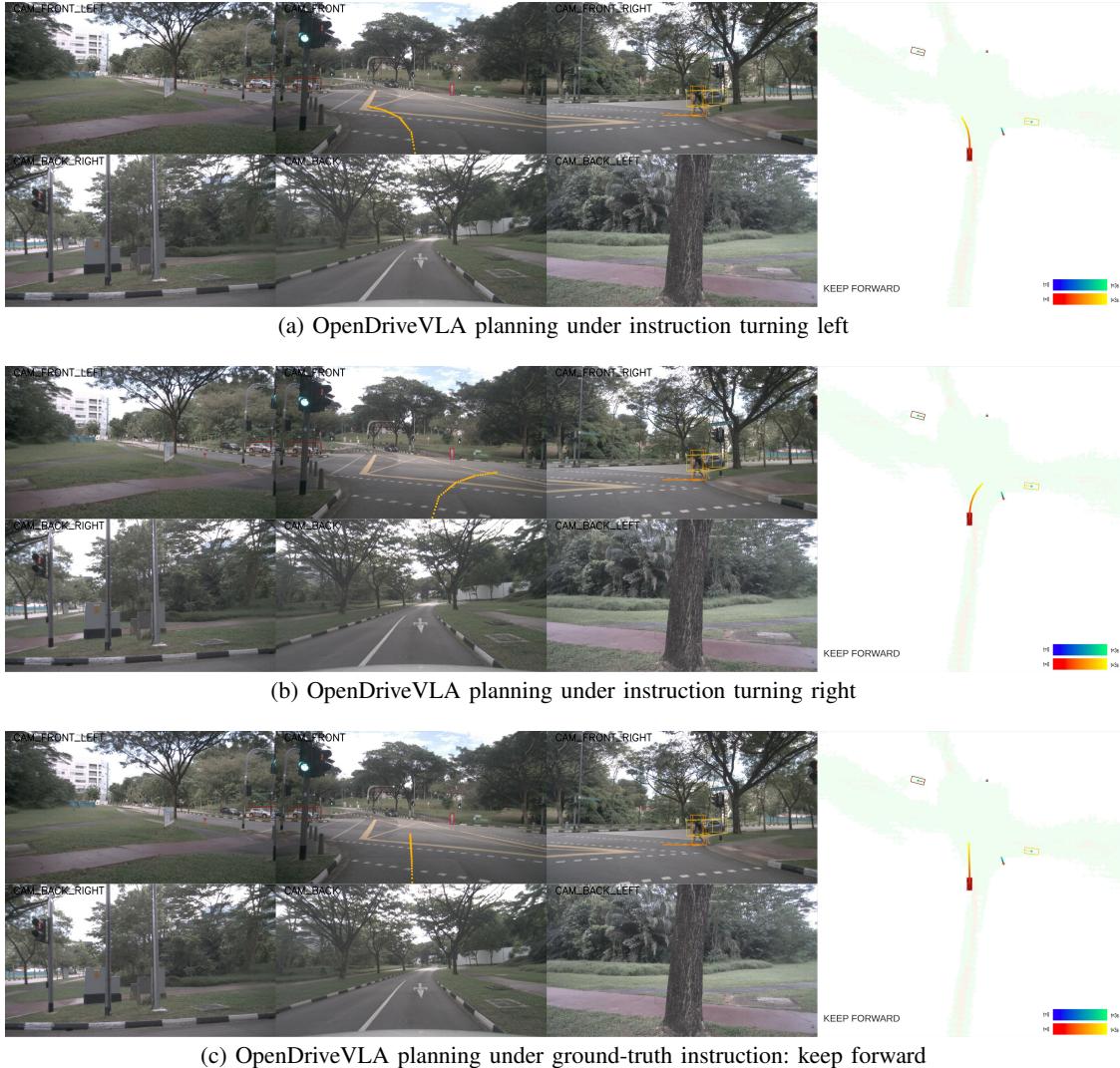


Fig. 14: Example 2 of trajectory planning results of OpenDriveVLA after stage 3 with different driving instructions.

