

Embodied Scene Understanding for Vision Language Models via MetaVQA

Weizhen Wang, Chenda Duan, Zhenghao Peng, Yuxin Liu, Bolei Zhou
 University of California, Los Angeles

Abstract

Vision Language Models (VLMs) demonstrate significant potential as embodied AI agents for various mobility applications. However, a standardized, closed-loop benchmark for evaluating their spatial reasoning and sequential decision-making capabilities is lacking. To address this, we present MetaVQA: a comprehensive benchmark designed to assess and enhance VLMs' understanding of spatial relationships and scene dynamics through Visual Question Answering (VQA) and closed-loop simulations. MetaVQA leverages Set-of-Mark prompting and top-down view ground-truth annotations from nuScenes and Waymo datasets to automatically generate extensive question-answer pairs based on diverse real-world traffic scenarios, ensuring object-centric and context-rich instructions. Our experiments show that fine-tuning VLMs with the MetaVQA dataset significantly improves their spatial reasoning and embodied scene comprehension in safety-critical simulations, evident not only in improved VQA accuracies but also in emerging safety-aware driving maneuvers. In addition, the learning demonstrates strong transferability from simulation to real-world observation. Code and data will be publicly available at <https://metadiverse.github.io/metavqa>.

1. Introduction

In many real-world robotic applications like autonomous driving and warehouse robots, embodied AI agents have started interacting with physical environments and impacting their surroundings. These agents should be sufficiently aware of their surroundings to interact with their environments safely. In this paper, we define this ability as *embodied scene understanding*, which we believe contains two intertwined facets: *spatial awareness* and *embodied understanding*. Spatial awareness refers to the ability to internalize spatial relationships among observed objects when perceiving the 3D world through the 2D image captured by a monocular camera. Embodied understanding is the ability to relate observed objects egocentrically, foresee the implication of action, and choose the optimal action to achieve

the instructed goal safely.

Recent advances demonstrate the potential of using Vision Language Models (VLMs) as embodied agents in applications from robot arms control [1, 2] to autonomous driving [3]. These tasks share the common components of following instructions, understanding the environment, and taking the optimal action to achieve specified goals. Benefiting from large-scale pre-training, VLMs retain embodied scene understanding to a certain extent. However, their spatial awareness is limited as most VLMs are pre-trained on offline text and images. Meanwhile, their embodied understanding is also constrained because instruction-following interaction with the environment occupies a very small portion of their training data.

In the task of autonomous driving, many prior works [3–9] address this training distribution mismatch by fine-tuning VLMs on Visual-Question-Answering (VQA) tasks tailored for driving scenarios with reported improvements on their benchmarks. However, these benchmarks are not commensurable or suitable for zero-shot evaluation on off-the-shelf general-purpose VLMs. This is because they follow different textual and visual expressions to describe the scene and refer to objects. For example, DriveLM [4] refers to objects by tuples composed of the object identifier, the ID of the corresponding camera, and the pixel positions of the 2D bounding box's vertices in the camera's coordinate. In contrast, in ELM [5], objects are grounded by a triple composed of the character “c” and the pixel coordinates of the center of the 2D bounding box. Not only do these works disagree in description conventions, but their chosen conventions drastically differ from how humans would intuitively refer to an object. A person would point to the object or ground the object by its features (for example, color or shape). This mismatch can weaken the diagnosing power of the VQA datasets: an unsatisfactory performance of a VLM may be caused by its inability to interpret the question expressions rather than its lack of scene understanding capability.

In addition, existing works mainly evaluate embodied scene understanding of VLMs on the VQA task in the open-loop setting. Nevertheless, embodied understanding capability should be examined more thoroughly

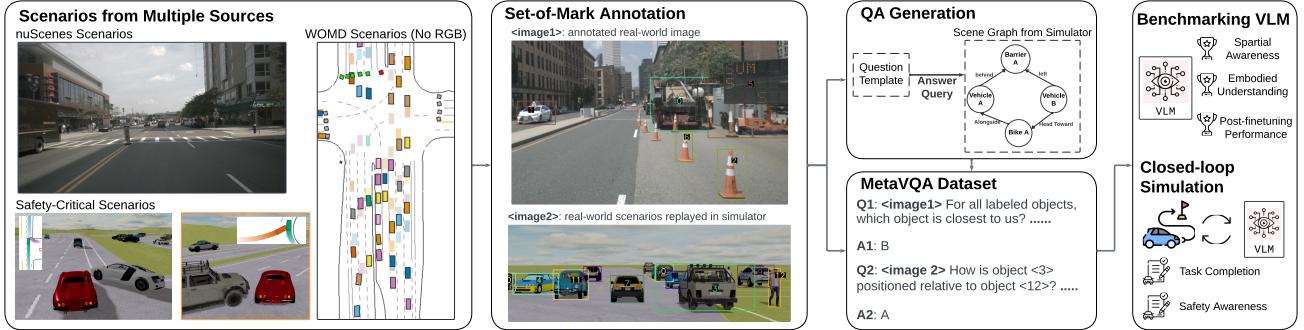


Figure 1. Constructing MetaVQA benchmark. We extract scene graphs from real-world traffic scenarios collected from nuScenes and Waymo datasets(WOMD) and then feed them into question-type-dependent queries to generate ground-truth answers. The real and simulated RGB observations are processed with Set-of-Mark prompting. We evaluate the VLMs on both open-loop VQA tasks and closed-loop navigation tasks in simulation.

