

TB-Bench: Training and Testing Multi-Modal AI for Understanding Spatio-Temporal Traffic Behaviors from Dashcam Images/Videos

Korawat Charoenpitaks^{1,*}, Van-Quang Nguyen^{2,*}, Masanori Suganuma¹, Kentaro Arai³,

Seiji Totsuka³, Hiroshi Ino³, Takayuki Okatani^{1,2,*}

¹Tohoku University, ²RIKEN AIP, ³DENSO CORPORATION

Abstract

The application of Multi-modal Large Language Models (MLLMs) in Autonomous Driving (AD) faces significant challenges due to their limited training on traffic-specific data and the absence of dedicated benchmarks for spatiotemporal understanding. This study addresses these issues by proposing TB-Bench, a comprehensive benchmark designed to evaluate MLLMs on understanding traffic behaviors across eight perception tasks from ego-centric views. We also introduce vision-language instruction tuning datasets, TB-100k and TB-250k, along with simple yet effective baselines for the tasks. Through extensive experiments, we show that existing MLLMs underperform in these tasks, with even a powerful model like GPT-4o achieving less than 35% accuracy on average. In contrast, when fine-tuned with TB-100k or TB-250k, our baseline models achieve average accuracy up to 85%, significantly enhancing performance on the tasks. Additionally, we demonstrate performance transfer to another driving benchmark by co-training a model on the other driving benchmark dataset with our proposed dataset. The benchmark, datasets, and code will be available at <https://github.com/TB-AD/TB-Bench>.

1. Introduction

Multi-modal Large Language Models (MLLMs) have been increasingly applied to Autonomous Driving (AD), particularly for predicting risks and planning actions based on images or videos from in-vehicle cameras. Notably, MLLMs have demonstrated their effectiveness in the international competitions like Autonomous Grand Challenge [37] and in specific tasks such as traffic sign detection [58]. However, two major challenges remain.

First, current MLLMs, ranging from proprietary models like GPT-4o [2] and Gemini [42] to open-source models

*Corresponding authors: {korawat, okatani}@vision.is.tohoku.ac.jp, quang.nguyen.jz@riken.jp

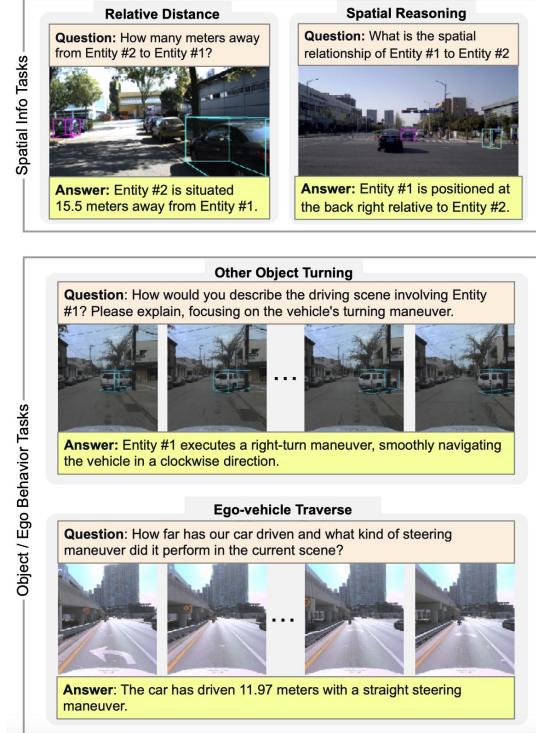


Figure 1. Examples of four tasks from TB-Bench; additional task examples are provided in the supplementary material.

like LLaVA [26], are not optimized for dashcam images or traffic scenes. These models are primarily trained on vast amounts of web-based text and image-text pairs, with minimal traffic-specific data, limiting their effectiveness in AD scenarios. To improve the generalizability of MLLMs, incorporating high-quality domain-specific datasets into the pre-training data is crucial, as shown in [21, 57].

Second, it lacks a dedicated benchmark for evaluating MLLMs' abilities in spatiotemporal understanding tasks, given their capabilities in vision-centric tasks are still developing. While these models are designed to handle diverse vision-language tasks, they struggle with complex visual understanding, such as spatial reasoning and object relation-

ships [43]. Even in common domains unrelated to traffic scenes, there are insufficient benchmarks (e.g., Cambrian-1 [44], V-star benchmark [50]). Given that AD requires sophisticated geometric and spatiotemporal understanding to capture dynamic vehicle-entity interactions, high-quality dedicated benchmarks are needed.

While MLLMs are increasingly applied in AD, the aforementioned challenges remain insufficiently addressed. Recent research has primarily focused on using pretrained MLLMs derived from web data for specific AD tasks, without thoroughly investigating these challenges. Another issue is to determine which AD tasks across different stages and levels should be addressed by MLLMs. Within the “perception, prediction, and planning” framework, the question becomes: which stages should MLLMs handle?

Focusing solely on the perception stage and its associated tasks, it may not necessarily appear optimal to use MLLMs. Established technologies like LIDAR and CV methods such as object detection and visual odometry can accurately capture the vehicle’s position and spatiotemporal relationships in Euclidean space.

However, when considering the use of MLLMs (or LLMs) in later stages, relying on such “Euclidean geometrically accurate” information in the earlier stage might not be optimal. Firstly, it is unclear how to input this information into LLMs. Moreover, achieving advanced understanding in later stages may require information representations specifically suited for MLLMs to perform higher-level tasks. This suggests the need for MLLM involvement from the perception stage.

This study adopts this perspective, aiming to involve MLLMs in the perception stage by expressing spatiotemporal traffic scene tasks in natural language text. The underlying conjecture, as mentioned above, is that this approach could be important for prediction and planning in later stages by MLLMs.

To address the challenge of lacking dedicated benchmarks, we introduce TB-Bench, one of the first comprehensive benchmarks specifically designed to evaluate MLLM’s understanding of traffic behaviors. This benchmark assesses MLLM’s capabilities to perform perception tasks based on dashcam images or videos from the ego-centric views of vehicles, including determining the spatial position or orientation of other vehicles and interpreting the behaviors of both ego-vehicles and surrounding traffic. Compared to existing benchmarks, TB-Bench encompasses a wider array of eight distinct perception tasks,¹ each corresponding to a typical driver maneuver. Figure 1 shows examples of several tasks. To ensure consistent evaluation across a diverse range of MLLMs, we employ a straightforward pro-

tocol. Specifically, we pair questions with images or video clips, requiring an MLLM to respond in plain text. Performance of the MLLM is then assessed by measuring response accuracy.

To address the challenge of insufficient training data for AD perception tasks, we introduce a high-quality dataset focused on traffic behavior understanding from ego-centric views. This dataset aligns with the task design of TB-Bench and is used for vision-language instruction tuning (VLIT) of MLLMs. We generate high-quality question-and-answer pairs using samples from established datasets such as KITTI, ONCE, and Argoverse 2. Specifically, we constructed **TB-Bench**, which consists of 2,000 manually created samples, along with two versions of training datasets: **TB-250k**, containing 250,000 samples, and **TB-100k**, a more balanced version with 100,000 samples.

In addition to evaluating existing MLLMs, we introduce a generic framework that serves as a strong baseline for our tasks, consisting of three standard components: a pre-trained vision encoder, a multi-modal connector, and a pre-trained LLM. The vision encoder extracts visual representations from inputs with varying number of frames, while the connector projects these embeddings into the LLM’s embedding space, finally the LLM generates task-specific responses on our benchmark. This lightweight model is designed for efficient fine-tuning on our proposed dataset(s).

Using TB-Bench to evaluate popular proprietary models (GPT-4o and Gemini) and various state-of-the-art open-source MLLMs (LLaVA, Bunny, and InternVL), we find that none of these models excels across all traffic behavior understanding tasks. On average, the open-source models underperform random guessing, while proprietary models achieve only slightly better results, with average accuracy below 35%. In contrast, when fine-tuned on TB-100k or TB-250k, our proposed baseline models demonstrate strong performance across all tasks, with average accuracy ranging from 77% to 85%. This highlights the effectiveness of our dataset in enhancing MLLM traffic behavior understanding.

Overall, our contributions are fourfold: 1) we introduce TB-Bench, a benchmark for assessing MLLMs on eight perception tasks of traffic behavior understanding; 2) we present the VLIT datasets (TB-100k and TB-250k) for the tasks, along with a generic baseline; 3) we conduct extensive experiments demonstrating the performance gap between existing MLLMs and the fine-tuned baselines; and 4) we show that our proposed dataset would be used as part of co-training datasets to generalize and improve the performance on other driving benchmarks, such as BDD-X [19].

2. Related Work

A summary of existing studies and benchmarks across various AD tasks is presented in Table 1.

¹Eight proposed tasks: Relative Distance, Spatial Reasoning, Orientation Reasoning, Other Lane to Ego, Other Lane Changing, Other Turning, Ego Turning, and Ego Traverse Distance.

Table 1. Summary of existing studies and benchmarks across AD tasks (brackets indicate tasks involving planning).

Benchmarking	Visual Data Modality	Perception (Planning) Tasks	Abbreviation	Meaning
Standalone Task in AD				
DRAMA [29]	single-image	PER, REA	OD	2D & 3D Object Detection
Rank2Tell [38]	single-image	PER, REA, LANE, TLS	OT	2D & 3D Object Tracking
BDD-X [19]	multi-frame	PER, (AC)	D	Depth Estimation
BDD-OIA [52]	single-image	PER	OBJ	Object Existence, Class, etc.
TrafficQA [51]	multi-frame	PER, PRED, REA	KNOW	World Knowledge
LingoQA [31]	multi-frame	PER, PRED, REA	LOC	Location or Coordinate
NuScenes-QA [35]	multi-view	OBJ, SP	LANE	Road, Lane, Intersection, etc
NuScenes-MQA [17]	multi-view	OBJ, RD, OD	PER	General Perception
MAPLM-QA [5]	multi-view, BEV-image	LANE	PRED	General Prediction
DriveLM [40]	single-image	PER, PRED, (PLAN)	PLAN	General Planning
Benchmark				
SpatialRGPT [7]	single-image	RD, SR, OR	REA	General Reasoning
SEED [20]	multi-image, multi-frame	PER, PRED, REA, AR	TLS	Traffic Light or Sign
MV Bench [23]	multi-frame	PER, PRED, REA, LOC, AR	AC	Action Category
MME [11] / MME-Realworld [59]	single-image	PER, PRED	AR	General Action Recognition
MMMU [55]	multi-image	PER, REA, KNOW	RD	Relative Distance
ELM [60]	multi-frame	PER, PRED, TLS, OD, OT, AR, (PLAN)	SR	Spatial Reasoning
Cambrian-I [44]	single-image	RD, SR, D	OR	Orientation Reasoning
OpenEQA [28]	multi-frame	OBJ, SR, KNOW, LOC, REA	EGO-LANE	Other Lane to Ego-vehicle
TB-Bench (Ours)	single-image, multi-frame	RD, SR, OR, EGO-LANE, OBJ-LANE, OBJ-TURN, EGO-TURN, EGO-TRA	OBJ-LANE	Other Lane Changing
			OBJ-TURN	Other Turning
			EGO-TURN	Ego Turning
			EGO-TRA	Ego Traverse Distance

2.1. Autonomous Driving Tasks

The majority of evaluations in the AD field are focused on either end-to-end driving systems, open-loop planning, or standalone task schemes, such as single-round visual question answering (VQA) or captioning. Traditionally, the AD framework consists of perception, prediction, and planning tasks [33], although slight variations exist, i.e., predicting intention-level outputs instead of trajectories [43].

Generally, perception tasks in end-to-end driving systems are mainly auxiliary tasks, consisting of all available supervision signals provided based on the data source. For example, NuScene [4] provides BEV information, segmentation labels, and more. Consequently, multi-task learning is applied to these tasks, such as object detection, tracking, and segmentation. This approach is consistent across recent similar AD planning datasets, whether in open-loop or simulation scenarios. Occasionally, pretrained VL models are utilized to enhance these modules.

Other popular traffic planning datasets are KITTI [14], ONCE [30], Waymo Open [41], and Argoverse2 [46], which are inherently similar to NuScene in characteristics.

Pretrained VL models are commonly known for their excellence in scene understanding, details, and visual cues. Still, it shows limitations in spatial grounding and reasoning [43]. In detail, most standalone task schemes focus on perception tasks, which include general event VQA [31, 51], environment and weather conditions, traffic signals, and lane information [5, 45]. These tasks also encompass critical object detection [29, 38] or tracking in various forms, such as bounding box coordinates [43], region proposals [9, 52], 2D [48], and 3D [49] language-guided object tracking, as well as scene analysis that includes attributes or motion of objects like size, position, direction, distance, spa-

tial position relationships [35], and orientation [7]. In particular, a comprehensive driving task integrates language with perception and prediction, features multi-choice VQA evaluation [59], and includes planning questions [40]. In the prediction tasks, all previous perception inputs are used to predict the object’s future trajectory, such as parking or moving, and interactions with the ego-vehicle. In the planning stage, it involves combining prior information to generate actions, decision descriptions [52], and trajectory waypoints [40, 43].

