

DRIVEVLM: The Convergence of Autonomous Driving and Large Vision-Language Models

Xiaoyu Tian^{1*} Junru Gu^{1*} Bailin Li^{2*} Yicheng Liu¹ Chenxu Hu¹
 Yang Wang² Kun Zhan² Peng Jia² Xianpeng Lang² Hang Zhao^{1†}

¹IIS, Tsinghua University ²Li Auto

Abstract

A primary hurdle of autonomous driving in urban environments is understanding complex and long-tail scenarios, such as challenging road conditions and delicate human behaviors. We introduce DriveVLM, an autonomous driving system leveraging Vision-Language Models (VLMs) for enhanced scene understanding and planning capabilities. DriveVLM integrates a unique combination of chain-of-thought (CoT) modules for scene description, scene analysis, and hierarchical planning. Furthermore, recognizing the limitations of VLMs in spatial reasoning and heavy computational requirements, we propose DriveVLM-Dual, a hybrid system that synergizes the strengths of DriveVLM with the traditional autonomous driving pipeline. DriveVLM-Dual achieves robust spatial understanding and real-time inference speed. Extensive experiments on both the nuScenes dataset and our SUP-AD dataset demonstrate the effectiveness of DriveVLM and the enhanced performance of DriveVLM-Dual, surpassing existing methods in complex and unpredictable driving conditions. ¹

1. Introduction

Autonomous driving, with its great promise to revolutionize transportation and urban mobility, has been one of the most active areas of research and development over the past two decades. A primary hurdle to a fully autonomous driving system is scene understanding [3], which involves navigating complex, unpredictable scenarios such as adverse weather, intricate road layouts, and unforeseen human behaviors.

Existing autonomous driving systems, typically comprising 3D perception, motion prediction, and planning, struggle with these scene understanding challenges. Specifically, 3D

perception [28, 30, 39, 45] is limited to detecting and tracking familiar objects, omitting rare objects and their unique attributes; motion prediction [16, 18, 33, 35, 56] and planning [2, 31, 38] focus on trajectory-level actions, often neglecting the decision-level interactions between objects and the vehicle.

We introduce **DriveVLM**, a novel autonomous driving system that aims at the scene understanding challenges, capitalizing on the recent Vision-Language Models (VLMs) [32, 51, 54, 57] which have demonstrated exceptional prowess in visual comprehension and reasoning. Specifically, DriveVLM contains a Chain-of-Though (CoT) process with three key modules: *scene description*, *scene analysis*, and *hierarchical planning*. The scene description module linguistically depicts the driving environment and identifies critical objects in the scene; the scene analysis module delves into the characteristics of the critical objects and their influence on the ego vehicle; the hierarchical planning module formulates plans step-by-step, from meta-actions and decision descriptions to waypoints. These modules respectively correspond to the components of the traditional *perception-prediction-planning* pipeline, but they differ in that they tackle *object perception*, *intention-level prediction* and *task-level planning*, which were extremely challenging to cope with in the past.

While VLMs excel in visual understanding, they have limitations in spatial grounding and reasoning, and their computational intensity poses challenges for onboard inference speed. Therefore we further propose **DriveVLM-Dual**, a hybrid system that combines the strengths of both DriveVLM and traditional systems. DriveVLM-Dual optionally integrates DriveVLM with traditional 3D perception and planning modules, such as 3D object detectors, occupancy networks, and motion planners, enabling the system to achieve 3D grounding and high-frequency planning abilities. This dual system design, akin to the human brain’s slow and fast thinking processes, adapts efficiently to varying complexity in driving scenarios.

*Equal contribution. Listing order is random.

†Corresponding to: hangzhao@mail.tsinghua.edu.cn

¹Project page: <https://tsinghua-mars-lab.github.io/DriveVLM>

Meanwhile, we formally define the scene understanding and planning (SUP) task, and propose new evaluation metrics to assess the scene analysis and meta-action planning capabilities of DriveVLM and DriveVLM-Dual. Furthermore, we carry out a comprehensive data mining and annotation pipeline to construct an in-house SUP-AD dataset for the SUP task.

Extensive experiments on the nuScenes dataset and our dataset demonstrate the superiority of DriveVLM, especially in few-shot situations. Moreover, DriveVLM-Dual surpasses state-of-the-art end-to-end motion planning methods.

In summary, the contributions of this paper are fourfold:

1. We introduce DriveVLM, a novel autonomous driving system that leverages VLMs for effective scene understanding and planning.
2. We further introduce DriveVLM-Dual, a hybrid system that incorporates DriveVLM and a traditional autonomous pipeline. DriveVLM-Dual achieves improved spatial reasoning and real-time planning capabilities.
3. We present a comprehensive data mining and annotation pipeline to construct a scene understanding and planning dataset, together with metrics to evaluate the SUP task.
4. Extensive experiments on the nuScenes dataset and our SUP-AD dataset demonstrate the superior performance of DriveVLM and DriveVLM-Dual in complex driving scenarios.

2. Related Work

Vision-Language Models (VLMs). Recently, there has been a surge in research on large Vision-Language Models (VLMs), exemplified by works such as MiniGPT-4 [57], LLaVA [32], Qwen-VL [1], and others [12, 44, 54, 55]. These models integrate pre-trained vision encoders with large language models, enabling large language models to address many tasks involving images as input. In general, these methods align image features with the input embedding space of the language model through Q-former [29] or linear mapping [32]. A crucial step in the training process is supervised fine-tuning using instructional data containing images and text, enhancing the overall performance of vision language models. VLMs can be used in various scenarios, especially robotics [4, 5, 15, 23, 37]. Specifically, given instructions, input images, and robot states, vision language models output corresponding actions that can be high-level instructions [15] or low-level robot actions [5]. DriveVLM focuses on utilizing VLMs to assist in autonomous driving, thereby establishing a novel framework. Concurrent to our work, [51] also shares a similar motivation.

Learning-based Planning. The integration of learning frameworks into motion planning has been an active area of research since Pomerleau [38] pioneering contributions. One promising line of work is Reinforcement learning and

imitation learning [8, 9, 43]. These methods can learn an end-to-end planning policy that directly maps raw sensory inputs to control actions [43]. They are particularly suited to high-dimensional state and action spaces, a common challenge in motion planning. However, the direct generation of control outputs from sensor data poses significant challenges in robustness and safety assurance [11]. Several works [7, 20, 46, 53] improve interpretability by explicitly building dense cost maps derived from learning-based modules. While dense cost maps effectively integrate predictions about traffic agents’ future movements and environmental factors, their performance heavily depends on costs tailored through human experience and the trajectory sampling distribution [25]. A recent trend involves training multiple blocks in an end-to-end fashion [7, 20, 22, 41]. These methods enhance overall performance, but rely on backpropagation from future trajectory predictions loss in a less interpretable decision-making process [10]. Our model, DriveVLM, addresses the complexities of long-tailed driving scenarios, often challenging for other methods, by leveraging the generalization and reasoning capabilities of vision-language models. Moreover, users can easily interact with our model through the intuitive language interface provided by the vision-language model, enhancing interpretability.

Driving Caption Datasets. Recent works [19, 51, 52] argue that language captions are an important medium to connect human knowledge with the driving objective, helping to inform decisions and actions. In support of this trend, some efforts have enhanced mainstream driving scene datasets. Refer-KITTI [48] annotates objects in the KITTI dataset [17] with language prompts that can reference a collection of objects. Talk2Car [13], NuPrompt [49] and nuScenes-QA [40] introduce free-form captions and QA annotation to the nuScenes dataset [6]. However, these works enrich the datasets that are perception-focused and often contain simple traffic scenes. Instead of augmenting existing datasets, BDD-X [27] and BDD-OIA [50] offer datasets with natural language explanations for the ego vehicle’s actions. HAD [26] employs natural language commands to produce salient maps from drivers’ gaze data. Rank2Tell [42] and DRAMA [34] annotate language explanations and risk localization for traffic scenarios. While these datasets provide scenes tailored for utilizing natural language, there is a lack of enough data that capture scenarios that are crucial for identifying issues that could lead to safety concerns in self-driving systems. Our SUP-AD dataset stands in contrast by purposefully gathering a diverse array of challenging, long-tail scenarios that are essential for addressing complex scene understanding and planning.