in the closed-loop setting, where the VLMs must act in an interactive environment for safe sequential decision-making. DriveVLM [3] attempts the closed-loop evaluation by prompting VLMs to describe the surroundings of a vehicle driven by a human driver. However, involving manual effort in closed-loop evaluations is not scalable, and this setting can not evaluate the embodied understanding of the VLMs since their actions have no consequences for the proceeding environment. Vista [10] uses a trained world model to simulate real-world driving scenarios. However, the observations from the trained world model suffer deteriorated distortion from reality in long-term trajectories. Finally, safety-critical scenarios are absent in existing works since they utilize only real-world data (and therefore, the collection of dangerous situations is costly and unlikely). However, we need a stress test for an embodied agent to verify the safe autonomy of the VLMs. Thus, it is not sufficient to only evaluate the VLMs as embodied agents against normal scenarios.

To address these issues, we present MetaVQA, a benchmark designed for performing the zero-shot evaluation of *general-purpose VLMs* and further improving their embodied scene understanding through finetuning, as shown in Fig. 1. For generalizable learning and evaluation, MetaVQA contains a large-scale and high-quality VQA corpus in common wordings, and its observations—including both real and simulated images from diverse scenarios—are annotated following the Set-of-Mark prompting[11] for clear reference and visual grounding. Furthermore, MetaVQA utilizes the simulation environment MetaDrive [12] for scalable closed-loop evaluation of the VLMs as embodied agents in real-world traffic scenarios collected from nuScenes [13] and Waymo Open Motion Dataset [14]. Models go through stress tests by driving in safety-critical scenarios for more thorough safety evaluations. We have established the baseline performance of many representative VLMs [15–20], and we observe significant improvement in embodied scene understanding af-

ter fine-tuning, shown by the VQA accuracy and the performance of the closed-loop driving task. Additional experiments also verify that *learning on simulated data can substantially improve the embodied understanding in the held-out real-world data*. More specifically, the VLM model fine-tuned on our synthetic VQA data becomes better in nuScene real-world VQA task. We summarize our contributions as follows:

1. We propose the MetaVQA dataset to evaluate and enhance the embodied scene understanding capabilities of the VLMs in a plug-and-go way and bring significant improvement when fine-tuned on our data.
2. We show that the embodied scene understanding of the VLMs is generalizable and transferable from simulated to real-world data. The VLM trained only on simulated data shows remarkable zero-shot performance on real-world visual question answering.
3. We conduct the closed-loop simulation evaluation of the VLMs as embodied agents in safety-critical driving scenarios. The fine-tuned VLMs exhibit reasonable sequential decision-making skills when charged with the unseen task of driving, validating the generalizability of learned embodied knowledge.

2. Related Work

Driving scene understanding datasets. Many datasets have been proposed for driving scene understanding [21–24]. For explaining driving behaviors, some works [4, 7–9, 25–34] provide annotations for scene descriptions, traffic elements, and high-level instructions. For example, BDD-X [32] and Talk2Car [7] supply succinct descriptions of driving scenarios and directions; Rank2Tell [33] and nuScenes-QA [34] focus on ranking object importance and annotating road element attributes, respectively; DriveLM-nuScenes/CARLA [4] utilizes a graph-based visual-question-answering for driving descriptions. Concurrent with this work, DriveMLM [9] inspects spa-

tial understanding in absolute (detailed to meters) and relative descriptions (for example, left or right). Additionally, AutoTrust [8], another concurrent work, focuses on the trustworthiness and safety-awareness of VLMs in driving. However, these datasets often rely on limited traffic data (primarily the nuScenes [13] dataset) and lack diversity in data/question types, limiting their potential for broader, generalizable applications. We aim to address these limitations by enhancing the data scale and providing diverse question types for improved scene understanding.

Vision language models as driving agents. Recent attempts leverage language models to enhance traditional autonomous driving tasks [35–37]. DriveGPT4 [6] delivers an interpretable system, while Lingo-1 [38] integrates vision, language, and action for model training and interpretations. GPT-Driver [39] leverages large language models for trajectory prediction, and ELM [5] and DriveVLM [40] expand capabilities in long-horizon space and handling corner cases. However, these models are often evaluated offline, lacking a closed-loop evaluation framework that captures the interactive nature of driving. Thus, closed-loop evaluation is crucial for assessing the robustness and reliability of vision-language models in real-world driving scenarios.