2.2. MLLMs and Benchmarks

VL pre-training and foundation models started with learning from a broader source of supervision, specifically raw text at an internet scale [36], enabling zero-shot transfer of the model to downstream tasks. Notably, approaches attempting to connect VL pre-training to existing LLMs, referred to as MLLMs [22], enable capabilities similar to those of LLMs, such as image-to-text generation, improved via instruction tuning and in-context learning capabilities. Current frontier families of MLLMs, such as LLaVA [26], VILA [24], and InternVL [6], utilize a similar architectural paradigm: vision encoder, multi-modal projector, and LLM connected in sequence. Despite some early work attempting resampler techniques like Q-Former [8], all state-of-the-art models use simpler linear layers with scaling to higher resolutions, focusing on higher quality VLIT instead. Another line of studies works on lightweight versions of MLLMs, optimizing for more informative, condensed training data and design choices [15, 39]. The latest MLLMs focus on simultaneously tackling multi-image, multi-frame (video), multi-view (3D), and multi-patch (single-image) scenarios, which show emergent capabilities and enhance overall per-

formance [21]. Nevertheless, it is a standard paradigm for MLLMs to evaluate on multiple general benchmarks, aiming to achieve overall performance.

The existing benchmarks, which refer to MLLM benchmarks, aim to comprehensively evaluate various dimensions, but there is no standardized taxonomy for benchmark design. General benchmarks in the VL space started with simple perception-oriented tasks [11], followed by multi-frame benchmarks [20, 23] with action recognition and VL knowledge-based reasoning [55]. Spatial or vision-centric benchmarks [7, 44] are becoming more relevant to address previously claimed weaknesses. Then, specialized benchmarks gained more attention, introducing tasks from different domains, such as robotics [28] and AD [40, 59]. In this case, there is still a lack of studies covering simple yet very important skills and behaviors in the AD context.

3. Benchmark Design

3.1. Task Design

We generate question-answer pairs in VQA format, where the model takes an image or video with a question as input and produces a corresponding answer. Both question and answer are expressed in a single sentence of free-form text.

To achieve the above goal, we consider multiple types of Q&A pairs, each linked to a specific driver’s maneuver behavior. We refer to the Pre-crash Scenarios typology from the National Automotive Sampling System (NASS) variables [32], which are also utilized in the CARLA simulator [10]. This typology includes a total of 65 pre-crash scenarios, categorized into nine accident types². Each scenario is described in the format of ‘an accident type: a detailed scenario.’ For example, the ‘lane change’ accident type includes scenarios like ‘one vehicle passing while another is turning.’ See the supp. material for the full list of scenarios.

Focusing on typical maneuver behaviors derived from NASS scenarios, we have identified eight distinct Q&A types, referred to as ‘tasks,’ as shown in Table 2. Some tasks require numerical outputs (e.g., ‘distance in meters’), while others require discrete classes (e.g., ‘back,’ ‘back left,’ etc.). It is important to note that the models are expected to provide these outputs in their natural language responses. Fig. 1 presents examples for four of the eight tasks, each of which consists of input image(s) accompanied by a question and a ground-truth answer. The visual input is either a single image or multiple images (up to eight), depending on the task, as will be explained later.

3.2. Referencing Entities

Some tasks require the model to determine the spatial position or orientation of other vehicles, as shown in Fig. 1.

²The accident types are Animal, Off-road, Pedalcyclist, Pedestrian, Backing, Lane Change, Opposite Direction, Rear-end, and Crossing-paths.

Table 2. **Tasks and Concepts Addressed in Each.** ‘Classes’ column indicates the types of outputs, i.e., the number of discrete classes or numerical outputs (indicated by \mathcal{R}); ‘Orientation Reasoning’ task contains both output types.

Task Type	Abstract Concepts	Classes
Spatial Information:		
Relative Distance	distance in meters	\mathcal{R}
Spatial Reasoning	back, back left, back right, front, front left, front right	6
Orientation Reasoning	opposite, perpendicular, similar, and degrees	$3/\mathcal{R}$
Object Behavior:		
Other Lane to Ego-Vehicle	front lane, front left lane, front right lane, oncoming traffic lane	4
Other Lane Changing	left lane change, no change, right lane change	3
Other Turning	go straight, left turn, right turn	3
Ego Behavior:		
Ego Turning	go straight, left turn, right turn	3
Ego Traverse Distance	distance traveled in meters	\mathcal{R}

When multiple vehicles are present in a scene, it is essential to distinguish between them in both the questions and answers. One approach is to describe the vehicle by its attributes, such as “black compact sedan,” but this can pose challenges in ensuring the model accurately identifies and differentiates similar objects using such descriptions. To avoid these complications and focus on evaluating the model’s spatial understanding, we label each target traffic entity as ‘Entity #n’ in the questions and answers, where n corresponds to its index in the input image(s); see examples in the upper part of Fig. 1. To identify these entities, we draw colored three-dimensional bounding boxes (BBs) directly in the input image(s), using a consistent color for each entity index n throughout the dataset. Specifically, we use cyan and magenta BBs for ‘Entity #1’ and ‘Entity #2,’ respectively. Our dataset includes up to two entities per scene, i.e., $n = 1$ or 2. An additional advantage of this method is that it requires minimal instruction tuning or even no extra learning for MLLMs to adapt. It is compatible with multi-view, multi-frame, and multi-scale modalities, as demonstrated in AnyRes [25], UniRes [57], and Interleave [21].

3.3. Evaluation

Our benchmark requires MLLMs to generate plain text outputs. Since the goal is to evaluate the spatiotemporal understanding capabilities of MLLMs, their output accuracy should be evaluated using methods suited to this goal.

The questions in the dataset are broadly classified into two categories based on the type of answers expected. One category includes questions about positional relationships or orientation, with typical answers like “positioned at the back right” or “a right-turn maneuver.” The other category involves questions requiring numerical answers, such as “is situated 15.53 meters away.”

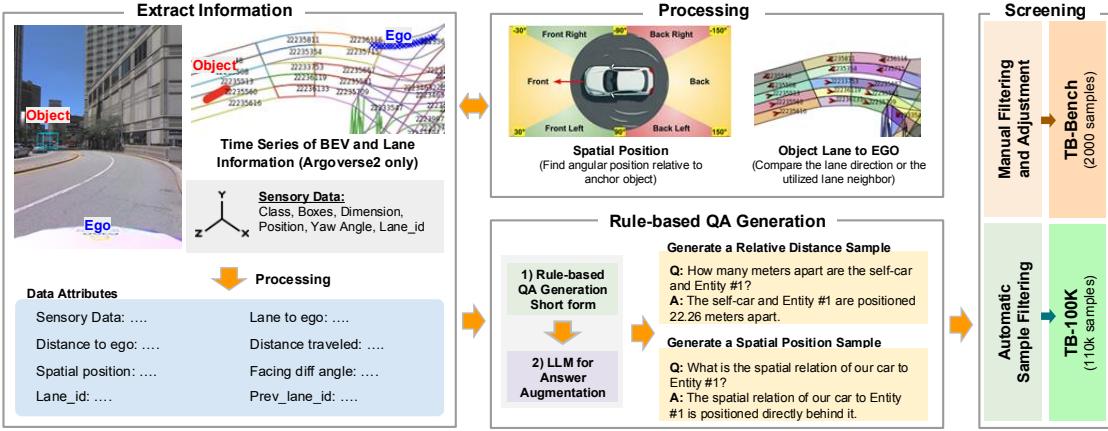


Figure 2. **Overview of Data Generation Pipeline.** Left: Sensory data is processed into higher-level attributes. Middle-Top: Spatial positioning and lane orientation relative to the ego-vehicle are determined. Middle-Bottom: Q&A samples are generated using rules and LLM augmentation. Right: Data is filtered and refined for the final dataset.

For the first category of Q&A, keywords are manually selected for each task or ground truth answer, and their presence in the output text is identified using rule-based methods (i.e., regular expressions). For the second category, the predicted value is compared to the correct answer, and if the difference falls within a specified range, the prediction is considered correct; otherwise, it is deemed incorrect. In the experiments, thresholds are set such that a difference within 25% of the correct value is considered acceptable for distance, and a difference within 15 degrees is acceptable for angle. Refer to the supp. material for more details.

4. Generation of VQA Data

4.1. Outline

To generate Q&A pairs for the eight tasks mentioned, we repurpose existing datasets, specifically KITTI [13], ONCE [30], and Argoverse2 [46]. These datasets are originally designed for studying object detection, localization, and tracking in three-dimensional space, providing detailed three-dimensional geometry of traffic entities. KITTI and ONCE, in particular, offer object class information and 3D bounding boxes for each traffic entity, including their position, dimensions, and yaw angle. Argoverse2 further enriches this with lane information relative to the ego vehicle.

To align with the task design mentioned (Table 2), the quantities provided by these datasets, mostly represented in the Euclidean space, are converted into abstract concepts, such as six discrete angles between two vehicles (e.g., front right, back left, etc.), lanes relative to the ego-car (i.e., front left lane, oncoming lane) and lane changing.

For the first three tasks—‘Relative Distance,’ ‘Spatial Reasoning,’ and ‘Orientation Reasoning’—we generate Q&A pairs using samples from KITTI and ONCE, as these tasks do not require lane information from the ego vehicle

Table 3. **Statistics of TB-Bench, TB-100k, and TB-250K.**
Source datasets: K (KITTI), O (ONCE), Arv2 (Argoverse2).

Task Type	Sources/ Frames	TB- Bench	TB- 250k	TB- 100k
Spatial Information:				
Relative Distance	[K, O]/1	250	35k	10k
Spatial Reasoning	[K, O]/1	250	70k	30k
Orientation Reasoning	[K, O]/1	250	70k	30k
Object Behavior:				
Other Lane to Ego	[Arv2]/8	250	50k	20k
Other Lane Changing	[Arv2]/8	250	1.5k	1.5k
Other Turning	[Arv2]/8	250	1.5k	1.5k
Ego Behavior:				
Ego Turning	[Arv2]/8	250	1.5k	1.5k
Ego Traverse Distance	[Arv2]/8	250	25k	15.5k
Total		2000	254k	110k

or others. Since these tasks can be performed using a single image, we utilize a static dashcam image as the visual input. For the remaining tasks—‘Other Lane to Ego,’ ‘Other Lane Changing,’ ‘Other Turning,’ ‘Ego Turning,’ and ‘Ego Traverse Distance’—which require lane information and a multi-frame source, we generate Q&A pairs using Argoverse2. Given that these tasks involve temporal changes, we extract eight image frames from the ‘long scenario’ sequences in the dataset for each Q&A pair³, using these sequences as the visual input for models.

After generating the data automatically, we conduct a manual screening process. Based on the extent of screening, the data is organized into three distinct datasets. One dataset, comprising 2,000 samples, is designated for evaluation purposes, which we will refer to as ‘benchmark’ in this paper. These samples undergone thorough manual inspection, removing low-quality samples and ensuring an equal

³Each ‘long scenario’ sequence in the dataset is 15 seconds long. From these, we extract 1.6-second clips, consisting of eight images captured at 0.2-second intervals.

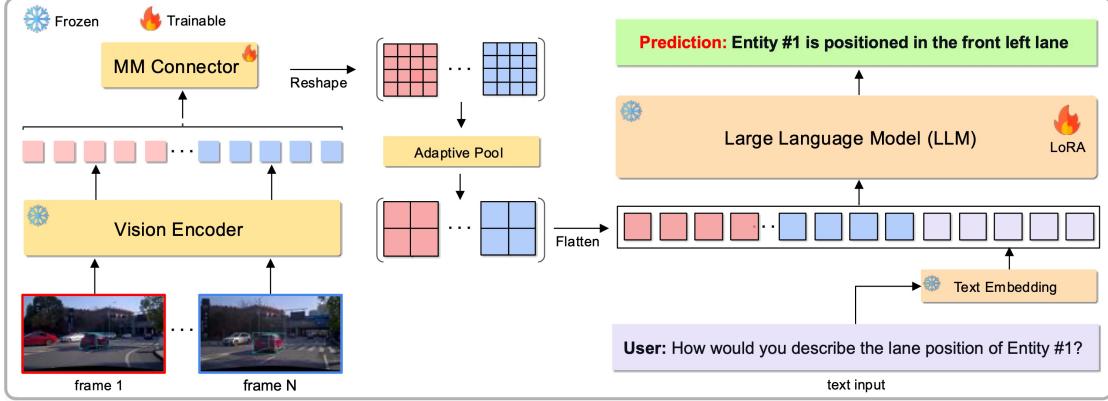


Figure 3. The overall architecture of our baseline framework.

number (i.e., 250) of samples per task. The remaining two datasets are intended for model training: the first, TB-250k, contains 250,000 samples; the second, TB-100k, includes over 100,000 samples that have been filtered to balance the number of samples per task. Table 3 summarizes the overall statistics of these datasets.