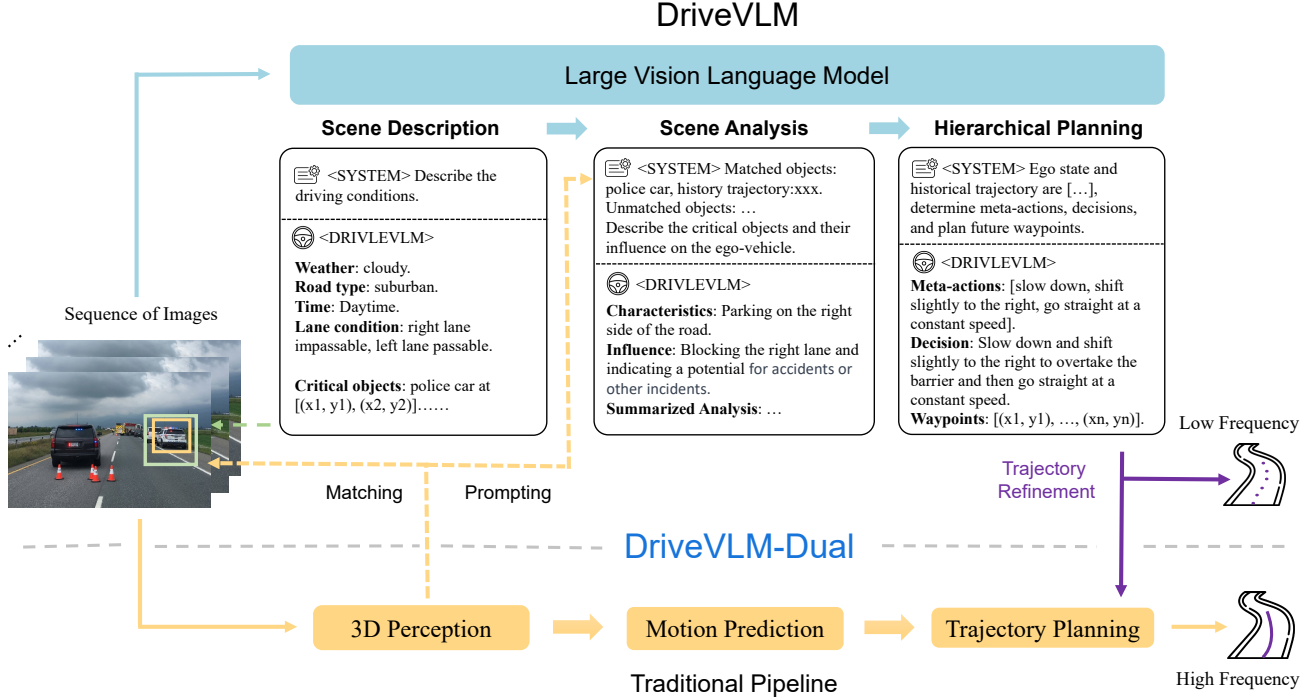


Figure 1. **DriveVLM and DriveVLM-Dual model pipelines.** DriveVLM accepts sequences of images as input and, through a Chain-of-Thought (CoT) mechanism, outputs scene description, scene analysis, and hierarchical planning results. DriveVLM-Dual further incorporates traditional 3D perception and trajectory planning modules to achieve spatial reasoning capability and real-time trajectory planning.

3. DriveVLM

3.1. Overview

The overall pipeline of DriveVLM is illustrated in Figure 1. A sequence of images is processed by a large Vision Language Model (VLM) to perform a special chain-of-thought (CoT) [47] reasoning to derive the driving planning results. The large VLM involves a vision transformer encoder [14] and a Large Language Model (LLM). The vision encoder produces image tokens; then an attention-based extractor aligns these tokens with the LLM; finally, the LLM performs CoT reasoning. The CoT process can be divided into three modules: scene description (Section 3.2), scene analysis (Section 3.3), and hierarchical planning (Section 3.4).

DriveVLM-Dual is a hybrid system that combines DriveVLM and the traditional autonomous driving pipeline, taking the best of both worlds. It incorporates 3D perception results as language prompts for enhanced 3D scene understanding capability, and further refines the trajectory waypoints with a real-time motion planner. We detail its design and advantages in Section 3.5.

3.2. Scene Description

The scene description module is composed of environment description and critical object identification.

Environment description. Driving environments, such as weather and road conditions, have a non-negligible impact on driving difficulty. Therefore, the model is first prompted to output a linguistic description E of the driving environment, including several conditions: $E = \{E_{\text{weather}}, E_{\text{time}}, E_{\text{road}}, E_{\text{lane}}\}$, each representing a crucial aspect of the driving environment.

- E_{weather} details the weather conditions, ranging from sunny to snowy. Conditions such as rain or fog demand more cautious driving approaches due to reduced visibility and road grip.
- E_{time} encapsulates the time of day, differentiating between daytime and nighttime driving scenarios. For example, nighttime driving, characterized by reduced visibility, necessitates cautious driving strategies.
- E_{road} classifies the type of roads, including urban, suburban, rural, or highway, where each road type presents unique challenges.
- E_{lane} gives a description of the lane conditions, identifying the vehicle’s current lane and potential alternatives for maneuvering. This information is vital for lane selection and safe lane changes.

Critical object identification. In addition to environmental conditions, various objects in driving scenarios significantly influence driving behaviors. Unlike traditional autonomous

driving perception modules, which detect all objects within a specific range, we solely focus on identifying *critical objects* that are most likely to influence the current scenario, inspired by human cognitive processes during driving. Each *critical object*, denoted as O_c , contains two attributes: the object category c and its approximate bounding box coordinates $b(x_1, y_1, x_2, y_2)$ on the image. The category and coordinates are mapped to their corresponding language *token_id* in the language modality, enabling seamless integration into the following modules. Moreover, taking advantage of the pre-trained vision encoder, DriveVLM can identify long-tail *critical objects* that may elude typical 3D object detectors, such as road debris or unusual animals.

3.3. Scene Analysis

In the traditional autonomous driving pipeline, the prediction module typically concentrates on forecasting the future trajectories of objects. The emergence of advanced vision-language models has provided us with the ability to perform a more comprehensive analysis of the current scene.

Critical Object Analysis. After identifying the critical objects, we analyze their characteristics and potential influence on the ego vehicle. Characteristics contain three aspects of a critical object: static attributes C_s , motion states C_m , and particular behaviors C_b . Static attributes C_s describe inherent properties of objects, such as a roadside billboard’s visual cues or a truck’s oversized cargo, which are critical in preempting and navigating potential hazards. Motion states C_m describe an object’s dynamics over a period, including position, direction, and action—characteristics that are vital in predicting the object’s future trajectory and potential interactions with the ego vehicle. Particular behaviors C_b refer to special actions or gestures of an object that could directly influence the ego vehicle’s next driving decisions. For instance, a traffic officer’s hand signals are critical in this context, as they can override standard traffic rules and necessitate a corresponding response from the autonomous system. We do not require the model to analyze the three characteristics (C_s, C_m, C_b) for all the objects. In practice, only one or two apply to a critical object.

Upon analyzing these characteristics, DriveVLM then predicts the potential influence I of each critical object on the ego vehicle. For example, a drunken pedestrian on the roadside could potentially step onto the road and block our way. Compared to trajectory-level prediction in the traditional pipeline, the analysis of the potential influence of critical objects is crucial for the system’s adaptability to real-world and long-tail driving scenarios.

Scene-level Summary S . The scene-level analysis summarizes all the critical objects together with the environmental description. This summary gives a comprehensive understanding of the scene, linking the following planning module.

3.4. Hierarchical Planning

We integrate the scene description and scene analysis to form a summary of the driving scenario. The summary is further combined with the route, ego pose and velocity to form a prompt for planning. Finally, DriveVLM progressively generates driving plans, in three stages: meta-actions, decision description, and trajectory waypoints.

Meta-actions A . A meta-action, denoted as a_i , represents a short-term decision of the driving strategy. These actions fall into 17 categories, including but not limited to acceleration, deceleration, turning left, changing lanes, minor positional adjustments, and waiting. To plan the ego vehicle’s future maneuver over a certain period, we generate a sequence of meta-actions. Each meta-action in this sequence is pivotal, contributing cumulatively to the strategic navigation of the vehicle in the scene.

Decision description D . Decision description D articulates the more fine-grained driving strategy the ego vehicle should adopt. It contains three elements: Action \mathcal{A} , Subject \mathcal{S} , and Duration \mathcal{D} . *Action* pertains to meta actions such as ‘turn’, ‘wait’, or ‘accelerate’. *Subject* refers to the interacting object, such as a pedestrian, a traffic signal, or a specific lane. *Duration* indicates the temporal aspect of the action, specifying how long it should be carried out or when it should start. An example of a decision description is: “Wait (\mathcal{A}) for the pedestrian (\mathcal{S}) to cross, then (\mathcal{D}) proceed to accelerate (\mathcal{A}) and merge into the right lane (\mathcal{S}).”. This structured decision description allows for clear, concise, and actionable instructions for the autonomous system.

Trajectory waypoints W . Upon establishing the decision description D , our next phase involves the generation of corresponding trajectory waypoints. These waypoints, denoted by $W = \{w_1, w_2, \dots, w_n\}$, $w_i = (x_i, y_i)$, depict the vehicle’s path over a certain future period with predetermined intervals Δt . We map these numerical waypoints into language tokens for auto-regressive generation. In this way, DriveVLM achieves seamless integration of its linguistic processing module with spatial navigation. The trajectory waypoints are the spatial manifestation of the meta-actions and decision descriptions, which can be directly fed into subsequent control modules.