Closed-loop evaluation of end-to-end driving agents. Driving simulators are crucial for autonomous driving, and they facilitate closed-loop testing before real-world trials. Notable examples include LimSim++ [41], MetaDrive [12], CARLA [42], and nuPlan [43] which use rule-based engines in evaluating driving behavior. However, their evaluation metrics typically focus on numerical values such as success rate and route completion rate, which lack interpretability. To address this limitation, we aim to incorporate scene understanding and the reasonableness of behaviors into the evaluation process by using more diverse and meaningful metrics.

3. Constructing MetaVQA Dataset

3.1. Our Design Principles

We aim to develop a VQA generation pipeline for benchmarking general-purpose VLMs on embodied scene understanding and for fine-tuning them to serve as an embodied agent in the driving task. We consider the following two key questions when designing our dataset:

How to effectively communicate with general-purpose VLMs? An analogy can be drawn for the VLMs performing the VQA task as students taking standardized tests. To fairly evaluate all students’ learning outcomes, the instructor should create a problem set where the question and answering instructions are clear and intuitive. We notice that when evaluating embodied scene understanding, existing works have a variety of heterogeneous prompting conventions and expected answer forms. For example, Driv-

eLM [4], ELM [5], and DriveVLM [3] refer to objects by pixel positions of 2D bounding box’s vertices in the camera’s coordinate. These works expect the models to associate pixel coordinates with regions on the images. However, such association convention is rare in the pre-training data corpus collected from the Web. In addition, ELM prompts models to convey spatial and dynamics (ego speed, heading, *etc.*) information in continuous values (world coordinates, top-down yaw angle, meters per second, *etc.*), contrasting to DriveLM, which discretizes this information. Nevertheless, both conventions are unfamiliar to general-purpose VLMs pre-trained on human-created Internet data. Therefore, if general-purpose VLMs are zero-shot evaluated on these tasks and protocols, it will be difficult to attribute the cause of deficient performance to a lack of embodied scene understanding or an inability to follow the question-answering convention. In contrast, CLEVR [44] uses template-generated English referrals. However, natural language referral can be ambiguous when the scene becomes complicated in layout and appearance, making the questions disputable and hindering fair evaluation.

How to thoroughly evaluate embodied scene understanding? As mentioned in the introduction section, we interpret the embodied scene understanding capabilities from two aspects: understanding spatial relationships and understanding an action’s consequence.

Existing works [3–5, 8, 9, 40] evaluate this capability on the VQA tasks by creating a sufficiently encompassing question suite. However, the metrics for evaluations are incommensurable. ELM introduced “Pr@k” metrics to evaluate the performance on localization tasks, while DriveLM uses classification error as the metrics since they discretize space. An extreme case can be found in DriveMLM [9], in which distinct accuracy metrics are defined for each type of question. In addition, in the driving field, closed-loop egocentric evaluation through interactive simulation is rarely explored, leaving the claim of learned embodied scene understanding untested in situations of actual embodiments.

To address the two questions above, we start with the Set-of-Mark (SoM) prompting technique [11]. SoM elevates the visual grounding capabilities of VLMs and provides an intuitive and unambiguous referral scheme with labeling. The suitability of using SoM prompting is further discussed in Sec. 4.2 and Sec. 4.3. We formulate each question in the multiple-choice setting with only one unique correct option to make fair evaluations. To support zero-shot evaluations of general-purpose VLMs, spatial and dynamics information is discretized into common phrases (for example, “front”) with fine-grained numerical descriptions added as additional explanations. Sec. 4.2 showcases the intuitiveness and descriptiveness of this setup through human study, and the zero-shot performances in the benchmark from Sec. 5.1 empirically validate the suitability. To

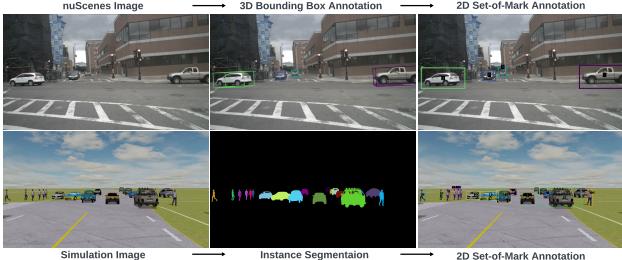


Figure 2. Set-of-Mark annotation process. For real-world images from the nuScenes (upper row) dataset, we cast the corresponding 3D bounding boxes into 2D space. For simulated images (lower row) rendered in MetaDrive, we extract 2D bounding boxes from the simulator’s instance segmentation.

thoroughly examine the embodied scene understanding of VLMs, we construct 30 question types covering all aspects of spatial reasoning and embodied understanding of VLMs with diverse scenarios using both real-world and simulated observations, and we further evaluate VLMs with closed-loop simulation.