4.2. Details of the Pipeline

The Q&A pairs are generated automatically, with manual inspection following the automated process. The only exception is the ‘Other Lane Changing’ task, where we manually generate Q&A pairs due to noisy lane information at intersections. Figure 2 illustrates the pipeline used for generating Q&A pairs from these datasets.

The process unfolds as follows: The input to the pipeline is a single sample from the datasets, which could be either a static image with a set of entity attributes from KITTI/ONCE or a list of sequences with similar data from Argoverse2. The pipeline begins by extracting key information from the input, as depicted in the left panel of Fig. 2. This is followed by a processing step shown in the middle-top panel of Fig. 2, where spatial positions and facing angles relative to an anchor object are calculated. Additionally, the ‘lane to ego’ task identifies on which side the entity is located relative to the ego vehicle. For turning behaviors, we record the accumulated turning angle of each object to determine its recent motion. For lane changes, a flag is recorded if there are changes in lane_id compared to the previous step. Similarly, all sensor numerical ground truth data—such as position, dimension, and angle of all entities—are processed into attributed data, such as distance to ego and spatial position.

Finally, a rule-based process, shown in the middle-bottom panel of Fig. 2, is triggered to identify the task and generate Q&A pairs. More details are in the supp. matt.

In the next phase, a rule-based system generates QA samples from processed data attributes. This depends on the type of task, i.e., tasks aside from lane change and turn-

ing behavior can be created based on any frame, without necessarily needing an event to trigger it. Thus, they naturally have more data samples generated. After this, the rule-based QA is generated with simple short answers, such as ‘oncoming lane’ or ‘turn left.’ Then, it is augmented to be a more complex sentence using text-only information with an LLM; we used Microsoft-Phi3-medium [1].

5. Baseline Framework

We present a generic framework that serves as a strong baseline for our tasks, comprising three standard components: a vision encoder, a multi-modal connector (a two-layer MLP), and an LLM. The vision encoder extracts visual representations from input frames, the multi-modal connector projects these representations into the LLM’s embedding space, and finally the LLM generates a response based on the given question and visual embeddings. Figure 3 illustrates the architecture of our framework.

We now explain how to adaptively extract visual representations from varying numbers of frames and input them into the LLM. Given N frames of $H \times W$ having color-coded bounding boxes, the vision encoder processes each frame individually to produce N visual representations of size $[H/p \times W/p, C]$, where p is the patch size and C is the embedding dimension of the encoder. These visual representations are then projected into the LLM’s embedding space of D using the multi-modal connector, resulting in N visual embeddings of size $[H/p \times W/p, D]$.

Inputting all visual embeddings of N frames into the LLM can be computationally expensive. To address this, we sample spatially a subset of these visual embeddings per frame. Specifically, we apply adaptive average pooling to reduce each frame’s embeddings, from $[H/p \times W/p, D]$ to $[k = h \times w, D]$, where $k \ll H/p \times W/p$. The value of k is determined as a hyperparameter. The sampled embeddings from all N frames are then reshaped and concatenated, preserving spatial and temporal order, which yields final visual embeddings of size $[N \times k, D]$ that are passed into the LLM

Table 4. Results of compared methods on TB-Bench are reported in accuracy (%), where higher indicates better performance. Random guess[†] results are considered zero. *In-context learning for single-frame tasks uses three in-context examples, while multi-frame tasks use one. Huggingface and API names are used for easy reference.

Model	TB-Bench Tasks								
	RD ↑	SR ↑	OR ↑	EGO-LANE ↑	OBJ-LANE ↑	OBJ-TURN ↑	EGO-TURN ↑	EGO-TRA ↑	Avg. ↑
Random [†]	0.0	16.7	17.1	25.0	33.3	33.3	33.3	0.0	19.8
Zero-shot									
LLaVA-1.5-7B	10.8	16.8	28.0	28.4	20.4	23.2	16.8	0.0	18.1
LLaVA-v1.6-Mistral-7B	4.0	25.6	30.8	20.4	26.0	22.4	27.2	0.0	19.6
LLaVA-NeXT-Video-7B	3.6	0.8	13.2	10.4	18.8	22.4	30.0	0.0	12.4
LLaVA-Interleave-Qwen-7B	5.6	24.8	10.8	31.6	19.2	26.8	20.4	0.0	17.4
Bunny-v1.1-4B	24.4	20.4	19.6	28.4	16.0	20.0	34.4	0.0	20.4
Bunny-v1.1-Llama-3-8B-V	7.6	16.4	30.0	26.8	18.4	21.6	20.0	1.2	17.8
InternVL2-8B	3.6	12.0	28.0	28.4	28.0	29.2	30.4	0.4	20.0
Magma-8B	1.2	24.8	11.6	34.4	36.8	46.8	42.0	2.8	25.1
Mini-InternVL2-1B-DriveLM	0.0	31.2	20.0	28.4	24.8	47.2	41.6	0.0	24.2
DriveLM-mantis-8B	0.0	34.8	23.2	30.0	57.6	50.8	48.8	0.0	30.7
Gemini-1.5-flash	21.2	16.8	22.0	34.8	48.0	23.2	27.6	4.8	24.8
GPT-4o-2024-08-06	8.4	32.0	40.8	54.4	39.6	43.2	40.4	16.0	34.4
In-context learning*									
LLaVA-Interleave-Qwen-7B	14.0	3.6	10.4	24.8	29.6	19.6	28.0	24.4	19.3
GPT-4o-2024-08-06	32.8	38.8	36.8	60.4	51.2	38.4	46.4	22.8	40.9
VLIT on TB-100k									
Ours (SigLIP-L-Qwen1.5-0.5B)	76.4	74.4	86.8	94.0	68.8	74.8	81.2	63.2	77.5
Ours (SigLIP-L-Qwen2-0.5B)	80.4	74.8	88.8	93.6	65.2	76.4	80.0	60.4	77.5
VLIT on TB-250k									
Ours (SigLIP-L-Qwen1.5-0.5B)	93.6	82.4	96.0	99.6	69.6	80.4	82.0	73.4	84.5
Ours (SigLIP-L-Qwen2-0.5B)	91.2	83.2	94.8	99.6	69.6	80.4	82.8	78.8	85.1

along with the textual embeddings.

To process text input, we tokenize the question and its ground-truth response, converting them into textual embeddings. These are then combined with the visual embeddings and input into the LLM. We train the model by minimizing cross-entropy loss on the response token predictions. During inference, only the question is used as text input.

6. Experiments

6.1. Experimental Settings

Our proposed framework is compatible with any vision encoder and LLM. We utilize pretrained SigLIP-L/14 [56] as the vision encoder and Qwen 0.5B (v1.5 or v2.0) [3, 53] as the LLM, initializing the multi-modal connector randomly. To preserve LLM capabilities and enable efficient fine-tuning, we apply LoRA [16] (rank 64). During training, we freeze the vision encoder and LLM, updating only the multi-modal connector and LoRA adapters.

For tasks requiring temporal information, the number of frames N is 8; otherwise $N = 1$. Each frame is resized to 384×384 as the input to SigLIP-L/14, with the number of sampled visual embeddings k set to 16 (i.e., $h = w = 4$).

We fine-tune our models on TB-100K or TB-250K and report accuracy on TB-Bench. Using AdamW [27] with a $2e-4$ learning rate and batch size 64, we train for 10 epochs with a cosine scheduler.

6.2. Zero-shot Evaluation for MLLMs

We report the zero-shot performance of various MLLMs on TB-Bench, including two popular proprietary models (GPT-4o, Gemini 1.5), several SOTA open-source general models, including LLaVA [26], Bunny [15], InternVL [6] and Magma [54], as well as open-source models with traffic domain adaptations trained on DriveLM [40], i.e., Mantis [18] and Mini-InternVL2 [12]. For class output questions, we use a multi-choice template listing all possible class options, while for numerical output questions, we specify the format, i.e., “Answer in xx.x meters.” See the supp. material for more details on the models and the prompt design.

6.3. Results on TB-Bench

Table 4 shows the results on TB-Bench tasks, categorized into four groups: zero-shot evaluation, in-context learning evaluation, VLIT on TB-100k, and VLIT on TB-250k.

In the zero-shot evaluation, although the proprietary

Table 5. Ablation results on (a) vision encoders, (b) number of visual embeddings per frame, and (c) number of frames.

(a) vision encoder		(b) # tokens/frame		(c) # frames	
Encoder	Acc	# tokens/fr	Acc	# frames	Acc
CLIP-L/14	72.0	4	72.7	2	72.1
SigL-B/16	74.3	16	77.5	4	73.8
SigL-L/14	77.5	36	76.2	8	77.5

Table 6. Cross-dataset generalization results of control signals prediction on BDD-X test dataset. RMSE denotes the root mean squared error, and A_τ measures the proportion of test samples with prediction errors less than τ .

Experiment	Dataset sampling ratio		Speed (m/s)				Turning angle (degree)					
	BDD-X	TB-100k	RMSE↓	$A_{0.1} \uparrow$	$A_{0.5} \uparrow$	$A_{1.0} \uparrow$	$A_{5.0} \uparrow$	RMSE↓	$A_{0.1} \uparrow$	$A_{0.5} \uparrow$	$A_{1.0} \uparrow$	$A_{5.0} \uparrow$
standard training	20	-	1.40	26.1	55.7	75.6	98.6	11.2	44.2	62.2	71.8	89.2
with co-training	20	1	1.38	26.3	57.6	76.1	98.8	11.3	44.5	63.7	73.0	89.3

models (GPT-4o and Gemini) outperform the open-source models overall, none of them excels across all traffic behavior tasks. Many open-source models underperform random guessing, while traffic domain adaptation models show significantly better performance in certain areas but still lag behind the proprietary models. The proprietary models achieve an average accuracy of less than 35%.

In in-context learning, examples significantly improve performance in specific areas, i.e., numerical outputs.

For baseline models fine-tuned on TB-100k, both with Qwen variants demonstrate strong performance across all tasks, with an average accuracy of 77.5%. Even the lowest-performing task exceeds 60% accuracy, showing a significant improvement of over almost 45% compared to GPT-4o and 57% over random chance. This underscores the effectiveness of VLIT when a high-quality dataset is available, enhancing traffic behavior understanding of MLLMs.

For baseline models fine-tuned on TB-250k, performance improves across all tasks, particularly those with increased data samples. Notably, accuracy in tasks like OBJ-LANE, OBJ-TURN, and EGO-TURN, with the same training samples to TB-100k, also benefits from additional samples in other tasks. This suggests that learning from tasks can be transferred to those with limited training data.

6.4. Ablation Study

We conduct an ablation study to identify which factors enhance performance during fine-tuning, regarding visual inputs to the models. All experiments use the same settings unless noted. The results are summarized in Table 5.

Table 5a compares different pretrained vision encoders, including CLIP-L/14 [36] and SigLIP-B/16 (processing 224×224 frames). It is seen that the SigLIP encoders outperform the CLIP encoder, with SigLIP-L/14 achieving the highest accuracy.

Table 5b presents the results of using varying numbers of sampled visual embeddings/tokens per frame, p (where $h = w = \sqrt{p}$). We observe that using 16 sampled visual tokens per frame is optimal.

Finally, we evaluate the impact of varying the number of sampled frames $N (= 2, 4, 8)$, on the tasks requiring temporal information. We consistently select the first and last frames, with the remaining $N - 2$ frames sampled uniformly in between. As shown in Table 5c, increasing temporal in-

formation significantly boosts performance. For more detailed on other ablation results, see the supp. material.

6.5. Cross-Dataset Generalization

We conduct additional experiments to demonstrate performance transfer from co-training with perception-stage tasks (TB-100k) and planning tasks on the BDD-X dataset [19] to evaluate improvements in downstream tasks.

Following [12], we use Mini-InternVL2 due to its similarity in frame index referencing, inherited from its pre-training process and the BDD-X annotation. We follow a standard MLLM training regime: stage 1 focuses on feature alignment using the pre-trained Mini-InternVL2 checkpoint, while stage 2 involves instruction tuning on the mixed datasets. We apply LoRA [16] to co-train on the mixed datasets configured as in Table 6, using the same hyperparameters as in [12].

The results on Table 6 show consistent performance improvements across most metrics when co-training the MLLM with our TB-100k dataset. It indicates the benefit of our proposed dataset on improving the downstream tasks.

7. Conclusion

We have introduced TB-Bench, a comprehensive benchmark that rigorously assesses MLLM performance across eight perception tasks, providing a much-needed standard for spatiotemporal evaluation in AD. Alongside TB-Bench, we have developed the vision-language instruction tuning datasets, TB-100k and TB-250k, which significantly improve MLLM performance when used to fine-tune our baseline models, resulting in substantial gains over existing models. Additionally, our VLIT datasets offer benefits as valuable assets for mixed training datasets in other driving use cases. Our contributions represent incremental progress while laying a foundation for integrating MLLMs into AD perception, prediction, and planning stages, supporting the development of more capable and reliable autonomous driving systems. Please refer to the supp. material for further discussion on broader impact, limitations, and future work.