3.5. DriveVLM-Dual

Although VLMs are adept at recognizing long-tail objects and understanding complex scenarios, they often struggle with precisely comprehending spatial positions and detailed motion states of objects. This shortfall, noted in previous research and our pilot studies, poses a significant challenge. What is worse, the humongous model size of VLMs leads to high latency, impeding their ability to respond in real-time for autonomous driving. To address these challenges, we propose DriveVLM-Dual, a collaboration between DriveVLM and the traditional autonomous driving system. This novel

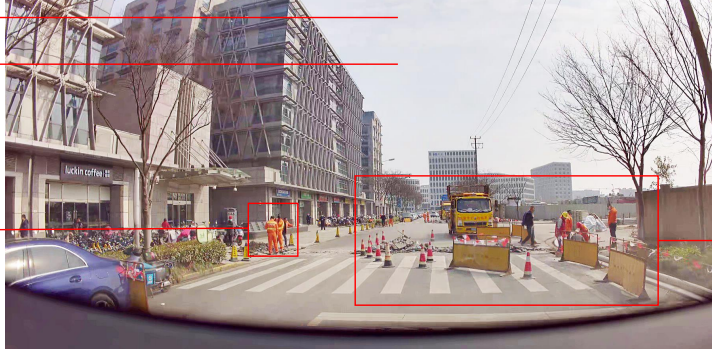
Scene Summary: The ego vehicle is moving at a constant speed along the current lane, with ongoing road construction work ahead; there are three construction workers working on the left side of the lane at the roadside.

Weather: Sunny

Time: Daytime

Critical Object 1:

Class:	Three Construction Workers
Characteristic:	Construction work on the side of the lane to the left of the host vehicle
Influence:	Affects the normal speed of the host vehicle



Road Condition: Construction

Lane Condition: Own Lane

Critical Object 2:

Class:	Construction Zone
Characteristic:	Road repair in front of the host vehicle lane
Influence:	Affects the host vehicle to drive straight normally

Meta Action: ["Slow down", "Change lane to the left", "Go straight slowly"]

Decision Description: Decelerate and change lanes to the left, keeping a safe distance from the construction workers on the left front side.

Figure 2. An annotated sample of the SUP-AD dataset.

approach involves two key strategies: incorporating 3D perception for critical object analysis, and high-frequency trajectory refinement.

Integrating 3D Perception. We represent objects detected by a 3D detector as $O_{3D} = \{c_{3D}^i, b_{3D}^i\}$, where b_{3D}^i denotes the i -th bounding box and c_{3D}^i denotes its category. These 3D bounding boxes are then back-projected onto 2D images to derive corresponding 2D bounding boxes b_{2D}^i . We conduct IoU matching between these 2D bounding boxes b_{2D}^i and b_c^j . b_c^j are the bounding boxes of previously identified critical objects $O_{critical} = \{c_c^j, b_c^j\}$. We classify critical objects that meet a certain approximate IoU threshold and belong to the same category as matched critical objects $O_c^{matched}$, defined as

$$O_c^{matched} = \{c_c^j, b_c^j\}, \quad \text{if exists } c_c^j = c_{2D}^i \quad (1)$$

$$\text{and } aIoU(b_c^j, b_{2D}^i) > \tau,$$

$$aIoU(b_c^j, b_{2D}^i) = \frac{S_{b_c^j \cap b_{2D}^i}}{S_{b_{2D}^i}} \quad (2)$$

Those critical objects without a corresponding match in the 3D data are noted as $O_c^{unmatched}$.

In the scene analysis module, for $O_c^{matched}$, the center coordinates, orientations, and historical trajectories of the corresponding 3D objects are used as language prompts for the model, assisting in object analysis. Conversely, for $O_c^{unmatched}$, analysis relies solely on the language tokens derived from the image. This novel use of 3D perception results as prompts enables DriveVLM-Dual to understand the locations and motions of critical objects more accurately, enhancing the overall performance.

High-Frequency Trajectory Refinement. Compared to traditional planners, DriveVLM, due to its immense parameter size inherent to Vision-Language Models (VLMs), exhibits

significantly slower speeds while generating a trajectory. To achieve real-time, high-frequency inference capabilities, we integrate it with a conventional planner to form a slow-fast dual system, combining the advanced capabilities of DriveVLM with the efficiency of traditional planning methods. After obtaining a trajectory from DriveVLM at low frequency, denoted as W_{slow} , we take it as a reference trajectory for a classical planner for high-frequency trajectory refinement. In the case of an optimization-based planner, W_{slow} serves as the initial solution for the optimization solver. For a neural network-based planner, W_{slow} is used as an input query, combined with additional input features f , and then decoded into a new planning trajectory denoted as W_{fast} . The formulation of this process can be described as:

$$W_{fast} = \text{Planner}([W_{slow}, f]). \quad (3)$$

This refinement step ensures that the trajectory produced by DriveVLM-Dual (1) achieves higher trajectory quality, and (2) meets real-time requirements. In practice, the two branches operate asynchronously in a slow-fast manner, where the planner module in the traditional autonomous driving branch can selectively receive trajectory from the VLM branch as additional input.

4. Task and Dataset

To fully exploit the potential of DriveVLM and DriveVLM-Dual in handling complex and long-tail driving scenarios, we formally define a task called *Scene Understanding for Planning* (Section 4.1), together with a set of evaluation metrics (Section 4.2). Furthermore, we propose a data mining and annotation protocol to curate a scene understanding and planning dataset (Section 4.3).

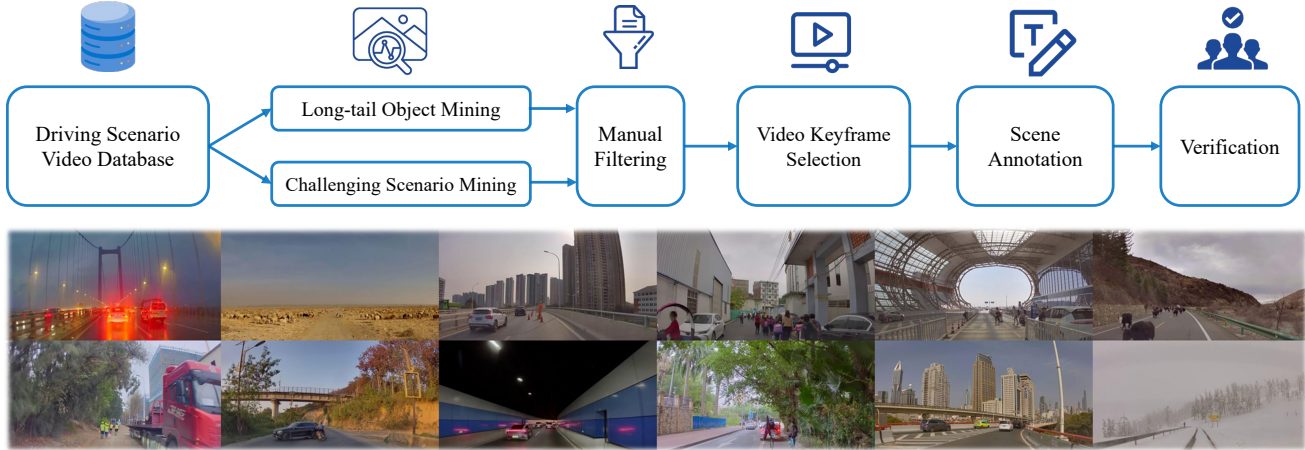


Figure 3. The proposed data mining and annotation pipeline for constructing a scene understanding and planning dataset (above); Scenario examples randomly sampled from the dataset (below) demonstrate the diversity and complexity of the dataset.

4.1. Task Definition

The Scene Understanding for Planning task is defined as follows. The input comprises multi-view videos \mathcal{V} from surrounding cameras and optionally 3D perception results \mathcal{P} from a perception module. The output includes the following components:

1. **Scene Description E** : Composed of weather condition E_{weather} , time E_{time} , road condition E_{road} , and lane conditions E_{lane} .
2. **Scene Analysis S** : Including object-level analysis and scene-level summary S .
3. **Meta Actions A** : A sequence of actions representing task-level maneuvers.
4. **Decision Description D** : A detailed account of the driving decisions.
5. **Trajectory Waypoints W** : The waypoints outlining the planned trajectory of the ego vehicle.

4.2. Evaluation Metrics

To comprehensively evaluate a model’s performance, we care about its interpretation of the driving scene and the decisions made. Therefore, our evaluation has two aspects: scene description/analysis evaluation and meta-action evaluation.

Scene description/analysis evaluation. Given the subjective nature of human evaluation in scene description, we adopt a structured approach using a pre-trained LLM. This method entails comparing the generated scene description with a human-annotated ground truth description. The ground truth description encompasses structured data such as environmental conditions, navigation, lane information, and critical events with specific objects, verbs, and their influences. The LLM assesses and scores the generated descriptions based on their consistency with the ground truth.