3.2. VQA Generation Pipeline

To generate diverse questions from large-scale driving scenarios in scale, we employ a search-based method to programmatically extract answers for template-generated questions from scene graphs and quality-check the dataset with human evaluators as discussed in Sec. 4.2. Fig. 1 illustrates the overall VQA generation pipeline of the MetaVQA. There are three essential stages: scenario aggregation, Set-of-Mark annotations, and QA generation. We describe each stage as follows.

3.2.1. Scenario Aggregation from Multiple Sources.

MetaVQA utilizes real-world traffic from Waymo Open Motion Dataset (WOMD) [45] and nuScenes dataset [13] to evaluate the safety and robustness of the VLMs against diverse situations. The WOMD dataset is much larger than the nuScenes dataset, which contains only around 800 20-second scenarios. However, the WOMD dataset lacks RGB observations, making its diverse scenarios inaccessible directly. Amending the missing observations, we use a lightweight simulator, MetaDrive[12], to reconstruct WOMD traffic scenarios for scalable and efficient simulation. We also create the digital twins of the nuScene dataset in simulation to augment the appearance diversity of collected scenarios.

Scene collection of nuScenes dataset. The well-annotated nuScenes dataset provides a devkit to visualize driving scenarios, and we use it to extract 3D scene graphs from each annotated keyframe. Each node in the graph contains spatial and pose information of objects, and labeled edges—representing spatial relationships—are connected algorithmically. Objects are also filtered by relevancy and

visibility before being loaded in as object nodes in the scene graph, and implementation details can be found in the supplementary materials.

Scene reconstruction with simulator. We also harness ScenarioNet [46] to aggregate the WOMD and the nuScenes datasets into a unified scene record. The MetaDrive simulator loads those scenarios and renders top-down layouts into egocentric RGB images. Following the nuScenes convention, keyframes are selected at 2Hz frequency, and we follow the sensor setup in the nuScenes dataset for cross-domain perspective consistency of observations. We additionally use nuScenes traffic to create digital twins of the original image, making the evaluation of embodied scene understanding less sensitive to variation in lighting and object appearances.

3.2.2. Set-of-Mark Prompting

MetaVQA aims to provide a generalizable evaluation of embodied scene understanding for off-the-shelf VLMs, and Set-of-Mark [11] (SoM) prompting provides a clear avenue for humans to issue instructions and VLMs to visually ground objects of interest. We devoted this work on the reasoning ability of VLMs as embodied agents. Therefore, we assume the perception task as a mostly solved problem, and directly highlighting relevant traffic objects is not an overdue hint. This presumption is supported since various detection methods have been established and widely applied [47–49]. As illustrated in Fig. 2, objects of interest are enclosed by 2D bounding boxes. We extract maximal 2D bounding based on 3D bounding box’s projections for the nuScenes images, and we use a shader-based instance segmentation camera to create annotated simulated images. We utilize the labeling algorithm suggested in the original Set-of-Mark paper to ensure maximal labeling consistency and visibility. Further discussions on the impact of marking style can be found in the supplementary materials.

3.2.3. Question-Answer Generation

Each question is a multiple-choice question. Since we want a generalizable evaluation without domain-specific fine-tuning of VLMs, this simple setup is straightforward and also complete to cover answer space as we abstract numerical spatial information into discrete options and thus can be fully covered as multiple-choice options. As illustrated in Fig. 3, during the question-answer generation process, a random object node in the scene graph is selected to be examined, and the ground truth answer—along with other multiple-choice candidates—is generated based on the scene graph using question-type-bounded queries. To prevent VLM collapses during fine-tuning, we additionally incorporate an “explanation” field in the generated VQAs to explain the selection of answers with dense captions. During training, this field will be part of the ground truth answers VLMs are tasked to generate to deepen their scene

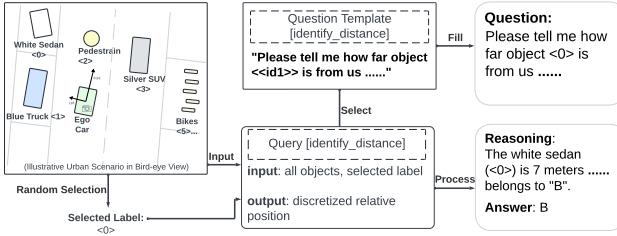


Figure 3. **Question-answer generation pipeline.** An illustrative example for generating the `identify_distance` question. Note that an additional “reasoning” field is generated along with the answer to improve VLM training. This field is not used in evaluation.

understanding. The implementation details can be found in the supplementary materials.