Acknowledgments This work was partly supported by JST [Moon-shot Research and Development], Grant Number [JPMJMS2032] and JSPS KAKENHI Grant Number 20H05952 and 23H00482.

References

- [1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024. 6
- [2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1
- [3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 7
- [4] 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 multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 3
- [5] Xu Cao, Tong Zhou, Yunsheng Ma, Wenqian Ye, Can Cui, Kun Tang, Zhipeng Cao, Kaizhao Liang, Ziran Wang, James M Rehg, et al. Maplm: A real-world large-scale vision-language benchmark for map and traffic scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21819–21830, 2024. 3
- [6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024. 3, 7
- [7] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatial-rgpt: Grounded spatial reasoning in vision language model. *arXiv preprint arXiv:2406.01584*, 2024. 3, 4
- [8] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3
- [9] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. Talk2Car: Taking control of your self-driving car. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 2088–2098, 2019. 3
- [10] Alexey Dosovitskiy, German Ros, Felipe Codella, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 4
- [11] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 3, 4
- [12] Zhangwei Gao, Zhe Chen, Erfei Cui, Yiming Ren, Weiyun Wang, Jinguo Zhu, Hao Tian, Shenglong Ye, Junjun He, Xizhou Zhu, et al. Mini-internvl: A flexible-transfer pocket multimodal model with 5% parameters and 90% performance. *arXiv preprint arXiv:2410.16261*, 2024. 7, 8
- [13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 5
- [14] 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. 3
- [15] Muyang He, Yixin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multi-modal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*, 2024. 3, 7
- [16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 7, 8
- [17] Yuichi Inoue, Yuki Yada, Kotaro Tanahashi, and Yu Yamaguchi. Nusenes-mqa: Integrated evaluation of captions and qa for autonomous driving datasets using markup annotations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 930–938, 2024. 3
- [18] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhui Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024. 7
- [19] 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, 3, 8
- [20] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 3, 4
- [21] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 1, 4
- [22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 3
- [23] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024. 3, 4

- [24] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024. 3
- [25] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 4
- [26] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 3, 7
- [27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7
- [28] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16488–16498, 2024. 3, 4
- [29] 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. 3
- [30] J Mao, M Niu, C Jiang, H Liang, J Chen, X Liang, Y Li, C Ye, W Zhang, Z Li, et al. One million scenes for autonomous driving: Once dataset. *arxiv* 2021. *arXiv preprint arXiv:2106.11037*. 3, 5
- [31] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Video question answering for autonomous driving. *arXiv preprint arXiv:2312.14115*, 2023. 3
- [32] Wassim G Najm, John D Smith, Mikio Yanagisawa, et al. Pre-crash scenario typology for crash avoidance research. Technical report, United States. Department of Transportation. National Highway Traffic Safety ..., 2007. 4
- [33] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. *arXiv preprint arXiv:2312.03661*, 2023. 3
- [34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 6
- [35] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4542–4550, 2024. 3, 2
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 8
- [37] Katrin Renz, Long Chen, Ana-Maria Marcu, Jan Hünermann, Benoit Hanotte, Alice Karnsund, Jamie Shotton, Elahe Arani, and Oleg Sinavski. Carllava: Vision language models for camera-only closed-loop driving. *arXiv preprint arXiv:2406.10165*, 2024. 1
- [38] Enna Sachdeva, Nakul Agarwal, Suhas Chundi, Sean Roelofs, Jiachen Li, Mykel Kochenderfer, Chiho Choi, and Behzad Dariush. Rank2tell: A multimodal driving dataset for joint importance ranking and reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7513–7522, 2024. 3
- [39] Zhenwei Shao, Zhou Yu, Jun Yu, Xuecheng Ouyang, Lihao Zheng, Zhenbiao Gai, Mingyang Wang, and Jiajun Ding. Imp: Highly capable large multimodal models for mobile devices. *arXiv preprint arXiv:2405.12107*, 2024. 3
- [40] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *Proceedings of the European Conference on Computer Vision*, 2024. 3, 4, 7
- [41] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 3
- [42] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1
- [43] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024. 2, 3
- [44] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 2, 3, 4
- [45] Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Bangjun Wang, Peijin Jia, Yuting Wang, Shengyin Jiang, et al. Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [46] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemobel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 3, 5

- [47] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chau-
mond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim
Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface’s
transformers: State-of-the-art natural language processing.
arXiv preprint arXiv:1910.03771, 2019. 6
- [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. 3
- [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. 3
- [50] Penghao Wu and Saining Xie. V?: Guided visual search as
a core mechanism in multimodal llms. In *Proceedings of
the IEEE/CVF Conference on Computer Vision and Pattern
Recognition*, pages 13084–13094, 2024. 2
- [51] Li Xu, He Huang, and Jun Liu. Sutd-trafficqa: A question
answering benchmark and an efficient network for video rea-
soning over traffic events. In *Proceedings of the IEEE/CVF
conference on computer vision and pattern recognition*,
pages 9878–9888, 2021. 3
- [52] 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 Vi-
sion and Pattern Recognition*, pages 9523–9532, 2020. 3
- [53] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen
Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng
Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint
arXiv:2407.10671*, 2024. 7
- [54] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng,
Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon
Ye, Joel Jang, et al. Magma: A foundation model for mul-
timodal ai agents. In *Proceedings of the IEEE/CVF Confer-
ence on Computer Vision and Pattern Recognition*, 2025. 7
- [55] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruqi
Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming
Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline
multimodal understanding and reasoning benchmark for ex-
pert agi. In *Proceedings of the IEEE/CVF Conference
on Computer Vision and Pattern Recognition*, pages 9556–
9567, 2024. 3, 4
- [56] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and
Lucas Beyer. Sigmoid loss for language image pre-training.
In *Proceedings of the IEEE/CVF International Conference
on Computer Vision*, pages 11975–11986, 2023. 7
- [57] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng,
Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan,
Chunyuan Li, and Ziwei Liu. Long context transfer from
language to vision. *arXiv preprint arXiv:2406.16852*, 2024.
1, 4
- [58] Weize Zhang, Mohammed Elmahiubi, Kasra Rezaee,
Behzad Khamidehi, Hamidreza Mirkhani, Fazel Arasteh,
Chunlin Li, Muhammad Ahsan Kaleem, Eduardo R Corral-
Soto, Dhruv Sharma, et al. Analysis of a modular au-
tonomous driving architecture: The top submission to carla
leaderboard 2.0 challenge. *arXiv preprint arXiv:2405.01394*,
2024. 1
- [59] YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu,
Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qing-
song Wen, Zhang Zhang, Liang Wang, and Rong Jin. MME-
realworld: Could your multimodal LLM challenge high-
resolution real-world scenarios that are difficult for humans?
In *International Conference on Learning Representations*,
2025. 3, 4
- [60] Yunsong Zhou, Linyan Huang, Qingwen Bu, Jia Zeng,
Tianyu Li, Hang Qiu, Hongzi Zhu, Minyi Guo, Yu Qiao, and
Hongyang Li. Embodied understanding of driving scenarios.
arXiv preprint arXiv:2403.04593, 2024. 3

TB-Bench: Training and Testing Multi-Modal AI for Understanding Spatio-Temporal Traffic Behaviors from Dashcam Images/Videos

Supplementary Material

This material includes the following sections:

- **Discussions:** The broader impact, limitations, and future directions of our work.
- **Access Information:** A URL for accessing the benchmark, datasets, and future update.
- **Task Definitions and Dataset Statistics:** A detailed overview of the task definitions and relevant dataset statistics.
- **Data Generation Pipeline:** Insights into the Data Generation Pipeline used in our study.
- **Evaluation Details:** Information on metrics, models, and evaluation methods.
- **Experiments and Results:** Implementation details, quantitative analyses, qualitative results, and ablation studies.

8. Discussions

8.1. Broader Impact

This study represents progress in enhancing the capabilities of Multi-Modal Large Language Models (MLLMs) by focusing on a limited set of AD perception tasks. Specifically, we introduce a new benchmark to evaluate MLLMs on understanding diverse traffic behaviors and provide high-quality VLIT datasets that enhance MLLMs' generalizability. We hope this will advance MLLMs' applications in AD, contributing to the development of more robust autonomous driving systems.

8.2. Limitations

Firstly, our study utilizes the moderate large language models (Qwen 0.5B series) due to limited computational resources, which can be scaled up as needed.

Secondly, we acknowledge the dataset imbalance arising from the natural occurrence of specific autonomous driving behaviors; please refer to Section Dataset Statistics for more details.

Lastly, the free-form text output templates in TB-100k and TB-250k are limited for certain tasks. However, we believe that the diversity of images is also important for the model to understand visual concepts. That being said, when combined with other (vision-)language instruction tuning datasets, our datasets still enhance the performance of MLLMs, enabling them to generalize better in traffic domains, particularly in understanding traffic behaviors.

8.3. Future Work

Future research could expand this work by incorporating a wider range of perception tasks or by exploring subsequent stages, such as prediction and planning.

Additionally, an important direction for future investigation is the optimal application of upstream perception tuning sets, including the TB-100k and TB-250k datasets, to relevant downstream traffic tasks. This approach may enhance model performance in real-world applications.

Furthermore, integrating real-time traffic data, such as video feeds and sensory inputs, could improve the MLLMs' understanding of dynamic traffic situations. Finally, enhancing the explainability of MLLMs in traffic behavior scenarios will help users understand the rationale behind model predictions.

9. Access to the Benchmark and Datasets

9.1. Availability

The Traffic Behavior Benchmark (TB-Bench) and the training datasets (TB-100k, TB-250k) will be publicly available at the following Github repository:

- <https://github.com/TB-AD/TB-Bench-110k-250k>

The source code for conducting and analyzing the experiments will also be publicly available in the repository upon publication, permitting free use for research purposes.

9.2. Future Update

We also plan to establish an evaluation server and leaderboard on HuggingFace in the future. Any updates will be communicated through the above Github repository to ensure users have access to the latest information.

10. Benchmark and Datasets

10.1. Task Definition

10.1.1. Relative Distance (RD).

The task is to predict the Euclidean distance in meters between two entities in an image; see Figure 14 for two examples.

10.1.2. Spatial Reasoning (SR).

The task is to predict the spatial position of one entity relative to another from the perspective of a reference entity; see Figure 15 for examples. Specifically, the relationship between two objects is defined by the angle θ , as follows:

$$\text{Relation} = \begin{cases} \text{front} & \text{if } -30^\circ < \theta \leq 30^\circ, \\ \text{front left} & \text{if } 30^\circ < \theta \leq 90^\circ, \\ \text{front right} & \text{if } -90^\circ < \theta \leq -30^\circ, \\ \text{back left} & \text{if } 90^\circ < \theta \leq 150^\circ, \\ \text{back right} & \text{if } -150^\circ < \theta \leq -90^\circ, \\ \text{back} & \text{otherwise.} \end{cases} \quad (1)$$

This angular relationship is similar to that defined in [35].

10.1.3. Orientation Reasoning (OR).

This task is to predict the facing relationship between two entities from the perspective of a reference entity, categorized as: ‘similar’, ‘opposite’, or ‘perpendicular’. Please refer to Figure 16 for examples. The relationship is defined based on the absolute difference in facing angles $|\theta|$, as follows:

$$\text{Relation} = \begin{cases} \text{similar} & \text{if } 0^\circ \leq |\theta| \leq 45^\circ, \\ \text{opposite} & \text{if } 135^\circ \leq |\theta| \leq 180^\circ, \\ \text{perpendicular} & \text{otherwise.} \end{cases} \quad (2)$$

It is noted that this angle is measured from the facing direction of a reference entity to the position of the target entity in Euclidean space, irrespective of the target entity’s facing direction.

10.1.4. Other Lane to Ego-Vehicle (EGO-LANE).

This task is to predict the lane of a target vehicle relative to the ego-vehicle’s perspective; see Figure 17 for examples. The categories include: ‘front lane’, ‘front left lane’, ‘front right lane’, and ‘oncoming traffic lane’ (the lane on the opposite side of the road).

It is noted that when the ego-vehicle is on a road with multiple lanes, the ‘front lane’ is further classified into three fine-grained categories: ‘front lane’, ‘front left lane’, and ‘front right lane’.

10.1.5. Other Lane Changing (OBJ-LANE).

This task is to predict whether the target vehicle is changing lanes, categorized as ‘left lane change’, ‘right lane change’, or ‘no change’; see Figure 18 for examples. Lane changes are evaluated based on the target vehicle’s viewpoint. For instance, if the target vehicle in the oncoming traffic lane executes a right lane change, the ego vehicle perceives it as moving to the left.

10.1.6. Other Turning (OBJ-TURN).

This task is to predict whether the target vehicle is making a turn, categorized as ‘turning left’, ‘turning right’, or ‘go straight’. The target vehicle is considered to be turning, if it changes direction by more than 25 degrees within a period of 1.6 seconds. Please refer to Figure 19 for examples.

10.1.7. Ego Turning (EGO-TURN).

This task is to predict whether the ego-vehicle is making a turn, categorized as turning left, turning right, or going straight. The turning maneuver of the ego-vehicle is also defined by a change in direction of more than 25 degrees within a period of 1.6 seconds. Please refer to Figure 20 for examples.