Meta-action evaluation. Meta-actions are a predefined set

of decision-making options. A driving decision is formulated as a sequence of meta-actions. Our evaluation method employs a dynamic programming algorithm to compare the model-generated sequences with a manually annotated ground truth sequence. The evaluation should also weigh the relative importance of various meta-actions, designating some as ‘conservative actions’ with a lower impact on the sequence’s overall context. To increase robustness, we first use the LLM to generate semantically equivalent alternatives to the ground truth sequence to enhance robustness. The sequence with the highest similarity to these alternatives calculates the final driving decision score. More details of the proposed metric are available in the Appendix B.

4.3. Dataset Construction

We propose a comprehensive data mining and annotation pipeline, shown in Figure 3, to construct a *Scene Understanding for Planning (SUP-AD) Dataset* for the proposed task. Specifically, we perform long-tail object mining and challenging scenario mining from a large database to collect samples, then we select a keyframe from each sample and further perform scene annotation. Dataset statistics are available in the Appendix A.

Long-tail object mining. According to real-world road object distribution, we first define a list of long-tail object categories, such as weird-shaped vehicles, road debris, and animals crossing the road. Next, we mine these long-tail scenarios using a CLIP-based search engine, capable of mining driving data using language queries from a large collection of logs. Following that, we perform a manual inspection to filter out scenes inconsistent with the specified categories.

Challenging scenario mining. In addition to long-tail objects, we are also interested in challenging driving scenarios,

where the driving strategy of the ego vehicle needs to be adapted according to the changing driving conditions. These scenarios are mined according to the variance of the recorded driving maneuvers.

Keyframe selection. Each scene is a video clip, it is essential to identify the ‘keyframe’ to annotate. In most challenging scenarios, a keyframe is the moment before a significant change in speed or direction is required. We select this keyframe 0.5s to 1s earlier than the actual maneuver, based on comprehensive testing, to guarantee an optimal reaction time for decision-making. For scenes that do not involve changes in driving behavior, we select a frame that is relevant to the current driving scenario as the keyframe.

Scene annotation. We employ a group of annotators to perform the scene annotation, including scene description, scene analysis, and planning, except for waypoints, which can be auto-labeled from the vehicle’s IMU recordings. To facilitate scene annotation, we make a video annotation tool with the following features: (1) the annotators can slide the progress bar back and forth to replay any part of a video; (2) while annotating a keyframe, the annotator can draw bounding boxes on the image together with language descriptions; (3) annotators can select from a list of action and decision candidates while annotating driving plans. Each annotation is meticulously verified by 3 annotators for accuracy and consistency, ensuring a reliable dataset for model training. Figure 2 illustrates a sample scenario with detailed annotations.

5. Experiments

5.1. Settings

5.1.1 Datasets

SUP-AD dataset. The SUP-AD dataset is a dataset built by our proposed data mining and annotation pipeline. It is divided into train, validation, and test splits with a ratio of 7.5 : 1 : 1.5. We train models on the training split and use our proposed scene description and meta-action metrics to evaluate model performance on the validation/test split.

nuScenes dataset. The nuScenes dataset [6] is a large-scale driving dataset of urban scenarios with 1000 scenes, where each scene lasts about 20 seconds. Keyframes are evenly annotated at a frequency of 2Hz over the entire dataset. Following previous works [22, 24], we adopt Displacement Error (DE) and Collision Rate (CR) as metrics to evaluate models’ performance on the validation split.

5.1.2 Base Model

We use Qwen-VL [1] as our default large vision-language model, which exhibits remarkable performance in tasks like question answering, visual localization, and text recognition. It contains a total of 9.6 billion parameters, including a

Table 1. **Results on the test set of our proposed SUP-AD dataset.** †: Using the official API of GPT-4V. For Lynx and CogVLM, we utilize the training split for fine-tuning purposes. In contrast, for GPT-4V, we employ in-context learning.

Method	Scene Description	Meta-actions
Fine-tuning w/ Lynx [54]	0.46	0.15
Fine-tuning w/ CogVLM [44]	0.49	0.22
GPT-4V† [36]	0.38	0.19
DriveVLM w/ Qwen	0.71	0.37

Table 2. **Planning results on the nuScenes validation dataset.** DriveVLM-Dual achieves the best performance. † means using perception and occupancy prediction results from Uni-AD. ‡ denotes cooperating with VAD [24]. All the models take ego states as input.

Method	L2 (m) ↓				Collision (%) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
NMP† [53]	-	-	2.31	-	-	-	1.92	-
SA-NMP† [53]	-	-	2.05	-	-	-	1.59	-
FF† [20]	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO† [25]	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [21]	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD [22]	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
VAD-Base [24]	0.17	0.34	0.60	0.37	0.07	0.10	0.24	0.14
DriveVLM	0.18	0.34	0.68	0.40	0.10	0.22	0.45	0.27
DriveVLM-Dual†	0.17	0.37	0.63	0.39	0.08	0.18	0.35	0.20
DriveVLM-Dual‡	0.15	0.29	0.48	0.31	0.05	0.08	0.17	0.10

visual encoder (1.9 billion), a vision-language adapter (0.08 billion), and a large language model (Qwen, 7.7 billion). Images are resized to a resolution of 448×448 before being encoded by the vision encoder. During training, we randomly select a sequence of images at the current time T s, $T - 1$ s, $T - 2$ s, and $T - 3$ s as input. The selected images ensure the inclusion of the current time frame and follow an ascending chronological order.

5.2. Main Results

SUP-AD. We present the performance of our proposed DriveVLM with several large vision-language models and compare them with GPT-4V, as shown in Table 1. DriveVLM, utilizing Qwen-VL as its backbone, achieves the best performance due to its strong capabilities in question answering and flexible interaction compared to the other open-source VLMs. Although GPT-4V exhibits robust capabilities in vision and language processing, its inability to undergo fine-tuning, restricting it solely to in-context learning, often results in the generation of extraneous information during scene description tasks. Under our evaluation metric, the additional information is frequently classified as hallucination, consequently leading to lower scores.

nuScenes. As shown in Table 2, DriveVLM-Dual achieves state-of-the-art performance on the nuScenes planning task

Table 3. **Ablations of design choices on the validation set of nuScenes.** “Base” refers to only indicating the hierarchical planning results without our proposed CoT inference. “CO” represents the addition of critical object analysis. “3D” denotes the inclusion of 3D perception results as an auxiliary language prompt.

ID	Base	CO	3D	L2 (m) ↓				Collision (%) ↓			
				1s	2s	3s	Avg.	1s	2s	3s	Avg.
1	✓			0.19	0.41	0.89	0.49	0.16	0.28	0.63	0.36
2	✓	✓		0.20	0.38	0.75	0.44	0.15	0.29	0.61	0.35
3	✓	✓	✓	0.18	0.34	0.68	0.40	0.10	0.22	0.45	0.27

Table 4. **Inference speed on the NVIDIA Orin platform.** “Trad. AD” represents end-to-end traditional autonomous driving method similar to VAD [24].

Method	Trad. AD	DriveVLM	DriveVLM-Dual
Inference time per scene	0.3s	1.5s	0.3s

when cooperating with VAD. It demonstrates that our method, although tailored for understanding complex scenes, also excels in ordinary scenarios. Note that DriveVLM-Dual significantly improves over UniAD: it achieves a reduction of 0.64 meters in terms of average planning displacement error, and a 51% reduction of collision rate.

5.3. Ablation Study

Model Design. To better understand the significance of our designed modules in DriveVLM, we conduct ablations on different combinations of modules, as shown in Table 3. The inclusion of critical object analysis enables our model to identify and prioritize important elements in the driving environment, enhancing the decision-making accuracy for safer navigation. Integrating 3D perception data, our model gains a refined understanding of the surroundings, which is crucial for capturing the motion dynamics and improving trajectory predictions.

Inference speed. The inference speed of DriveVLM and DriveVLM-Dual are measured on the NVIDIA Orin platform, shown in Table 4. Due to the huge number of parameters of LLM, the inference speed of DriveVLM is an order slower than the conventional autonomous driving method similar to VAD, preventing it from running onboard. However, after cooperating with the traditional autonomous driving pipeline in a slow-fast cooperation pattern, the overall latency depends on the speed of the fast branch, making DriveVLM-Dual an ideal solution for real-world deployment.

5.4. Qualitative Results

Qualitative results of DriveVLM are shown in Figure 4. In Figure 4a, DriveVLM accurately predicts the current scene conditions and incorporates well-considered planning deci-

sions regarding the cyclist approaching us. In Figure 4b, DriveVLM effectively comprehends the gesture of the traffic police ahead, signaling the ego vehicle to proceed, and also considers the person riding a tricycle on the right side, thereby making sensible driving decisions. These qualitative results demonstrate our model’s exceptional ability to understand complex scenarios and make suitable driving plans. More visualization of our model’s output is shown in the Appendix C.