4. MetaVQA Dataset

We present the MetaVQA Dataset. In addition to the dataset composition, we verify the dataset quality and its helpfulness for evaluating embodied scene understanding. We validate that (1) MetaVQA Dataset’s Set-of-Mark annotated questions are suitable for zero-shot evaluation of general-purpose VLMs. (2) The embodied scene understanding capability that emerges from the finetuning is generalizable and transferable from the simulated world to the real world, supporting the utility of simulator-conditioned synthetic data. (3) The learning scales with the training set size using MetaVQA Dataset, warranting the need for a large-scale question-answer corpus.

4.1. Dataset Composition

MetaVQA Dataset consists of a large corpus of multiple-choice questions, which contains 4,305,450 questions using 442,102 annotated frames extracted from 400 nuScenes scenarios and 6,900 Waymo scenarios covering 59,682 seconds (16.5 hours) of driving log. The questions can be categorized into three supercategories: spatial questions, embodied questions, and grounding questions. The former two supercategories cover the two facets of embodied scene understanding: spatial awareness and embodied understanding, and the latter one diagnoses VLMs’ capabilities to associate marked objects in the observation with textual referral. Detailed formulation for each subcategory is discussed in the supplementary materials, and we lay out the compositions and exemplar VQA pairs in Fig. 4. Noticeably, we design a special training-only `describe_scenario` question that asks models to describe all labeled objects in the observation to accentuate embodied scene understanding. For expedited experimentation, we curate a representative training set of 150,000 questions, with 50,000 coming from Waymo scenarios, 50,000 coming from nuScenes scenarios, and 50,000 coming from

the simulated nuScenes scenarios in the simulator. In addition, we curate a withheld test set of 9,725 questions with 2,524 annotated frames from 212 traffic scenarios. Approximately half of the questions use observations reconstructed from simulation, and the other half use real-world images.

4.2. Zero-shot Answerability with Set-of-Mark Prompting

Human evaluation. We create a questionnaire comprising 35 sampled questions and distribute it to six novice human participants, who are instructed to answer the questions without assistance. Since VLMs are trained on human-generated texts, good human amateur performance suggests the intuitiveness of the wordings in the question bodies and the clarity in permitted answers. Upon reporting, novice participants achieve an average of 88% accuracy despite acknowledging question difficulty. This finding showcases the answerability and directness of the MetaVQA Dataset. Details are included in the supplementary materials.

Set-of-Mark prompting. Previous work verifies that using Set-of-Mark prompting on real pictures improves the visual grounding abilities of VLMs [11]. To further validate that VLMs can associate referred labels in text with marked regions in both simulated and real images, we introduce grounding questions. Objects are labeled randomly for each such question, and one of the labels is enclosed with a white box. As illustrated in Fig. 4, VLMs are prompted to identify the singular correct label located inside the box. We benchmark the zero-shot grounding performance on multiple VLMs on the withheld test set from Sec. 4.1, as shown in Tab. 1, VLMs achieve, on average, 69.6% zero-shot accuracies on the 467 grounding questions, with the best-performing VLM achieving 87.4% accuracies. Noticeably, LLaVA-NeXT (llava-1.6-vicuna-7b) [15] underperforms significantly, and this abnormality is caused by the VLM failing to generate tokens in the expected convention, which will be discussed in Sec. 5.1. This finding suggests that most VLMs can accurately associate labels referred in text with objects. Therefore, using labels for object referral in the question body is an unambiguous and effective wording convention, and we can be confident in associating bad testing performance on the MetaVQA Dataset with a deficiency in embodied scene understanding.

4.3. Transfer learning with simulated observations

A concern can be raised against the use of simulated images. The MetaDrive simulator [12], while capable of importing real-world traffic scenarios, falls short in visual fidelity compared to real-world photos. We conduct four trials on the withheld test set from Sec. 4.1 using InternVL2-8B [50] as the learning VLM to address this concern. In the first trial, InternVL2-8B is evaluated without any finetuning. In the next three trials, InternVL2-8B is fine-tuned