10.1.8. Ego Traverse Distance (EGO-TRA).

This task is to predict the traverse distance of the ego vehicle in meters over a period of 1.6 seconds. Please see Figure 21 for examples.

10.2. Dataset Statistics

Table 7, 8, and 9 show the distribution of categories for the TB-Bench, TB-100k, and TB-250k datasets, respectively, detailing the count and percentage of samples for various task types.

To create the TB-Bench, we manually screened the frames thoroughly to select samples with clearly visible target entities. Each task in TB-Bench has an equal count of 250 samples. We ensure that the distribution of categories in each task closely resembles that of the instruction tuning datasets.

It is seen from Table 8, and 9 that TB-250k represents a normal scene occurrence distribution in real-world scenarios, while TB-100k is a more label-balanced version.

11. Data Generation Pipeline Details

11.1. Information Extraction

Figure 4 shows the extraction process. It begins with obtaining raw sensory data from input samples, which may include static images with entity attributes from datasets like KITTI or ONCE, or sequential data from Argoverse2. This sensory data is processed to filter out insignificant scene information.

For Argoverse2, lane geometry information is processed concurrently. Lane coordinates are used to create polygons with attributes, such as neighboring, successor, and predecessor lanes. This information helps determine lane direction and angle, which are then projected onto vehicle attributes to obtain the vehicle’s lane ID and relevant lane information. This data is subsequently passed to the next processing step to extract all scene attributes.

11.2. Rule-based Q&A Generation

The process begins with obtaining attribute data from either the nodes or edges of the relationship graph. This data is then processed through rule-based functions to extract behavioral or spatial information. Next, we generate behavioral attributes in a Q&A format using templates provided in Table 10.

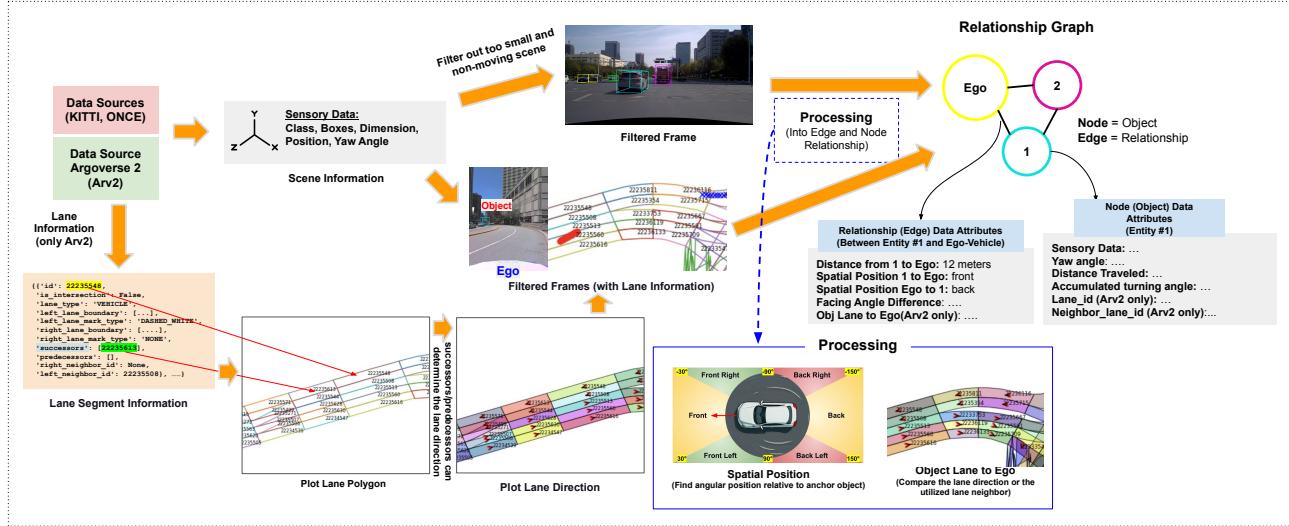


Figure 4. Data Extraction Process.

Table 7. TB-Bench Statistics

Task Type	Category	Count	Percentage (%)
Relative Distance	numerical value	250	12.5
Spatial Reasoning	back	61	3.0
	back left	30	1.5
	back right	9	0.4
	front	87	4.3
	front left	45	2.2
	front right	18	0.9
Orientation Reasoning	numerical value	122	6.1
	opposite	51	2.5
	perpendicular	16	0.8
	similar	61	3.0
Other Lane to Ego-Vehicle	front lane	71	3.5
	front left lane	40	2.0
	front right lane	31	1.6
	oncoming traffic lane	108	5.4
Other Lane Changing	left lane change	62	3.1
	no change	142	7.1
	right lane change	46	2.3
Other Turning	go straight	126	6.3
	left turn	67	3.4
	right turn	57	2.9
Ego Turning	go straight	122	6.1
	left turn	38	1.9
	right turn	90	4.5
Ego Traverse Distance	numerical value	250	12.5

Table 8. TB-100k Statistics

Task Type	Category	Count	Percentage (%)
Relative Distance	numerical value	10000	9.1
Spatial Reasoning	back	3580	3.3
	back left	3183	2.9
	back right	3115	2.8
	front	7873	7.2
	front left	7321	6.7
	front right	4928	4.5
Orientation Reasoning	numerical value	10000	9.1
	opposite	10013	9.1
	perpendicular	2387	2.2
	similar	7600	6.9
Other Lane to Ego-Vehicle	front lane	3889	3.5
	front left lane	3231	2.9
	front right lane	4182	3.8
	oncoming traffic lane	8698	7.9
Other Lane Changing	left lane change	414	0.4
	no change	807	0.7
	right lane change	279	0.3
Other Turning	go straight	744	0.7
	left turn	435	0.4
	right turn	321	0.3
Ego Turning	go straight	753	0.7
	left turn	331	0.3
	right turn	416	0.4
Ego Traverse Distance	numerical value	15500	14.1

Generation depends on the task type. Tasks 1-4 and task 8 ('Relative Distance,' 'Spatial Reasoning,' 'Orientation Reasoning,' 'Other Lane to Ego,' and 'Ego Traverse Distance') can be created in any frame, as their attributes

are available in all frames.

In contrast, tasks 5-7 ('Other Lane Changing,' 'Other Turning,' and 'Ego Turning') require a triggering event, specifically a change in attributes. The following details

Table 9. TB-250k Statistics

Task Type	Category	Count	Percentage (%)
Relative Distance	numerical value	34721	13.7
Spatial Reasoning	back	17023	6.7
	back left	6247	2.5
	back right	3966	1.6
	front	26917	10.6
	front left	10793	4.3
	front right	4804	1.9
Orientation Reasoning	numerical value	34872	13.7
	opposite	19242	7.6
	perpendicular	3355	1.3
	similar	12283	4.8
Other Lane to Ego-Vehicle	front lane	14312	5.6
	front left lane	4454	1.8
	front right lane	6401	2.5
	oncoming traffic lane	24833	9.8
Other Lane Changing	left lane change	414	0.2
	no change	807	0.3
	right lane change	279	0.1
Other Turning	go straight	744	0.3
	left turn	435	0.2
	right turn	321	0.1
Ego Turning	go straight	753	0.3
	left turn	331	0.1
	right turn	416	0.2
Ego Traverse Distance	numerical value	25000	9.9

explain how to trigger an event:

Event Triggering: Other Lane Changing

- Check if the current `lane_id` is in the `future_right_neighbor_id`.
If yes, then assign: **Right Lane Change**.
- Check if the current `lane_id` is in the `future_left_neighbor_id`.
If yes, then assign: **Left Lane Change**.
- If neither condition is met, assign: **No Change**.

Note: `future_right_neighbor_id` refers to the `right_neighbor_id` of the next time step; the same applies to the left side.

Event Triggering: Other Turning

- Check if the accumulated object yaw angle is greater than 25 degrees in 1.6 seconds.
If yes, then assign: **Turn Left**.
- Check if the accumulated object yaw angle is less than -25 degrees in 1.6 seconds.
If yes, then assign: **Turn Right**.
- If neither condition is met, assign: **Go straight**.

Event Triggering: Ego Turning

- Check if the accumulated ego-vehicle yaw angle is greater than 25 degrees in 1.6 seconds.
If yes, then assign: **Turn Left**.
- Check if the accumulated ego-vehicle yaw angle is less than -25 degrees in 1.6 seconds.
If yes, then assign: **Turn Right**.
- If neither condition is met, assign: **Go straight**.

11.3. Q&A Augmentation

The augmentation process converts short question-answer (Q&A) pairs into natural language sentences. Each short QA pair was expanded into a full sentence using a pre-defined structure. We employ the Microsoft-Phi3-medium model to generate these sentences, using the following prompt:

Complete Prompt

```
system_text = "You are a language expert
assistant. In this task, we want to expand
the following answer to longer wording but
no additional information."
full_prompt = f"{system_text}. The question
is: {question} and the short answer is
{answer}. Give the complex answer in a
short sentence no more than 15 words."
```

The parameters for `{question}`, and `{answer}` are dynamically inserted for each instance. This approach ensures that the augmented data remains concise (up to 15 words) while incorporating the original short answer in a more elaborated context, maintaining the correctness and relevance of the response.

11.4. Pre-crash Scenarios

Figure 5 presents the full list of 65 pre-crash scenarios as described in Section Task Design, based on National Automotive Sampling System. Each scenario is categorized into a specific accident type, such as ‘Animal’, ‘Off-road’, etc.

12. Evaluation Details

12.1. Evaluation Metrics

As mentioned in the main paper, we employ the rule-based methods for evaluation. Figure 6 shows the keyword list and regular expression used in the evaluation pipeline.

12.2. Additional Details on Evaluated Models

In this study, we evaluate open-source state-of-the-art models and proprietary models on our TB-Bench in a zero-shot manner. We provide additional information for the evaluated models in Table 11.

Table 10. **Q&A Templates.** The placeholder <entity_n> refers to any entity, such as ‘Entity #1’, ‘Entity #2’, or ‘Ego-vehicle’, ensuring that no sentence contains duplicate entities. ‘Short Answer Template’ denotes a basic class of concise responses that can be expanded into more complex sentences.

Task Type	Question Template	Short Answer Template
Relative Distance	Can you measure straight-line distance in meters between <entity_n> and <entity_n>? How far is <entity_n> from <entity_n> in meters? How many meters apart are <entity_n> and <entity_n>? What is distance from <entity_n> to <entity_n> along road’s surface in meters?	xx.xx meters
Spatial Reasoning	How are <entity_n> and <entity_n> spatially related, from <entity_n> perspective? What is spatial position of <entity_n> relative to <entity_n>? What is spatial relation of <entity_n> to <entity_n>?	back, back left, back right, front, front left, front right
Orientation Reasoning	How do you describe orientation of <entity_n> relative to <entity_n>, similar, opposite or perpendicular? How is <entity_n> oriented relative to <entity_n>, similar, opposite or perpendicular? What is angle between <entity_n> and <entity_n>, in degrees? What is facing angle of <entity_n> relative to <entity_n>, in degrees? What is orientation of <entity_n> relative to <entity_n>, similar, opposite or perpendicular? What is yaw angle different between <entity_n> and <entity_n>, in degrees?	opposite, perpendicular, similar, xx.xx degrees
Other Lane to Ego-Vehicle	How would you describe lane position of Entity#1? Options: front lane, front left lane, front right lane, or oncoming traffic lane.	front_lane, front_left_lane, front_right_lane, oncoming_traffic_lane
Other Lane Changing	How would you describe driving scene involving Entity#1? Please explain, focusing on vehicle’s lane change maneuver.	left_lane_change, no_change, right_lane_change
Other Turning	How would you describe driving scene involving Entity#1? Please explain, focusing on vehicle’s turning maneuver.	go_straight, left_turn, right_turn
Ego Turning	How would you describe driving scene involving our car? Please explain, focusing on our car’s turning maneuver.	go_straight, left_turn, right_turn
Ego Traverse Distance	How far has our car driven and what kind of steering maneuver did it perform in current scene?	xx.xx meters

The first category consists of open-source models (LLaVA, Bunny, and InternVL), which are accessible via the Hugging Face API. These models are fully fine-tuned with specific settings for each version available in their Huggingface repositories.

The second category consists of proprietary models (GPT-4o and Gemini), which require specific API calls and image formatting. It is noted that we evaluate the latest version of these models on our TB-Bench at the time of submission.

12.3. Prompt for Zero-Shot Evaluation

For zero-shot evaluation of existing models, we use an Option Template that presents multiple-choice options to define possible answer classes. This approach accommodates the varied terminology that pre-trained models may employ

to describe situations.

The details of the Option Template, which varies based on the task type, are as follows:

Option Template
Distance-Related Tasks:
• Answer in xx.x meters format.
Angle-Related Tasks:
• Answer in xx.x degrees format.
Tasks with Predefined Answer Choices:
• Retrieve the answer choices.
• Assign a letter to each choice (e.g., A, B, C).
• Present options as follows:
Options:
A. choice1,
B. choice2,
C. choice3, ...