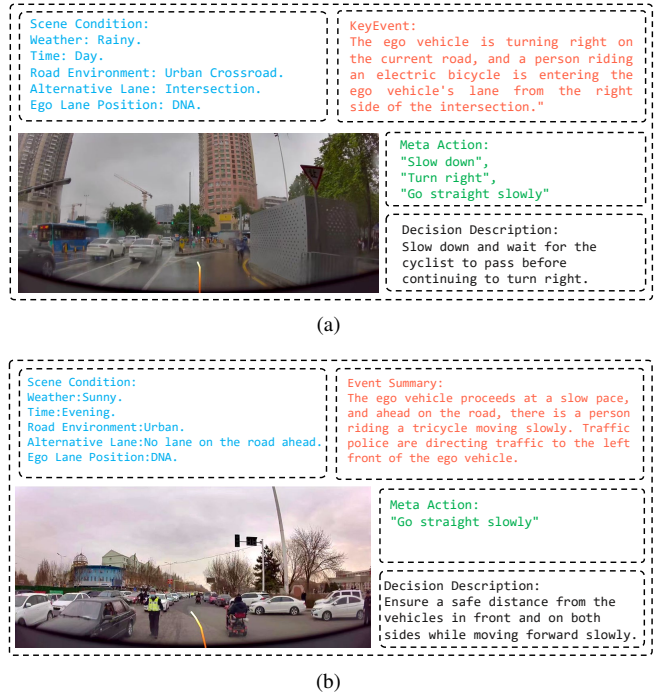


Figure 4. **Qualitative results of DriveVLM.** The orange curves represent the model’s planned future trajectories for the next 3 seconds.

6. Conclusion

In summary, we introduce DriveVLM and DriveVLM-Dual. DriveVLM leverages VLMs, significantly progressing in interpreting complex driving environments. The DriveVLM-Dual further enhances these capabilities by synergizing existing 3D perception and planning approaches, effectively addressing the spatial reasoning and computational challenges inherent in VLMs. Moreover, we define a scene understanding for planning task for autonomous driving, together with evaluation metrics and dataset construction protocol. Through rigorous evaluation, DriveVLM and DriveVLM-Dual have demonstrated their ability to surpass state-of-the-art methods in autonomous driving, especially in handling intricate and dynamic scenarios. We believe this research offers a roadmap for the development of safe and interpretable autonomous vehicles in the future.

References

- [1] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 2, 7
- [2] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 1
- [3] Istvan Barabas, Adrian Todoruț, N Cordoş, and Andrea Molea. Current challenges in autonomous driving. In *IOP conference series: materials science and engineering*, page 012096. IOP Publishing, 2017. 1
- [4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2
- [5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Hong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023. 2
- [6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 7
- [7] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021. 2
- [8] Raphael Chekroun, Marin Toromanoff, Sascha Hornauer, and Fabien Moutarde. Gri: General reinforced imitation and its application to vision-based autonomous driving. *Robotics*, 12(5):127, 2023. 2
- [9] Dian Chen, Vladlen Koltun, and Philipp Krähenbühl. Learning to drive from a world on rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15590–15599, 2021. 2
- [10] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023. 2
- [11] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019. 2
- [12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 2
- [13] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2car: Taking control of your self-driving car. *arXiv preprint arXiv:1909.10838*, 2019. 2
- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3
- [15] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model, 2023. 2
- [16] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1
- [17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2
- [18] Junru Gu, Chen Sun, and Hang Zhao. Densentnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021. 1
- [19] Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023. 2
- [20] Peiyun Hu, Aaron Huang, John Dolan, David Held, and Deva Ramanan. Safe local motion planning with self-supervised freespace forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12732–12741, 2021. 2, 7
- [21] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, pages 533–549. Springer, 2022. 7
- [22] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai

- Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. [2](#), [7](#)
- [23] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. [2](#)
- [24] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. *arXiv preprint arXiv:2303.12077*, 2023. [7](#), [8](#)
- [25] Tarasha Khurana, Peiyun Hu, Achal Dave, Jason Ziglar, David Held, and Deva Ramanan. Differentiable raycasting for self-supervised occupancy forecasting. In *European Conference on Computer Vision*, pages 353–369. Springer, 2022. [2](#), [7](#)
- [26] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017. [2](#)
- [27] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018. [2](#)
- [28] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. [1](#)
- [29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. [2](#)
- [30] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022. [1](#)
- [31] Zenan Li, Fan Nie, Qiao Sun, Fang Da, and Hang Zhao. Uncertainty-aware decision transformer for stochastic driving environments. *arXiv preprint arXiv:2309.16397*, 2023. [1](#)
- [32] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. [1](#), [2](#)
- [33] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021. [1](#)
- [34] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. Drama: Joint risk localization and captioning in driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1043–1052, 2023. [2](#)
- [35] Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2980–2987. IEEE, 2023. [1](#)
- [36] OpenAI. Gpt-4 technical report, 2023. [7](#), [14](#)
- [37] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. [2](#)
- [38] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988. [1](#), [2](#)
- [39] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. [1](#)
- [40] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. *arXiv preprint arXiv:2305.14836*, 2023. [2](#)
- [41] Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations. *arXiv preprint arXiv:2210.14222*, 2022. [2](#)
- [42] Enna Sachdeva, Nakul Agarwal, Suhas Chundi, Sean Roelofs, Jiachen Li, Behzad Dariush, Chiho Choi, and Mykel Kochenderfer. Rank2tell: A multimodal driving dataset for joint importance ranking and reasoning. *arXiv preprint arXiv:2309.06597*, 2023. [2](#)
- [43] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020. [2](#)
- [44] Weihang Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. [2](#), [7](#)
- [45] Yue Wang, Vitor Campagnolo Guizilini, Tianyu Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. [1](#)
- [46] Bob Wei, Mengye Ren, Wenyuan Zeng, Ming Liang, Bin Yang, and Raquel Urtasun. Perceive, attend, and drive: Learning spatial attention for safe self-driving. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4875–4881. IEEE, 2021. [2](#)
- [47] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022. [3](#)
- [48] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14633–14642, 2023. [2](#)
- [49] Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for

- autonomous driving. *arXiv preprint arXiv:2309.04379*, 2023. [2](#)
- [50] Yiran Xu, Xiaoyin Yang, Lihang Gong, Hsuan-Chu Lin, Tz-Ying Wu, Yunsheng Li, and Nuno Vasconcelos. Explainable object-induced action decision for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9523–9532, 2020. [2](#)
- [51] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023. [1](#), [2](#)
- [52] Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023. [2](#)
- [53] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019. [2](#), [7](#)
- [54] Yan Zeng, Hanbo Zhang, Jiani Zheng, Jiangnan Xia, Guoqiang Wei, Yang Wei, Yuchen Zhang, and Tao Kong. What matters in training a gpt4-style language model with multi-modal inputs? *arXiv preprint arXiv:2307.02469*, 2023. [1](#), [2](#), [7](#)
- [55] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition, 2023. [2](#)
- [56] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. Tnt: Target-driven trajectory prediction. In *Proceedings of the 2020 Conference on Robot Learning*, pages 895–904. PMLR, 2021. [1](#)
- [57] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. [1](#), [2](#)

A. SUP-AD Dataset

A.1. Meta-actions

Meta-action statistics. We use the meta-action sequence to formally represent the driving strategy. Meta actions are classified into 17 categories. We show the distribution of each meta-action being the first/second/third place in the meta-action sequence, as shown in Figure 5. It indicates that the meta-actions are quite diverse in the SUP-AD dataset. We also show the distribution of the length of meta-actions per scene in Figure 6. Most scenes contain two or three meta-actions, and a few scenes with complex driving strategies contain four or more meta-actions.

Annotation of meta-actions. The meta-action sequence for each driving scene is manually annotated based on the actual driving strategy in the future frames. These meta-actions are designed to encompass a complete driving strategy and are structured to be consistent with the future trajectory of the ego vehicle. They can be divided into three primary classes:

1. **Speed-control actions.** These actions can be discerned from acceleration and braking signals within the ego state data. They include *speed up*, *slow down*, *slow down rapidly*, *go straight slowly*, *go straight at a constant speed*, *stop*, *wait*, and *reverse*.
2. **Turning actions.** Deduced from steering wheel signals, these actions consist of *turn left*, *turn right*, and *turn around*.
3. **Lane-control actions.** Encompassing lane selection decisions, these actions are derived from a combination of steering wheel signals and either map or perception data. They involve *change lane to the left*, *change lane to the right*, *shift slightly to the left*, and *shift slightly to the right*.

A.2. Scenario Categories

As shown in Figure 7, the SUP-AD Dataset encompasses diverse driving scenarios, spanning over 40 categories. Detailed explanations for certain scenario categories are provided below:

AEB Data: Automatic Emergency Braking (AEB) data.

Road Construction: A temporary work zone with caution signs, barriers, and construction equipment ahead.

Close-range Cut-ins: A sudden intrusion into the lane of the ego vehicle by another vehicle.

Roundabout: A type of traffic intersection where vehicles travel in a continuous loop.

Animals Crossing Road: Animals crossing the road in front of the ego vehicle.

Braking: Brake is pressed by human driver of the ego vehicle.