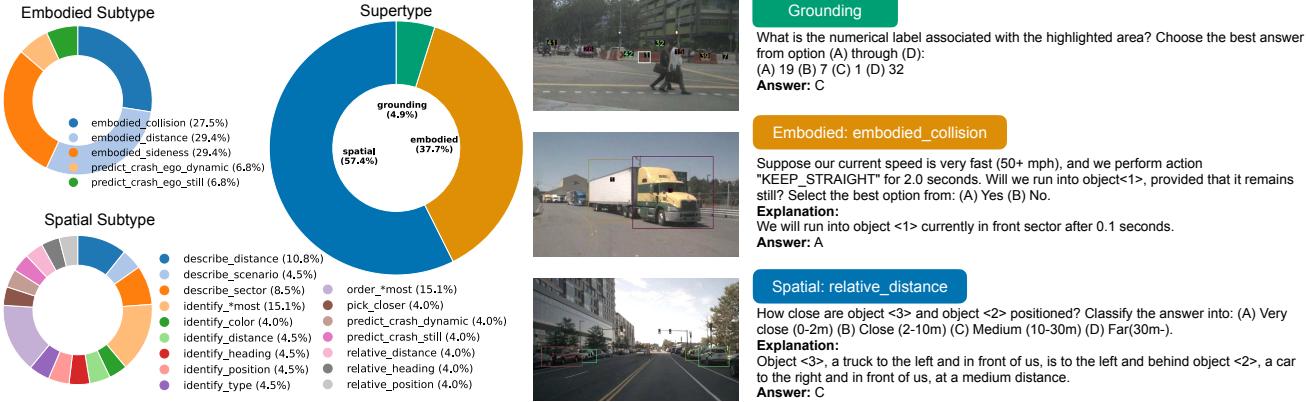


Figure 4. **Left:** Distribution of the question types. **Right:** Example for each question supertype.

Model	Overall	Sim-Grounding	Real-Grounding
LLaVA-NeXT [15]	0.248	0.271	0.229
LLaVA-OneVision [16]	0.728	0.615	0.827
GPT-4o [18]	0.831	0.766	0.888
Qwen2 [17]	0.874	0.890	0.859
Llama3.2 [19]	0.790	0.716	0.855
InternVL2-8B [50]	0.702	0.610	0.783
Average	0.696	0.644	0.740

Table 1. **Zero-shot grounding accuracy.** Most VLMs successfully associate label referrals in the question body with marked regions, showing that Set-of-Mark prompting effectively conveys instruction.

Training Set	Overall	Sim	Real	Training Size	Overall	Sim	Real
Zeroshot	0.592	0.552	0.632	9,375	0.794	0.764	0.824
Sim only	0.807	0.795	0.819	37,500	0.845	0.825	0.865
Real only	0.825	0.792	0.858	150,000	0.869	0.853	0.884
Sim+Real	0.869	0.853	0.884				

Table 2. **Sim-to-Real transferability.** InternVL2-8B achieves the best test accuracy when trained using both simulated and real images, indicating the transferability of embodied knowledge learned from the MetaVQA Dataset.

on (1) 50,000 questions with only simulated observations, (2) 50,000 questions with only real images, and (3) 150,000 questions with both simulated and real observations. The third training set comprises the second set and a superset of the first training set.

Referring to Tab. 2, InternVL2-8B achieves the best test accuracy when learning from both simulated and real observations. Noticeably, training on simulated observations alone significantly increases real-world image test accuracy, and training on real observations benefits simulated image test performance reciprocally. Therefore, the learned embodied scene understanding is transferrable between simulation and the real world, and this finding warrants the benefits of adding simulated observation into the training data.

Table 3. **Data scalability.** We observe a positive correlation between InternVL2-8B’s test performance and training data size. This finding warrants the scalability of learning using the MetaVQA Dataset.

4.4. Data Scalability of Learning

We conduct this experiment to demonstrate the positive correlation between learned embodied understanding and the size of the training set to support the need for a large VQA corpus. Using the training set specified in Sec. 4.1, we gradually down-sample the questions with a factor of 4, generating training sets of 150,000 questions, 37,500 questions, and 9,375 questions correspondingly. We fine-tune InternVL2-8B on each of the three sets and report the test accuracy on the withheld test set specified in Sec. 4.1. As illustrated in Tab. 3, InternVL2-8B’s test performance scales with the training set size, warranting the encompassing size of the MetaVQA dataset.

5. Benchmark Results

5.1. Visual Question Answering

Task formulation & Metrics. Under this task, VLMs receive multiple-choice questions and corresponding images annotated with Set-of-Marks prompting. The model is trained to select the best matching options provided in the question bodies. As shown in Fig. 4, the model is instructed to answer in single capitalized characters. We evaluate the model based on its failure rate (of outputting valid responses) and the accuracy of the selected options. Implementation for the answer parsing will be discussed in the supplementary materials. We benchmark the zero-shot performance of state-of-the-art VLMs on the curated test set mentioned in Sec. 4.1. In addition, we establish the performance of VLMs fine-tuned on the training set mentioned in Sec. 4.1.

Benchmarks. Tab. 4 lists the performance of various VLMs on the withheld test set, and Fig. 7 visualize the performance across question supertypes (along with the total test accuracy in the “overall” dimension). As



Figure 5. **Improved embodied scene understanding after fine-tuning** of InternVL2-8B on the withheld training set from Sec. 4.1. The VLM demonstrates improved spatial understanding and embodied knowledge after learning the MetaVQA Dataset. In addition, the model attains better grounding capability.