Table 11. Additional information of the models evaluated on TB-Bench.

Model Name	Full Repository/API Name	Vision Part	Language Part
Open-source models			
LLaVA-1.5-7B	llava-hf/llava-1.5-7b-hf	CLIP-L/14	Vicuna-7b-v1.5
LLaVA-v1.6-Mistral-7B	llava-hf/llava-v1.6-mistral-7b-hf	CLIP-L/14	Mistral-7B-Instruct-v0.2
LLaVA-NeXT-Video-7B	llava-hf/LLaVA-NeXT-Video-7B-hf	CLIP-L/14	Vicuna-7B-v1.5
LLaVA-Interleave-Qwen-7B	llava-hf/llava-interleave-qwen-7b-hf	SigLIP-L/14	Qwen1.5-7B-Chat
Bunny-v1.1-4B	BAAI/Bunny-v1_1-4B	SigLIP-L/14	Phi-3-mini-4k-instruct
Bunny-v1.1-Llama-3-8B-V	BAAI/Bunny-v1_1-Llama-3-8B-V	SigLIP-L/14	Llama-3-8B-Instruct
InternVL2-8B	OpenGVLab/InternVL2-8B	InternViT-300M-448px	Qwen2-8B-Instruct
Magma-8B	microsoft/Magma-8B	ConvNext-XXlarge	LLaMA-3-8B
Mini-InternVL2-1B-DriveLM	OpenGVLab/Mini-InternVL2-1B-DA-DriveLM	InternViT-300M-448px	Qwen2-0.5B
DriveLM-mantis-8b	francepl/DriveLM-mantis-8b-idefics2_8192	SigLIP	Mistral-7B-v0.1
Proprietary models			
Gemini-1.5-flash	Gemini-1.5-flash	Unknown	Unknown
GPT-4o-2024-08-06	GPT-4o-2024-08-06	Unknown	Unknown

Pre-trained models often use specific vocabularies based on their training data. For instance, a model might say ‘opposite side of the road’ instead of ‘oncoming traffic lane’ if it lacks specific instruction training. By offering explicit choices, the model can select the appropriate terminology despite variations.

For numerical answers, we specify the expected format within the prompt to ensure clarity and consistency, such as instructing the model to Answer in xx.x meters format.

This structured approach allows the model to account for variations in wording and select the most appropriate option, demonstrating its understanding.

13. Experiments and Results

13.1. Implementation Details

Table 12. Hyper-parameter settings for finetuning our models on TB-100k or TB-250k.

Hyper-parameter	Value
Epochs	10
Warmup steps	2,000
Learning rate	1e-5
LoRA learning rate	1e-4
Effective Batch size	64
AdamW β	(0.9, 0.999)
Weight decay	0.05
Drop path	0
Attention dropout	0
Torch data type	bf16
Inference temperature	0

All models are finetuned on an Ubuntu 20.04 server equipped with four A6000 GPUs, each with 48GB of memory. The source code is built on the Transformers library

[47] and utilizes the PyTorch 2.4 framework [34].

Additional information on hyper-parameter settings for finetuning our baseline models on TB-100k and TB-250k is presented in Table 12.

13.2. Quantitative Analyses

We provide quantitative analyses and the qualitative results of the model’s predictions on TB-Bench. The baseline model ((SigLIP-L/14 and Qwen1.5-0.5b) finetuned on TB-100k. For numerical output tasks, we visualize error distributions using box plots. On the other hand, we use confusion matrices for classification tasks.

13.2.1. Relative Distance and Ego Traverse Distance Tasks.

Figure 7 shows the box plot for distance errors of our model predictions on the two tasks. For RD, distance errors are generally centered around zero, with a narrow interquartile range, indicating consistent performance, though a few outliers suggest overestimation. Predictions on EGO-TRA show a similar error distribution, with the median slightly above zero and more positive outliers, indicating a tendency to overestimate distance.

13.2.2. Orientation Reasoning Task.

Figure 8 shows the box plot for angular errors of our model predictions on the Orientation Reasoning (OR) task. The median and interquartile range are close to zero, indicating precise and consistent predictions. Short whiskers further highlight this accuracy. Outliers are grouped near 0, 90, and 180 degrees, suggesting small angle misestimations. Overall, the model demonstrates minimal errors in this task.

13.2.3. Spatial Reasoning Task.

Figure 9 shows the confusion matrix of our model predictions on the Spatial Reasoning (SR) task. The ‘front’ position is classified most accurately at 85.1%, while ‘back’ and ‘back left’ positions have lower accuracies of 63.3% and

No.	Scenario Definition
1	Animal: other
2	Animal: vehicle going straight and animal in road
3	Animal: vehicle negotiating a curve and animal in road
4	Off-road: single vehicle performing avoidance maneuver
5	Off-road: single vehicle going straight and departing road edge
6	Off-road: single vehicle going straight and losing control
7	Off-road: single vehicle initiating a maneuver and departing road edge
8	Off-road: single vehicle initiating a maneuver and losing control
9	Off-road: single vehicle negotiating a curve and departing road edge
10	Off-road: single vehicle negotiating a curve and losing control
11	Off-road: single vehicle and other loss of control
12	Off-road: single vehicle due to vehicle failure
13	Off-road: single vehicle and other road edge departure
14	Off-road: single vehicle with other/unknown
15	Off-road: backing
16	Off-road: no impact
17	Pedalcyclist: other/unknown
18	Pedalcyclist: vehicle going straight on crossing paths
19	Pedalcyclist: vehicle going straight on parallel paths
20	Pedalcyclist: vehicle starting in traffic lane on crossing paths
21	Pedalcyclist: vehicle turning left on crossing paths
22	Pedalcyclist: vehicle turning left on parallel paths
23	Pedalcyclist: vehicle turning right on crossing paths
24	Pedalcyclist: vehicle turning right on parallel paths
25	Pedestrian: other
26	Pedestrian: vehicle backing
27	Pedestrian: vehicle going straight and pedestrian crossing road
28	Pedestrian: vehicle going straight and pedestrian darting onto road
No.	Scenario Definition
29	Pedestrian: vehicle going straight and pedestrian playing/working on Road
30	Pedestrian: vehicle going straight and pedestrian walking along road
31	Pedestrian: vehicle turning left and pedestrian crossing road
32	Pedestrian: vehicle turning right and pedestrian crossing road
33	Backing: at driveways
34	Backing: at intersections
35	Backing: other
36	Lane change: 2 vehicles going straight and 1 vehicle encroaching in same lane
37	Lane change: 2 vehicles going straight and 1 vehicle encroaching into another lane
38	Lane change: 1 vehicle going straight and another changing lanes
39	Lane change: 1 vehicle going straight and another entering or leaving parking position
40	Lane change: 1 vehicle going straight and another passing
41	Lane change: 1 vehicle going straight and another turning
42	Lane change: 2 vehicles in other combinations
43	Lane change: 1 vehicle passing and another turning
44	Opposite direction: control loss
45	Opposite direction: 2 vehicles going straight and 1 vehicle encroaching
46	Opposite direction: 2 vehicles going straight both in same lane
47	Opposite direction: 2 vehicles negotiating a curve and 1 vehicle encroaching
48	Opposite direction: 2 vehicles negotiating a curve both in same lane
49	Opposite direction: other/unknown
50	Opposite direction: involves 1 vehicle passing
51	Opposite direction: involves vehicle failure
52	Rear-end: following vehicle changing lanes
53	Rear-end: lead vehicle accelerating
54	Rear-end: lead vehicle changing lanes
55	Rear-end: lead vehicle decelerating
56	Rear-end: lead vehicle moving at constant, slower speed
57	Rear-end: lead vehicle stopped
58	Rear-end: other/unknown
59	Crossing paths: left turn across path from lateral direction (LTAP/LD)
60	Crossing paths: left turn across path from opposite direction (LTAP/OD)
61	Crossing paths: left turn into path (LTIP)
62	Crossing paths: other/unknown
63	Crossing paths: right turn across path from lateral direction (RTAP/LD)
64	Crossing paths: right turn into path (RTIP)
65	Crossing paths: straight crossing paths (SCP)

Figure 5. List of pre-crash scenarios based on National Automotive Sampling System (NASS) variables.

66.7%. The matrix also shows moderate confusion between similar positions, such as ‘back left’ being misclassified as ‘front right’ (23.33%) and ‘back’ as ‘front’ (19.67%).

13.2.4. Other Lane to Ego-Vehicle Task.

Figure 10 shows the confusion matrix of our model predictions on the Other Lane to Ego-Vehicle (EGO-LANE) task. Overall, the model shows high accuracy on most categories (over 96%), except for the ‘front lane,’ which has an accuracy of only 81.7%. The primary misclassification pattern involves confusion between the ‘front lane’ and its adjacent lanes, with 9.9% of ‘front lane’ samples being misclassified as ‘front right lane.’

13.2.5. Other Lane Changing Task.

Figure 11 shows the confusion matrix on the Other Lane Changing (OBJ-LANE) task, where samples are categorized into ‘no change,’ ‘left lane change,’ and ‘right lane change.’ In this case, the model shows decent performance with an accuracy of around 78.87% in the ‘no change’ category. However, it struggles significantly with lane change predictions. For both ‘left lane change’ and ‘right lane change’ classifications, the most misclassified predictions are in the ‘no change’ category, with 32.3% and 30.4% misclassified, respectively. This indicates the model’s difficulty in distinguishing between lane changes and no change, underscoring the task’s challenges.

13.2.6. Other Lane Changing Task.

Figure 12 shows the confusion matrix on the Other Turning (OBJ-TURN) task, where samples are categorized as ‘left turn,’ ‘go straight,’ and ‘right turn.’ The model excels in identifying the ‘go straight’ category, achieving an accuracy of 80.16%. However, it shows over 30% misclassification rates for both ‘left turn’ and ‘right turn.’ Notably, misclassifications of ‘left turn’ are nearly evenly divided between ‘right turn’ and ‘go straight,’ despite ‘right turn’ errors being more theoretically opposed. The model’s performance indicates that it struggles to accurately interpret turns from the perspective of other vehicles, influenced by road orientation and vehicle positioning.

13.2.7. Ego Turning Task.

Figure 13 shows the confusion matrix on the task, where the actions are categorized as ‘left turn,’ ‘go straight,’ and ‘right turn.’ The model demonstrates strong performance in identifying turns, with high accuracy rates of 86.8% for ‘left turn’ and 86.67% for ‘right turn.’ Interestingly, the turning maneuvers have stronger performance than the ‘go straight’ action, with a notable 20.49% of ‘go straight’ samples being misclassified as ‘right turn.’

13.3. Qualitative Results

For brevity, we present two samples per task, each with input frame(s), the task question, and the ground truth answer. Each sample also includes predictions from our fine-tuned baseline model (SigLIP-L/14 and Qwen1.5-0.5b) and the