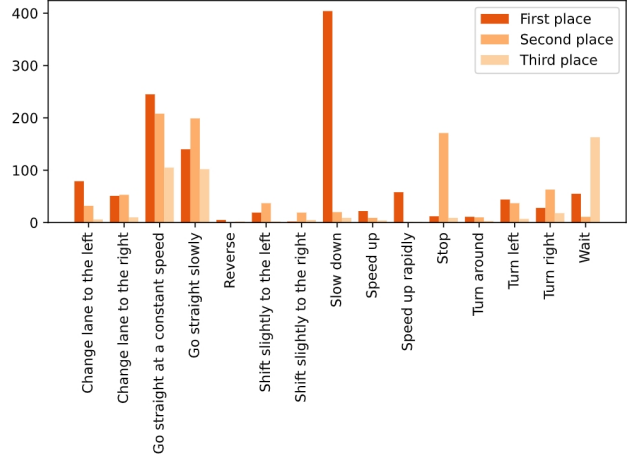


Figure 5. Distribution of each meta action being the first, second, and third place of the meta action sequence, respectively.

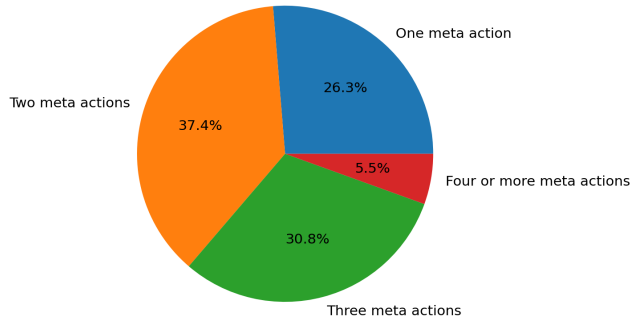


Figure 6. Distribution of the length of meta actions per scene.

Traffic Police Officers: Traffic police officers managing and guiding traffic.

Blocking Traffic Lights: A massive vehicle obscuring the visibility of the traffic signal.

Cutting into Other Vehicle: Intruding into the lane of another vehicle ahead.

Ramp: A curved roadway that connects the main road to the branch road in highway.

Debris on the Road: Road with different kinds of debris.

Narrow Roads: Narrow roads that require cautious navigation.

Pedestrians Popping Out: Pedestrians popping out in front of the ego vehicle, requiring slowing down or braking.

People on Bus Posters: Buses with posters, which may interfere the perception system.

Merging into High Speed: Driving from a low-speed road into a high-speed road, requiring speeding up.

Barrier Gate: Barrier gate that can be raised obstructing the road.

Fallen Trees: Fallen trees on the road, requiring cautious navigation to avoid potential hazards.



Figure 7. Diverse driving scenarios in the SUP-AD dataset.

Complex Environments: Complex driving environments that requiring cautious navigation.

Mixed Traffic: A congested scenario where cars, pedestrians, and bicycles appear on the same or adjacent roadway.

Crossing Rivers: Crossing rivers by driving on the bridge.

Screen: Roads with screens on one side, which may interfere the perception system.

Herds of Cattle and Sheep: A rural road with herds of cattle and sheep, requiring careful driving to avoid causing distress to these animals.

Vulnerable Road Users: Road users which are more susceptible to injuries while using roads, such as pedestrians, cyclists, and motorcyclists.

Road with Gallet: A dusty road with gallet scattered across the surface.

The remaining scenario categories are: Motorcycles and Trikes, Intersection, People carrying Umbrella, Vehicles Carrying Cars, Vehicles Carrying Branches, Vehicles with Pipes, Strollers, Children, Tunnel, Down Ramp, Sidewalk Stalls, Rainy Day, Crossing Train Tracks, Unprotected U-turns, Snowfall, Large Vehicles Invading, Falling Leaves, Fireworks, Water Sprinklers, Potholes, Overtaken Motorcycles, Self-ignition and Fire, Kites, Agricultural Machinery.

A.3. Annotation Examples

We provide more examples of annotation contents in Figure 8, 9, 10, 11, 12, and 13. The scenario categories of these examples are overturned bicycles and motorcycles, herds of cattle and sheep, collapsed trees, crossing rivers, barrier gate, and snowfall respectively.

B. Evaluation Method

The ability of an autonomous driving system to accurately interpret driving scenes and make logical, suitable decisions is of paramount importance. As presented in this paper, the evaluation of VLMs in autonomous driving concentrates on two primary components: the evaluation of scene description/analysis and the evaluation of meta-actions.

B.1. Scene Description/Analysis Evaluation

In terms of scene description/analysis evaluation, the process of interpreting and articulating driving scenes is subject to inherent subjectivity, as there are numerous valid ways to express similar descriptions textually, which makes it difficult to effectively evaluate the scene description using a fixed metric. To overcome this challenge, we utilize GPT-4 [36] to evaluate the similarity between the scene descriptions generated by the model and the manually annotated ground truth. Initially, we prompt GPT-4 to extract individual pieces of information from each scene description. Subsequently, we score and aggregate the results based on the matching status of each extracted piece of information.

The ground truth labels for scene descriptions encompass both environment descriptions and event summaries. Environmental condition description includes weather conditions, time conditions, road environment, and lane conditions. Event summaries are the characteristics and influence of critical objects. We employ GPT-4 to extract unique key information from both environment descriptions and event summaries. The extracted information is then compared and quantified. Each matched pair is assigned a score, which is estimated based on the extent of the matching, whether complete, partial, or absent. Instances of hallucinated information incur a penalty, detracting from the overall score. The aggregate of these scores constitutes the scene description score.

$$\text{Score} = \frac{1.0 \times n_{\text{matched}} + 0.5 \times n_{\text{partial}}}{\frac{n_{\text{gt}}}{0.25 \times n_{\text{hallucination}}}} \quad (4)$$

The prompt for GPT-4 in evaluating scene descriptions is carefully designed, as shown in Table 5. Initially, a role prompt is employed to establish as an intelligent and logical evaluator, possessing a comprehensive understanding of appropriate driving styles. This is followed by specifying the input format, which informs GPT-4 that its task involves comparing an output description with a ground truth description. This comparison is based on the extraction and analysis of key information from both descriptions. Lastly, the prompt outlines the criteria for scoring, as well as the format for the evaluation output, ensuring a structured and systematic approach to the evaluation process.

B.2. Meta-action Evaluation

The evaluation process for the meta-action sequence must consider both the quantity and the sequential arrangement of the matched meta-actions. We employ dynamic programming to compare the model’s output and the annotated ground truth. Our dynamic programming approach is similar to the method utilized in identifying the longest common subsequence, albeit with two supplementary considerations.

The first consideration acknowledges the unequal weighting of different meta-actions. For instance, certain meta actions such as “Slow Down”, “Wait”, and “Go Straight Slowly” exhibit a greater emphasis on attitude rather than action. The presence or absence of these actions from a meta-action sequence does not alter the basic semantic essence of driving decisions but rather modifies the driving strategy to be either more assertive or more cautious. For example, a meta action sequence of “Slow Down -> Stop -> Wait” conveys a similar driving decision as a sequence with only the meta action “Stop”. Consequently, these sequences should not incur a penalty comparable to other meta actions such

System Prompt:

You are a smart and logical evaluator with extensive driving experience, and you will try your best to give reasonable and logical evaluation result.

Input Prompt:

Given two driving scenario descriptions, one is the reference description, the other is the output description from a model. Please extract key information from the reference description and use it as comparison standards to compare the proximity of the output descriptions. Additionally, evaluate the completeness of each piece of information. Key information extracted can be classified into environmental conditions and critical events. Environmental condition information includes weather, time, road environment, and lane conditions. A critical event describes the event related to ego vehicle's driving decisions, which is usually given in the format of critical object + verb/adj + influence on ego vehicle. An example of a critical event is like this: A construction zone is blocking the ego lane and the right lane in front, so the car needs to go around to avoid it. Now I will give you the reference description and the output description.

The reference description:

Weather: Sunny. || Time: Day. || Road Environment: Urban. || Lane Options: Left Lane, Own Lane, Right Lane. || Ego Lane Position: Middle Lane.

Critical Events: There are police officers ahead inspecting passing vehicles. The police officer in front of your lane is signaling you to stop for inspection. There are also vehicles queuing in the left lane. The right lane is a merging lane and is currently separated, so you cannot change lanes to the right.

The output description:

Weather: Sunny. || Time: Day.

There is a vehicle driving in the distance in front of you. There are traffic police on the left and right, and the traffic police signal to stop.

Compare the two descriptions using extracted key information and output a similarities score on the scale from 0 to 1.0 and an explanation to describe the logic of calculating the similarity scale. Each matched key information scores 1.0, and each partially matched key information scores 0.5. Hallucinations will be penalized as well. If any key information exists in the output but not in the reference, a score of -0.25 will be assigned. The final score is the total score divided by the number of key information extracted in the reference description. For example, if the total score is 1.5, and the total number of key information in reference description is 3, then the final score would be $1.5/3 = 0.5$. Please output the detailed extraction and evaluation process.