D. Discussions

1) *Results Regarding Model Size.*: The results in Main Content (Table 1,2,3) show that, while increasing model size generally leads to improved performance, the larger OpenDriveVLA-7B model does not consistently outperform the smaller 0.5B and 3B variants across all benchmarks. Specifically, on the Nu-X dataset, the 0.5B model achieves obviously higher CIDEr and ROUGE-L scores than the 7B version. In open-loop trajectory planning, the differences in average L2 error and collision rate among the three models are relatively small. Although the 7B model remains competitive overall, its performance gains are not always significant and, in some cases, fall behind those of the smaller models.

There are several possible reasons based on our experimental observations. First, the performance gains associated with scaling large language models typically depend on access to extensive and diverse training data. As detailed in Table X, the instruction tuning data and domain-specific driving annotations used in our current setup may not be sufficient to fully leverage the representation capacity of the 7B model, especially in the autonomous driving domain, where large-scale open-source vision language datasets are still very few. Hence, the expected improvements from model scaling are not consistently realized across all tasks.

Second, larger autoregressive models tend to rely more heavily on language priors during generation. In structured multimodal reasoning tasks, this reliance may weaken the model’s ability to maintain accurate visual grounding, particularly in scenarios requiring fine-grained spatial understanding or factual precision. Smaller models, though with limited capacity, can retain stronger coupling between vision and language inputs, which can lead to more stable behavior under constrained supervision. This also suggests that additional high-quality visual-textual data is necessary to facilitate further development.

Third, the optimization dynamics of larger models can be more sensitive to hyperparameters and experimental settings in domain-specific tasks. Without sufficient diversity in the training dataset, the larger model may overfit to dominant patterns in the training set, resulting in reduced robustness and generalization during inference.

In summary, model size alone is not a robust indicator of performance in this domain-specific autonomous driving task with limited data sources. Instead, data quantity and quality, and multi-stage training strategies play a critical role in achieving robust and scalable performance.

2) *Current Limitations*: While OpenDriveVLA achieves promising results, several limitations remain. First, the model relies on implicit reasoning patterns acquired through instruction tuning, without explicit step-by-step chain-of-thought deduction during inference. Although this design helps maintain inference efficiency, it may deteriorate the reasoning performance and limit the model’s ability to handle complex scenarios. Second, the autoregressive nature of LLM introduces inference latency. Even with careful control of token lengths, the sequential decoding process remains a bottleneck for deployment in high-speed driving scenarios. Further improvements in model quantization and improved decoding strategies will be beneficial to meet real-time requirements. Finally, our current planning evaluation is limited to an open-loop setting. This does not capture the interactive feedback dynamics of realistic driving and may lead to overestimated robustness, underscoring the importance of extending OpenDriveVLA to closed-loop, language-aware simulation environments.

3) *Extension to Closed-Loop Planning*: As mentioned, the current planning evaluation of OpenDriveVLA is conducted in an open-loop setting on the nuScenes dataset. While this setup allows for controlled and reproducible benchmarking, it does not account for the feedback dynamics of ego-agent interactions in real-world traffic. Prior studies also show that nuScenes open-loop benchmark may lead to overly optimistic conclusions [34]. Several established closed-loop benchmarks, such as nuPlan [59], Bench2Drive [60], and NaviSim [61], have been introduced for more reliable planning assessment.

Unfortunately, these benchmarks currently lack vision-language annotations that are central to the training of domain-specific LLM-based Autonomous Driving models. While nuScenes-derived datasets such as TOD3Cap, nuScenes QA, nuCaption, and nuX provide rich multimodal supervision in the form of instance-level and scene-level captions and question–answer pairs. These additional datasets are the result of extensive annotation efforts from multiple prior works and serve as a foundation for both training and evaluating LLM-based autonomous driving models. To extend OpenDriveVLA toward closed-loop evaluation, future work involves building an automated data generation pipeline that (semi-)automatically generates instance-level, map-level, and scene-level descriptions, as well as question–answer pairs linked to driving behaviors.

Despite the limitations, OpenDriveVLA still demonstrates strong performance across both open-loop planning and multiple driving VQA benchmarks, including nuScenes-QA, nuCaption, and nuX. The model consistently exhibits robust scene understanding, accurate instruction following, and semantically grounded action generation. These results validate the effectiveness of the proposed model and training techniques, and provide a solid experimental foundation for future extensions into closed-loop evaluation frameworks.