<p>Helper Function: Regex pattern to find distance/angle mentioned in text in the form of numbers followed by the word meter/meters/degree/degrees</p> <p>Relative Distance Evaluation: This function assesses the accuracy of predicted distances by comparing them to the ground truth.</p> <p>Distance Extraction: It extracts numerical distances from the predicted and ground truth texts using a helper function, returning 0 if extraction fails.</p> <p>Evaluation Logic: The function checks if the predicted distance falls within 25% of the ground truth. If it does, it returns a score of 1 for a correct prediction; otherwise, it returns 0.</p>	<p>Spatial Reasoning Evaluation: This function uses keyword lists for different spatial positions: front, front right, front left, back, back right, and back left. It checks the predicted text for these keywords.</p> <p>Keyword Lists: <code>front_right_keywords: ['front right', ...]</code> <code>front_left_keywords: ['front left', ...]</code> <code>front_keywords: ['positioned directly ahead of our car', ...]</code> <code>back_keywords: ['positioned directly behind', ...]</code> <code>back_right_keywords: ['back right', ...]</code> <code>back_left_keywords: ['back left', ...]</code> </p> <p>Checking Logic: The function verifies if the predicted text contains keywords from exactly one category. If so, it returns a score of 1 if it matches the ground truth, otherwise 0. If no or multiple matches are found, it returns 0 to prevent ambiguity.</p>
<p>Other Lane Changing Evaluation: This function uses keyword lists to identify lane change maneuvers: no change, left lane change, and right lane change. It checks the predicted text for these keywords.</p> <p>Keyword Lists: <code>no_change_list: ['maintains its lane', ...]</code> <code>left_lane_change_list: ['change to the left lane', ...]</code> <code>right_lane_change_list: ['change to the right lane', ...]</code> </p> <p>Checking Logic: The function verifies if the predicted text contains keywords from exactly one category. If so, it returns a score of 1 if it matches the ground truth, otherwise 0. If no or multiple matches are found, it returns 0 to prevent ambiguity.</p>	<p>Other Turning Evaluation: This function uses keyword lists to identify vehicle turning maneuvers: left turn, right turn, and go straight. It checks the predicted text for these keywords.</p> <p>Keyword Lists: <code>left_turn_list: ['turn left', ...]</code> <code>right_turn_list: ['turn right', ...]</code> <code>go_straight_list: ['go straight', ...]</code> </p> <p>Checking Logic: The function verifies if the predicted text contains keywords from exactly one category. If so, it returns a score of 1 if it matches the ground truth, otherwise 0. If no or multiple matches are found, it returns 0 to prevent ambiguity.</p>
<p>Orientation Reasoning Evaluation: This function uses keyword lists to identify vehicle orientations: perpendicular, opposite, and similar to the ego-vehicle. It checks the predicted text for these keywords.</p> <p>Keyword Lists: <code>perpendicular_list: ['perpendicular', ...]</code> <code>opposite_list: ['opposite', ...]</code> <code>similar_list: ['similar', ...]</code> </p> <p>Checking Logic: The function ensures the predicted text contains keywords from exactly one category to prevent ambiguity. If the ground truth is an angle, it calculates the angular difference between predicted and ground truth angles. If the difference is within 15 degrees, it returns a score of 1; otherwise, it returns 0. If no or multiple matches are found, it returns 0.</p>	<p>Other Lane to Ego Evaluation: This function uses keyword lists to identify lane positions: front lane, front left lane, front right lane, and oncoming traffic lane. It checks the predicted text for these keywords.</p> <p>Keyword Lists: <code>front_lane_list: ['front lane', ...]</code> <code>front_left_lane_list: ['front-left lane', ...]</code> <code>front_right_lane_list: ['front-right lane', ...]</code> <code>oncoming_traffic_lane_list: ['oncoming traffic lane', ...]</code> </p> <p>Checking Logic: The function verifies if the predicted text contains keywords from exactly one category. If so, it returns a score of 1 if it matches the ground truth, otherwise 0. If no or multiple matches are found, it returns 0 to prevent ambiguity.</p>
<p>Ego Turning Evaluation: This function uses keyword lists to identify turning maneuvers: right turn, left turn, and go straight. It checks the predicted text for these keywords.</p> <p>Keyword Lists: <code>right_turn_list: ['right turn', ...]</code> <code>left_turn_list: ['left turn', ...]</code> <code>go_straight_list: ['go straight', ...]</code> </p> <p>Checking Logic: The function verifies if the predicted text contains keywords from exactly one category. If so, it returns a score of 1 if it matches the ground truth, otherwise 0. If no or multiple matches are found, it returns 0 to prevent ambiguity.</p>	<p>Ego Traverse Distance Evaluation: This function assesses the accuracy of predicted distances traveled by the ego-vehicle.</p> <p>Distance Extraction: It retrieves distances using a helper function, returning 0 if extraction fails.</p> <p>Evaluation Logic: The function checks if the predicted distance is within 25% of the ground truth. If the ground truth distance is less than 1.0 meter, it checks if the predicted distance is within the adjusted range. It returns a score of 1 for a correct prediction and 0 otherwise.</p>

Figure 6. **Evaluation Metric Methodology for Each Task:** The method uses rule-based and regular expressions techniques to assess accuracy.

best performing zero-shot model, GPT-4o (GPT-4o-2024-08-06 version).

Figures for each task are as follows:

- Figure 14: Relative Distance (RD)

- Figure 15: Spatial Reasoning (SR)
- Figure 16: Orientation Reasoning (OR)
- Figure 17: Other Lane to Ego-Vehicle (EGO-LANE)
- Figure 18: Other Lane Changing (OBJ-LANE)

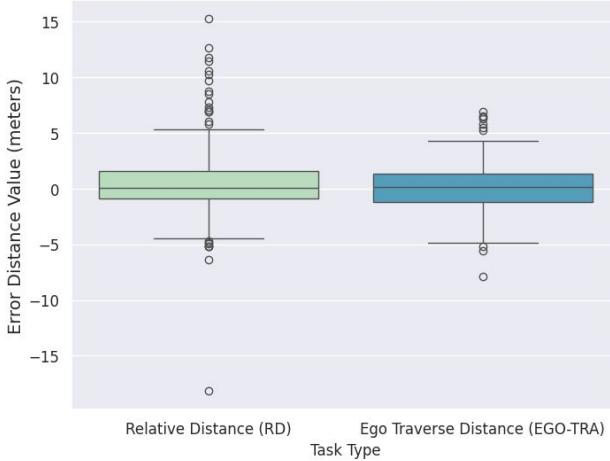


Figure 7. Distance error on Relative Distance (RD) and Ego Traverse Distance (EGO-TRA) tasks.

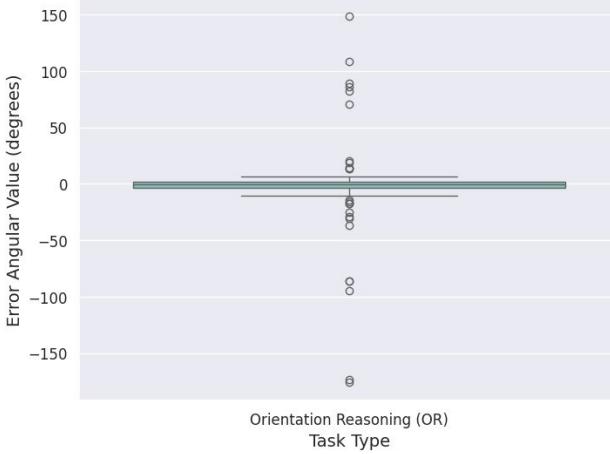


Figure 8. Angular error on Orientation Reasoning (OR) task.

- Figure 19: Other Turning (OBJ-TURN) task
- Figure 20: Ego Turning (EGO-TURN)
- Figure 21: Ego Traverse Distance (EGO-TRA)

13.4. Ablation Study Details

We provide detailed ablation results across eight tasks in Table 13.

Results indicate that stronger visual encoders significantly improve performance. For instance, comparing CLIP-L/14 to SigLIP-L/14 shows improvements of over 15.2% in Relative Distance (RD), 4.0% in Orientation Reasoning (OR), 5.6% in Other Turning (OBJ-TURN), and 10.4% in Ego Turning (EGO-TURN).

The optimal number of visual tokens is 16. Increasing this to 36 tokens improves Ego Traverse Distance (EGO-TRA) by only 2.8%, while performance in other tasks de-

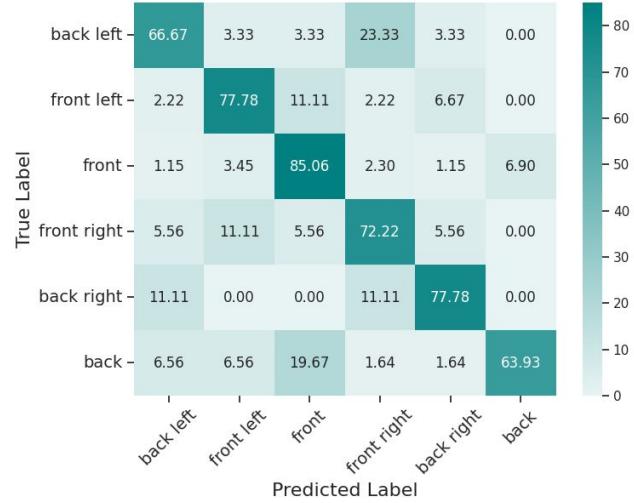


Figure 9. Confusion matrix on Spatial Reasoning (SR) task.

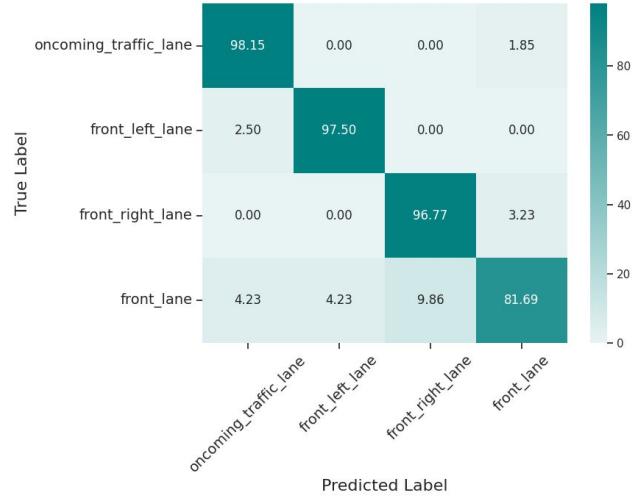


Figure 10. Confusion Matrix on Other Lane to Ego-Vehicle (EGO-LANE).

cines compared to the 16-token variant.

Utilizing more sequential frames generally enhances performance, especially in the tasks requiring temporal information (tasks 3-8). Single-frame tasks like Spatial Reasoning also benefit from training on multi-frame tasks, showing notable improvements. For ego-focused tasks, using 8 frames instead of 2 results in significant gains of over 14% in EGO-TURN and 12.8% in EGO-TRA, indicating that the number of frames is more critical for ego-focused tasks than for object-focused ones.

Table 13. **Ablation results per task.** All the models are finetuned on the TB-100k dataset, with their performance evaluated on TB-Bench and reported in accuracy (percentage).

Model	TrafficBehaviorBenchmark (TB-Bench)								
	RD ↑	SR ↑	OR ↑	EGO-LANE ↑	OBJ-LANE ↑	OBJ-TURN ↑	EGO-TURN ↑	EGO-TRA ↑	Avg ↑
Visual encoder									
CLIP-L/14	61.2	72.8	82.8	91.6	61.2	69.2	70.8	66.0	72.0
SigLIP-B/16	65.2	70.4	86.8	90.4	70.0	69.6	75.2	65.6	74.3
SigLIP-L/14	76.4	74.4	86.8	94.0	68.8	74.8	81.2	63.2	77.5
Visual tokens per frame									
4	68.8	70.0	86.4	94.0	67.6	74.0	71.6	49.2	72.7
16	76.4	74.4	86.8	94.0	68.8	74.8	81.2	63.2	77.5
36	75.5	70.8	84.4	91.2	64.8	71.2	77.6	66.0	76.2
Number of frames									
2	72.4	70.8	86.0	92.8	67.2	70.0	67.2	50.4	72.1
4	74.4	72.0	87.2	92.4	66.8	66.0	72.8	58.4	73.8
8	76.4	74.4	86.8	94.0	68.8	74.8	81.2	63.2	77.5

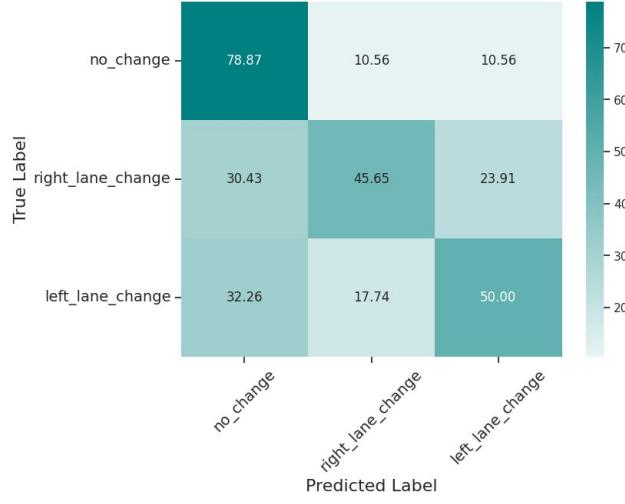


Figure 11. Confusion Matrix on Other Lane Changing (OBJ-LANE).

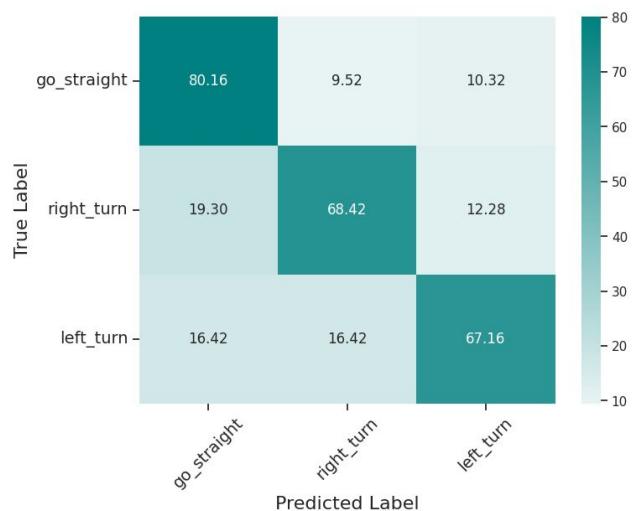


Figure 12. Confusion matrix on Other Turning (OBJ-TURN).