Response:

Reference description key information:

1. Weather: Sunny (environmental conditions)
2. Time: Day (environmental conditions)
3. Road Environment: Urban (environmental conditions)
4. Lane Options: Left Lane, Own Lane, Right Lane (environmental conditions)
5. Ego Lane Position: Middle Lane (environmental conditions)
6. Police officers ahead inspecting vehicles (critical event)
7. The police officer in front of ego vehicle signalling to stop (critical event)
8. Vehicles queuing in the left lane (critical event)
9. Cannot change to the right lane as it's a merging lane and currently separated (critical event)

Output description key information:

1. Weather: Sunny.
2. Time: Day.
3. Vehicle driving in the distance in front of driver
4. Traffic police on the left and right
5. Traffic police signal to stop

By comparing the output to the reference:

- Key information 1, 2, 6, and 7 are matched, each for 1.0 points (weather and time, police officers and their signal to stop)
- Key information 3 in the output is not mentioned in reference, deduction of -0.25 points.

No scores for:

- Key information 3, 4, 5, 8, and 9.

Total score = $(1.0 + 1.0 + 1.0 + 1.0 - 0.25) / 9 = 0.417$.

Table 5. **Evaluation process of scene description by GPT-4.** Our proposed evaluation method is not only capable of extracting and evaluating information from structured scene descriptions but is also applicable to unstructured descriptions.

as ‘‘Turn Left’’ or ‘‘Change Lane to the Right’’. Therefore, these are designated as ‘‘conservative actions’’, and a reduced penalty is applied when they do not match during sequence evaluation.

The second consideration addresses the potential semantic equality among different meta-action sequences. For example, the sequences ‘‘Change Lane to the Left -> Speed Up -> Go Straight At a Constant Speed -> Change Lane to the Right’’ and ‘‘Change Lane to the Left -> Speed Up Rapidly -> Go Straight At a Constant Speed -> Change Lane to the Right’’ might both represent valid approaches to overtaking a slow-speed vehicle ahead. Recognizing that different meta-action sequences might convey similar meanings, we initially use GPT-4 to generate variant sequences that have comparable semantic meanings, in addition to the unique ground truth meta-action sequence, as shown in Table 6. In the subsequent sequence-matching phase of the evaluation, all these variations, together with the manually annotated ground truth, are taken into consideration. The highest-scoring matching is then adopted as the definitive score for the final decision evaluation.

The state of dynamic programming is saved in a 2D matrix, wherein each row corresponds to a meta action in the ground truth action sequence, and each column corresponds to a meta action in the model output action sequence, noted as $S^{r,c}$. The dynamic programming initiates recursive calculations beginning from the first meta action of both sequences. Each element of the 2D matrix encompasses the optimal total score at the current matching position, as well as the preceding matching condition that yielded the optimal matching. In our dynamic programming algorithm, three transition equations govern distinct cases: S_{missing} for missing matching, $S_{\text{redundant}}$ for redundant matching, and S_{matching} for successful matching. Successful matching occurs when the meta action is identical at the r^{th} position in the reference sequence and the c^{th} position in the model-generated sequence. In the case of missing matching, the meta action at the r^{th} position in the reference sequence is unmatched, prompting a comparison with the $r - 1^{\text{th}}$ position in the reference sequence and the c^{th} position in the model-generated sequence. Conversely, redundant matching implies that the meta action at the c^{th} position in the model-generated sequence is unmatched, leading to further examination of the r^{th} position in the reference and the $c - 1^{\text{th}}$ position in the model-generated sequence. The transformation equations for these cases are as follows:

$$\begin{aligned}
 S_{\text{missing}}^{r,c} &= S^{r-1,c} - p_{\text{missing}}, \\
 S_{\text{redundant}}^{r,c} &= S^{r,c-1} - p_{\text{redundant}}, \\
 S_{\text{matching}}^{r,c} &= S^{r-1,c-1} + s_{\text{matching}}, \\
 S^{r,c} &= \max(S_{\text{missing}}^{r,c}, S_{\text{redundant}}^{r,c}, S_{\text{matching}}^{r,c}),
 \end{aligned} \tag{5}$$

where $s_{\text{matching}} = 1.0$ represents the reward score after

a successful matching. If an action considered missing or redundant is classified as a conservative action, the penalties p_{missing} and $p_{\text{redundant}}$ are quantified as half of s_{matching} , i.e., 0.5. Conversely, if an action is not conservative, both penalties are assigned the same magnitude as s_{matching} , i.e., 1.0. This approach is based on the premise that omitting a crucial meta action or inaccurately introducing a non-existent one equally hampers the effectiveness of the action sequence. The final score $Score_{\text{action}}$ should be divided by the length of the selected reference meta-action sequence, formulated as follow:

$$Score_{\text{action}} = \frac{S^{r,c}}{N_r} \tag{6}$$

C. Qualitative Results

To further demonstrate the effectiveness and robustness of our proposed DriveVLM, we provide additional visualization results in Figure 14, 15, 16, 17, and 18. In Figure 14, DriveVLM recognizes the slowly moving vehicle ahead and provides a driving decision to change lanes for overtaking. In Figures 15 and 16, DriveVLM accurately identifies the type of unconventional vehicles and a fallen tree, demonstrating its capability in recognizing long-tail objects. In Figure 17, the traffic police signaling to proceed with hand gestures has been accurately captured by DriveVLM. In Figure 18, DriveVLM successfully recognizes the road environment of a roundabout and generates a planned trajectory with a curved path.

Prompt:

Given a predefined set of action tokens and a sequence of reference driving actions, find all alternative sequences of actions that have similar meanings but choose slightly different action tokens from the predefined action tokens.

Predefined action tokens: [Speed up, Slow down, Speed up rapidly, Slow down rapidly, Go straight slowly, Go straight at a constant speed, Turn left, Turn right, Change lane to the left, Change lane to the right, Shift slightly to the left, Shift slightly to the right, Stop, Wait, Turn around, Reverse].

For example, if the reference driving action sequence is: Slow down -> Shift slightly to the right -> Go straight at a constant speed, alternative action sequences with similar meanings could be:

1. Slow down -> Change lane to the right -> Go straight at a constant speed;
2. Slow down rapidly -> Shift slightly to the right -> Go straight at a constant speed;
3. Slow down -> Change lane to the right -> Go straight slowly;
4. Slow down -> Shift slightly to the right -> Go straight slowly.

Now the reference action sequence is: Slow down -> Shift slightly to the right -> Go straight at a constant speed. What are the alternative action sequences?

Response:

Given the reference sequence: "Slow down -> Shift slightly to the right -> Go straight at a constant speed." Alternative action sequences with similar meanings based on the provided action tokens are: 1. Slow down -> Change lane to the right -> Go straight at a constant speed.

2. Slow down rapidly -> Shift slightly to the right -> Go straight at a constant speed.

3. Slow down -> Change lane to the right -> Go straight slowly.

4. Slow down -> Shift slightly to the right -> Go straight slowly. You've provided this example in your question.

Based on the tokens and the context, these are indeed appropriate alternative sequences.

Table 6. Example of generating alternative action sequences by GPT-4.

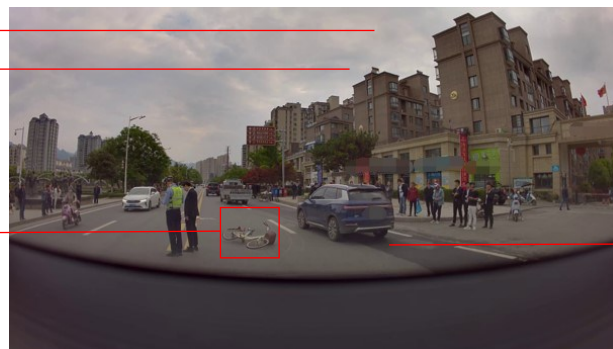
Scene Summary: The ego vehicle changes lanes from the wrong-way lane to the right-way lane, with a bicycle fallen in front.

Weather: Cloudy

Time: Daytime

Critical Object:

Class:	Bicycle
Characteristics:	Fallen in front of the vehicle
Influence:	Blocking vehicle future path



Lane Options: Right Lane

Road Environment: Urban

Meta Action: ["Slow down", "Change lane to the right", "Go straight slowly"]

Decision Description: Slow down and make sure there's no vehicle coming from the right rear before changing lanes to the right.

Figure 8. An example of overturned bicycles and motorcycles in the SUP-AD dataset. A bicycle has fallen in front of the ego vehicle, requiring the ego vehicle to change lanes.

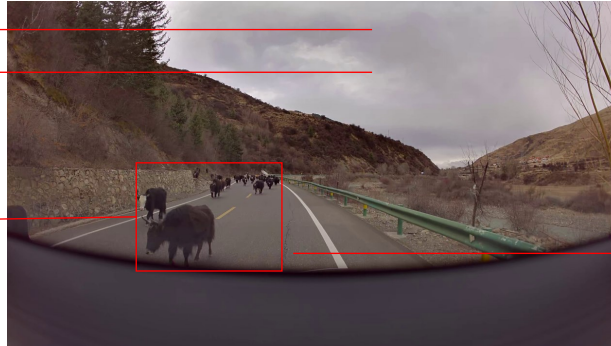
Scene Summary: The ego vehicle is traveling straight in the current lane, and there is a group of slow-moving cows ahead in the lane.