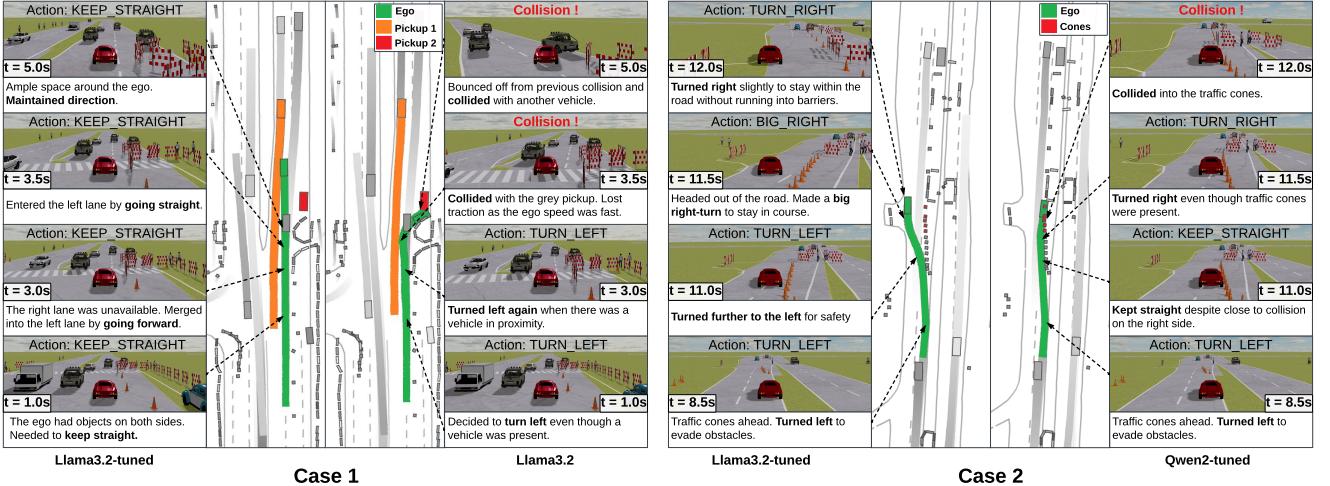


Figure 6. **Qualitative result of closed-loop evaluation.** Case 1 compares the performance of fine-tuned Llama3.2 (left) versus its zero-shot counterpart (right) in the same scenario. Case 2 compares the performance of fine-tuned Llama3.2 (left) versus fine-tuned Qwen2 (right). As shown, fine-tuned Llama3.2 gains elevated situational awareness and can avoid collision. It also demonstrates superior safety capability compared to its trained peers.

shown, most models successfully generate valid token sequences, and their zero-shot performances are significantly better than random guessing, indicating that some embodied scene understanding capabilities are already present. GPT-4o achieves the best zero-shot performance as it contains the largest parameters with an unprecedented scale of pre-training data. In addition, the low parse fail rate in most baselines validates the answerability of questions. Noticeably, LLaVA-NeXT [15] is the only outlier, with a zero-shot accuracy lower than random guessing. The shocking underperformance of LLaVA-NeXT is attributed to two factors. To begin with, the VLM fails

to generate legal answer tokens consistently (failure rate of 27%). In addition, the model, quite frequently, refuses to answer the asked questions. For example, when asked with `relative_distance` questions, LLaVA-NeXT typically responds with "...I cannot provide a definitive answer to your question without more information about the simulation or the specific positions of the objects within it." Consequently, LLaVA-NeXT reports surprisingly bad metrics.

The subfigure (b) of Fig. 7 emphasizes the improvements of the benchmarked VLM after fine-tuning. The dotted contours draw out the zero-shot performance of VLMs,

Model	Overall	Sim	Real	Parse Fail↓
Random	0.329	0.326	0.332	0.0000
LLaVA-NeXT	0.295	0.287	0.302	0.2750
LLaVA-OneVision	0.581	0.550	0.613	0.0000
GPT-4o	0.628	0.602	0.655	0.0004
Qwen2	0.539	0.527	0.552	0.0000
Qwen2-finetuned	0.844	0.839	0.848	0.0000
Llama3.2	0.500	0.478	0.523	0.0080
Llama3.2-finetuned	0.774	0.744	0.803	0.0672
InternVL2-8B	0.592	0.552	0.632	0.0000
InternVL2-8B-finetuned	0.869	0.853	0.884	0.0000

Table 4. **Visual question answering benchmark.** Performance comparison of different models on overall, simulation-only-part, and real-only-part of the withheld test sets. The parsing failure rate is also provided. Models report consistent improvements after fine-tuning, with InternVL2-8B achieving the best performance.