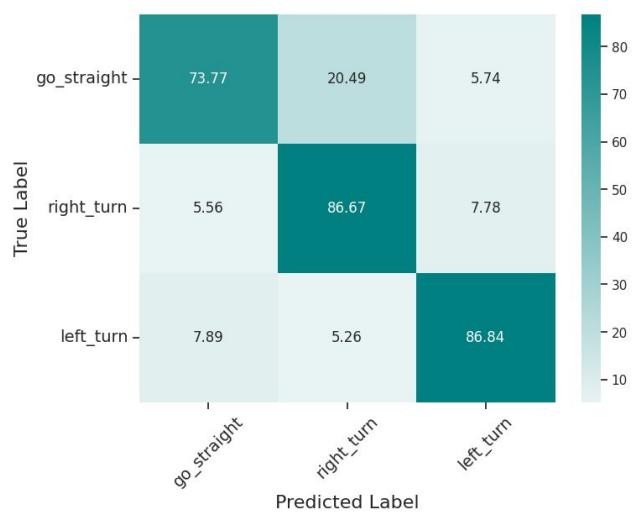


Figure 13. Confusion matrix on Ego Turning (EGO-TURN)

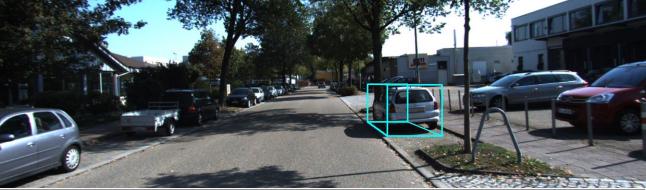
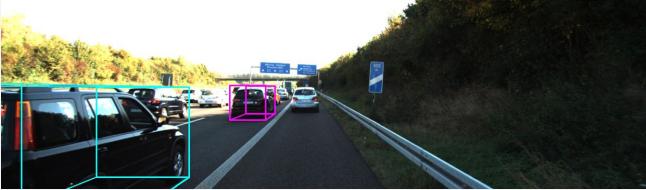
Question: How far is Entity #1 from the self-car in meters?	Question: Can you measure the straight-line distance in meters between Entity #1 and Entity #2?
	
Annotation: Entity #1 is positioned at a distance of 12.23 meters from the self-car.	Annotation: Yes, the straight-line distance between Entity #1 and Entity #2 is precisely 13.41 meters.
Ours: Entity #1 is situated at a distance of 13.72 meters from the self-car. ✓	Ours: Yes, the straight-line distance between Entity #1 and Entity #2 is precisely 14.80 meters. ✓
GPT-4o: The distance from Entity #1 to the self-car is approximately 23.5 meters. ✗	GPT-4o: I'm unable to measure distances in images. ✗

Figure 14. Examples and predictions from our baseline method and GPT-4o for the Relative Distance (RD) task.

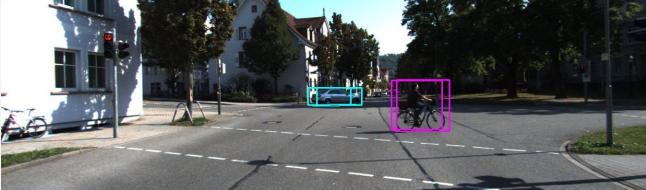
Question: How are Entity #1 and the self-car spatially related, from the self-car perspective?	Question: How are Entity #1 and Entity #2 spatially related, from the Entity #2 perspective?
	
Annotation: Entity #1 is positioned directly ahead of the self-car, establishing a forward spatial relationship from the self-car's perspective.	Annotation: From the perspective of Entity #2, Entity #1 is positioned at the back left relative to its own orientation.
Ours: Entity #1 is positioned directly ahead of the self-car, establishing a forward spatial relationship from the self-car's perspective. ✓	Ours: From the perspective of Entity #2, Entity #1 is positioned at the back right side. ✗
GPT-4o: from the self-car perspective, Entity #1 is located at the front. the correct option is a. Front. ✓	GPT-4o: From the perspective of Entity #2 (the bicycle), Entity #1 (the car) is spatially related as option b: front right. ✗

Figure 15. Examples and predictions from our baseline method and GPT-4o for the Spatial Reasoning (SR) task.

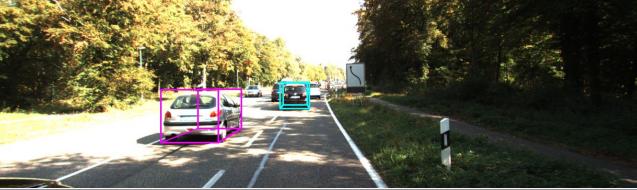
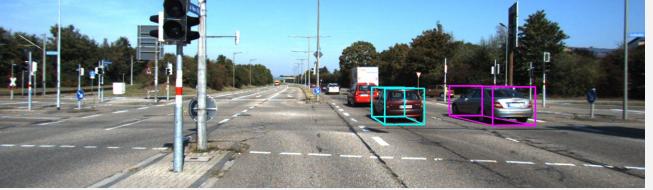
Question: What is the angle between Entity #2 and Entity #1, in degrees?	Question: What is the orientation of Entity #2 relative to Entity #1, similar, opposite or perpendicular?
	
Annotation: The angle between Entity #2 and Entity #1 is approximately 1.72 degrees.	Annotation: Entity #2 is oriented in a similar direction or alignment to Entity #1.
Ours: The angle between Entity #2 and Entity #1 is precisely 0.00 degrees, indicating they are perfectly aligned. ✓	Ours: Entity #2 is oriented in a similar direction or alignment to Entity #1. ✓
GPT-4o: The angle between Entity #2 and Entity #1 is 115.5 degrees. ✗	GPT-4o: The orientation of Entity #2 relative to Entity #1 is a. similar. both vehicles are facing in the same direction along the road. ✓

Figure 16. Examples and predictions from our baseline method and GPT-4o for the Orientation Reasoning (OR) task.

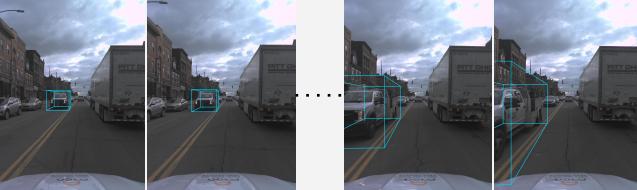
Question: How would you describe the lane position of Entity #1?	Question: How would you describe the lane position of Entity #1?
	
Annotation: Entity #1 is positioned in the lane designated for oncoming traffic.	Annotation: Entity #1 is positioned in the front lane.
Ours: Entity #1 is positioned in the lane designated for oncoming traffic. ✓	Ours: Entity #1 is positioned in the front lane. ✓
GPT-4o: Entity #1 is in the oncoming traffic lane (option d). ✓	GPT-4o: The lane position of Entity #1 is in the **d. oncoming traffic lane**. ✗

Figure 17. Examples and predictions from our baseline method and GPT-4o for the Other Lane to Ego-Vehicle (EGO-LANE) task.

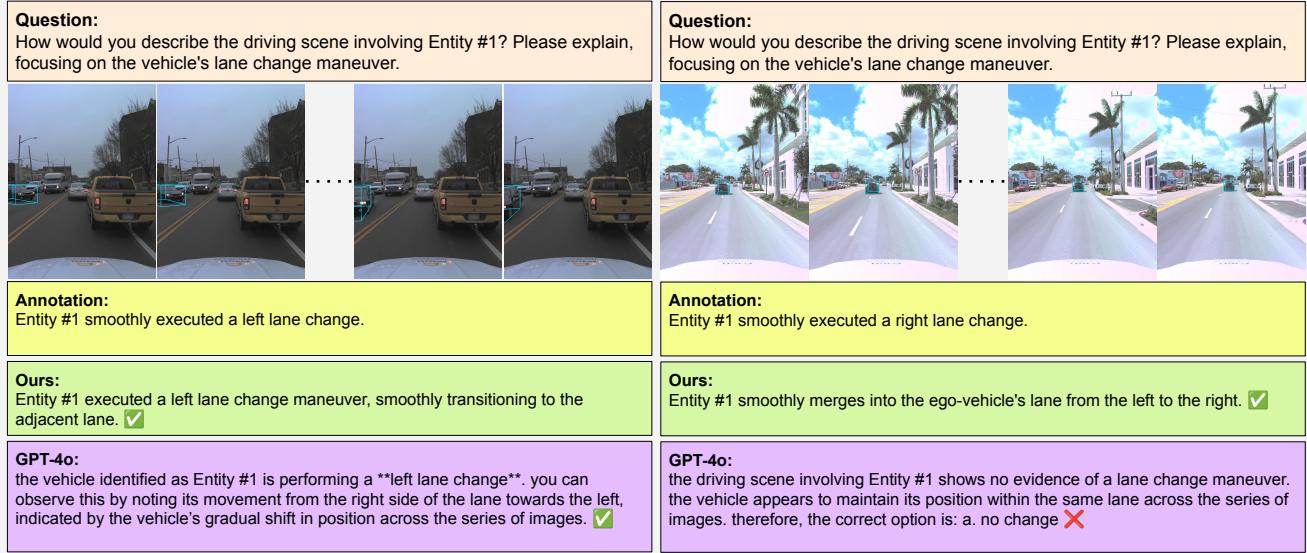


Figure 18. Examples and predictions from our baseline method and GPT-4o for the Other Lane Changing (OBJ-LANE) task.

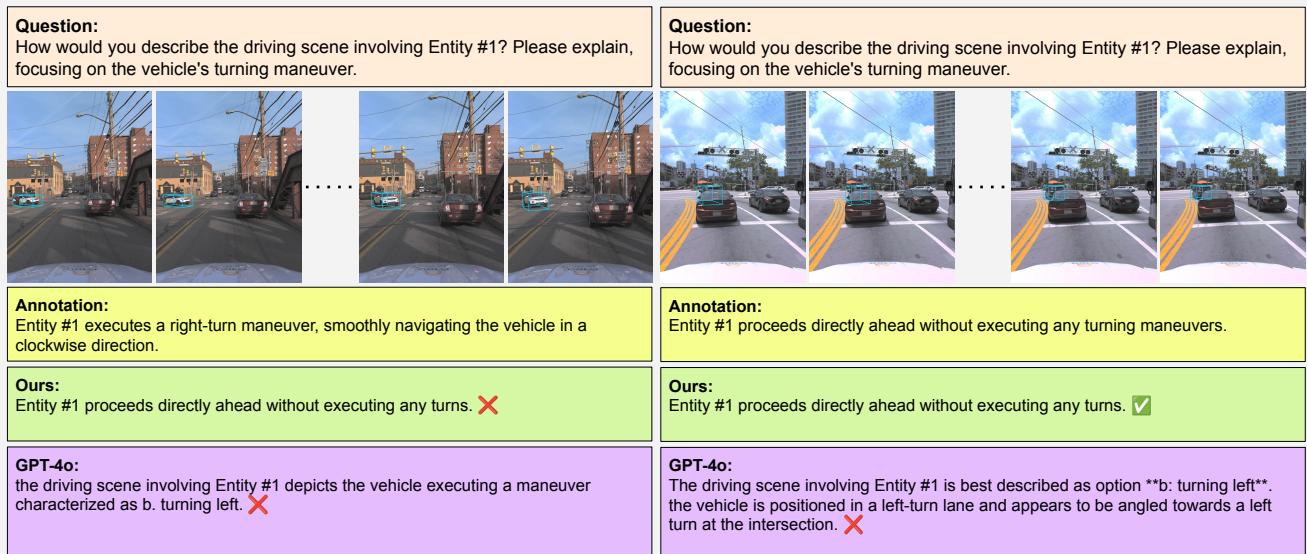


Figure 19. Examples and predictions from our baseline method and GPT-4o for the Other Turning (OBJ-TURN) task.

Question: How would you describe the driving scene involving our car? Please explain, focusing on our car's turning maneuver.	Question: How would you describe the driving scene involving our car? Please explain, focusing on our car's turning maneuver.
	
Annotation: Our car executed a precise right-turn maneuver, smoothly navigating the corner with adept control.	Annotation: Our car executed a precise left-turn maneuver, showcasing excellent handling and driver control.
Ours: Our car executed a precise right turn, smoothly navigating the corner. ✓	Ours: Our car executed a precise right turn, smoothly navigating the corner. ✗

GPT-4o: Based on the sequence of images, the car is making a **right turn** (option c). the car approaches an intersection and moves towards the right-hand lane, making it suitable for turning right. ✓	GPT-4o: The driving scene involving your car shows it traveling straight along the road. there is no indication of a turning maneuver. therefore, the correct option is: a. go straight ✗
---	---

Figure 20. Examples and predictions from our baseline method and GPT-4o for the Ego Turning (EGO-TURN) task.

Question: How far has our car driven and what kind of steering maneuver did it perform in the current scene?	Question: How far has our car driven and what kind of steering maneuver did it perform in the current scene?
	
Annotation: Our car has not driven any distance and is not performing any steering maneuver.	Annotation: Our car has driven 10.83 meters with a straight steering maneuver.
Ours: Our car has driven 1.01 meters with a straight steering maneuver. ✗	Ours: Our car has driven 12.42 meters with a straight steering maneuver. ✓

GPT-4o: The car has driven 6.3 meters, performing a left steering maneuver. ✗	GPT-4o: the car has driven 39.3 meters, and it performed a left lane change maneuver. ✗
---	---

Figure 21. Examples and predictions from our baseline method and GPT-4o for the Ego Traverse Distance (EGO-TRA) task.