Weather: Cloudy

Time: Daytime

Critical Object:

Class:	A herd of cows
Characteristics:	Slowly moving ahead in the ego lane
Influence:	Influencing the driving speed of the ego vehicle



Lane Options: Own Lane

Road Environment: Mountain

Meta Action: ["Slow down", "Go straight slowly"]

Decision Description: Reduce speed and proceed slowly while maintaining a safe distance from the group of cows.

Figure 9. An example of herds of cattle and sheep in the SUP-AD dataset. A group of cattle move slowly in front of the ego vehicle, requiring the ego vehicle to proceed slowly and maintain a safe distance from the cattle.

Scene Summary: The ego vehicle is moving forward on the current road, and a tree suddenly falls towards the ego vehicle from the left front side.

Critical Object:

Class:	Tree
Characteristics:	Leaning towards our vehicle on the left front
Influence:	Blocking our vehicle from moving forward



Weather: Cloudy

Time: Daytime

Lane Options: Own Lane

Road Environment: National Road

Meta Action: ["Slow down rapidly", "Stop", "Wait"]

Decision Description: Immediately decelerate and come to a stop, wait for the fallen tree to be cleared before resuming driving.

Figure 10. An example of collapsed trees in the SUP-AD dataset. A tree suddenly falls towards the ego vehicle, requiring the ego vehicle to decelerate immediately.

Scene Summary: The ego vehicle travels at a constant speed along the current road towards a bridge ahead. The bridge width allows only a single vehicle to pass.

Critical Object:

Class:	Narrow bridge
Characteristics:	Passable width for only a single vehicle
Influence:	No stopping allowed



Weather: Cloudy

Time: Daytime

Lane Options: No Lane Marking

Road Environment: Narrow Bridge

Meta Action: ["Slow down", "Go straight slowly"]

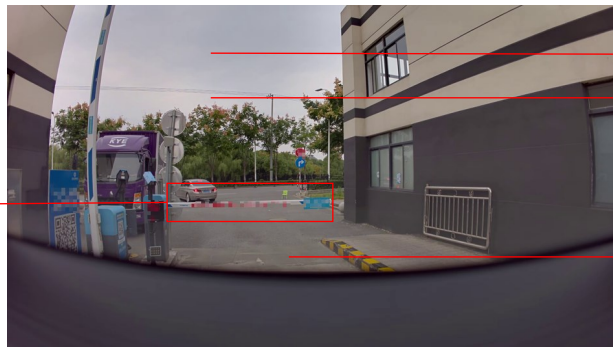
Decision Description: Brake and decelerate, drive slowly towards the bridge without stopping on it.

Figure 11. An example of crossing rivers in the SUP-AD dataset. The ego vehicle is going across a bridge of which width allows only a single vehicle to pass, requiring the ego vehicle to drive without stopping.

Scene Summary: The ego vehicle turns left towards the park entrance, a horizontal bar is blocking the entrance ahead.

Critical Object:

Class:	Crossbar
Characteristics:	At the entrance/exit ahead
Influence:	Blocking the vehicle's driving route



Weather: Cloudy

Time: Daytime

Lane Options: No Lane Marking

Road Environment: Park

Meta Action: ["Slow down", "Stop", "Wait"]

Decision Description: "Slow down and stop in front of the horizontal barrier, waiting for permission to continue."

Figure 12. An example of barrier gate in the SUP-AD dataset. A horizontal barrier blocks the entrance of a park, requiring the ego vehicle to stop and wait for permission to continue.

Scene Summary: The ego vehicle is currently driving slowly along a snow-covered road, following the snow-free tire tracks on the road surface.

Critical Object: None



Weather: Snowy

Time: Daytime

Lane Options: No Lane Marking

Road Environment: Snow-covered Road

Meta Action: ["Go straight slowly"]

Decision Description: "Continue to move forward cautiously at a slow speed."

Figure 13. **An example of snowfall in the SUP-AD dataset.** Most of the road is covered by snow, requiring the ego vehicle to move forward cautiously by following the snow-free tire tracks.

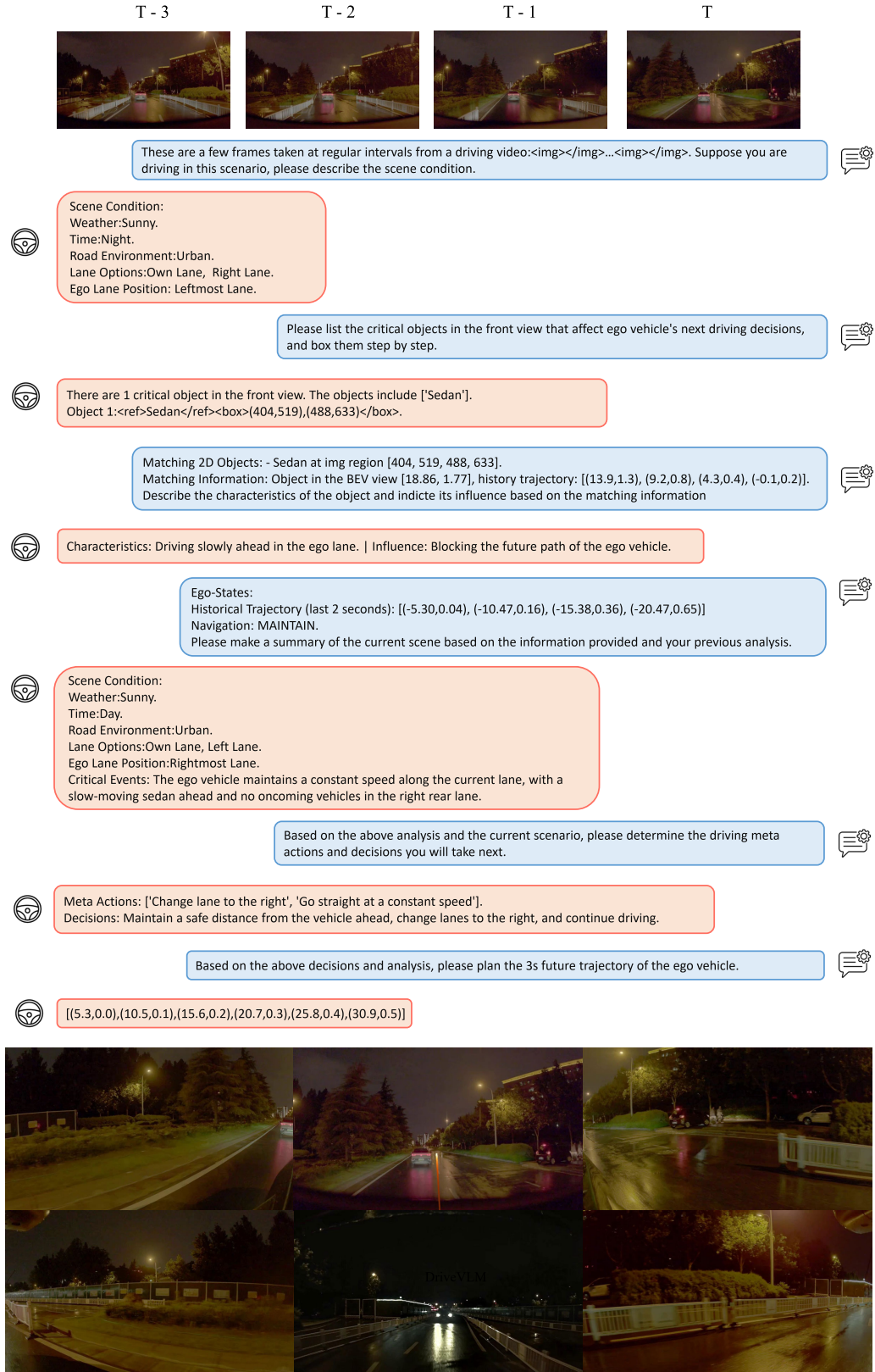


Figure 14. **Visualization of DriveVLM’s output.** DriveVLM recognizes the slowly moving vehicle ahead and provides a driving decision for changing lanes to overtake.



Figure 15. Visualization of DriveVLM's output. DriveVLM has identified the type of the unconventional vehicle ahead.



Figure 16. **Visualization of DriveVLM's output.** DriveVLM precisely detect the fallen tree and its position, subsequently planning an appropriate detour trajectory.



Figure 17. **Visualization of DriveVLM's output.** The traffic police signaling to proceed with hand gestures has been accurately captured by DriveVLM.



Figure 18. **Visualization of DriveVLM’s output.** DriveVLM successfully recognizes the road environment of a roundabout and generates a planned trajectory with a curved path.