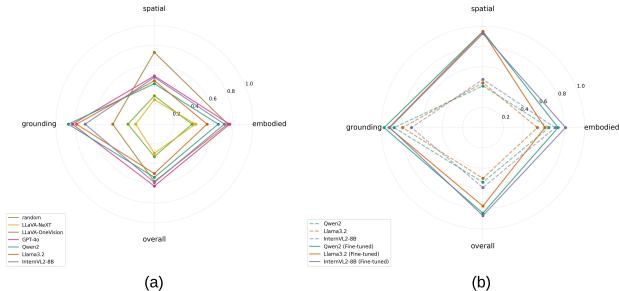


Figure 7. **Performance contours of VLMs.** Sub-figure (a) draws the zero-shot performance comparisons, and sub-figure (b) illustrates the fine-tuning improvements. As illustrated, fine-tuned models perform consistently better on all supertypes of questions, and the improvements in spatial understanding are especially pronounced. In addition, models report comparable increases across question supertypes, suggesting that the MetaVQA Dataset is generally learnable.

and the solid contours describe their fine-tuned counterparts. As shown, fine-tuning on the MetaVQA Dataset results in elevated embodied scene understanding in general, with the most pronounced improvements observed in spatial questions. In addition, comparable gains in accuracy (along each question supertype) are reported across models, suggesting the generalizability of learning for the dataset. We take InternVL2-8B, the best performing VLM after fine-tuning, for a case study: Fig. 5 showcases questions successfully answered by the model after fine-tuning. The model not only gains spatial awareness with a better understanding of pedestrians' projected heading but also shows improved attentive power by reasoning about multiple observed objects. In addition, InternVL2-8B shows improved embodied knowledge: it forecasts the potential hazard of running into traffic barriers and predicts the consequence of its action. Collectively, the improvements in both spatial reasoning and embodied knowledge of VLMs demonstrates the contribution of the MetaVQA Dataset, and we will further validate its effectiveness in Sec. 5.2.

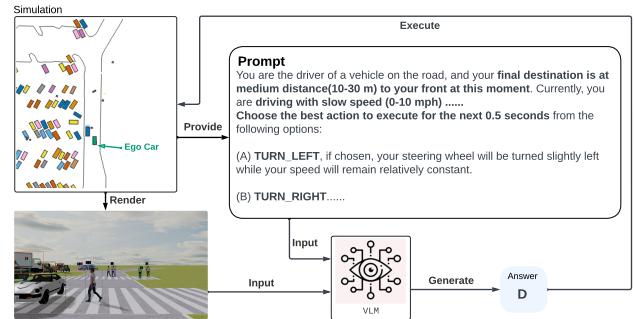


Figure 8. **Formulation of closed-loop evaluation.** At every five simulation steps (0.5 seconds wall time), the evaluated VLM is provided with annotated observations and current navigation command. The chosen action will be fed into the simulation.

5.2. Closed-loop Evaluation

Task formulation. We use the MetaDrive [12] simulator and load scenarios from real-world datasets through ScenarioNet [46] to create interactive environments to test VLMs as embodied agents. In this task, VLMs are tasked as the self-driving planners of vehicles. Fig. 8 illustrates the interaction paradigm. At every five simulation steps (0.5s in the wall time), the tested VLM receives a first-person view image annotated in the Set-of-Mark convention. The destination, current speed, and allowed actions are provided as textual prompts. Each action is mapped to a fixed value pair of steering and acceleration, which will be fed into the simulator for vehicle dynamic simulation. Further details are discussed in the supplementary materials. For generalizable evaluations across varied situations, we augment real-world scenarios with safety-critical scenarios generated using CAT [51], in which an adversarial agent will intercept the trajectory of the ego vehicle. During evaluation, the trial is considered a failure if the tested VLM drives off the roads, upon which the trial terminates. Details on implementation can be found in the supplementary materials.

Experiments Setup. We curate 120 driving scenarios, with 60 scenarios selected from the nuScenes dataset and 60 safety-critical scenarios generated using CAT from WOMD's [45] training split. All scenarios are handpicked to cover diverse behaviors such as stop-and-go, U-turn, and crossing intersections. We evaluate learned knowledge gained from the open-loop VQA task using the InternVL2 model family, Llama3.2, and QwenVL2, each trained on a shared condensed training dataset from the larger training set from Sec. 4.1.

Noticeably, the prompt format used in the closed-loop evaluation is absent from the MetaVQA Dataset. However, the skill sets required for driving–spatial awareness and embodied knowledge—are practiced by fine-tuning with the MetaVQA Dataset. Therefore, we expect fine-tuned

Model	Test Accuracy↑	Route Completion↑	Off-Road Rate↓	Collision Rate↓	ADE↓	FDE↓
random	-	0.376	0.658	0.375	20.937	15.207
brake	-	0.188	0.167	0.533	23.225	44.659
straight	-	0.592	0.583	0.358	13.272	12.678
Qwen2	0.539	0.615	0.583	0.367	13.075	30.214
Qwen2-tuned	0.792	0.667	0.442	0.300	14.873	27.973
Llama3.2	0.500	0.529	0.658	0.483	18.335	40.665
Llama3.2-tuned	0.757	0.632	0.558	0.267	15.118	31.811
InternVL2-4B [50]	0.507	0.259	0.850	0.225	24.824	53.020
InternVL2-4B-tuned	0.719	0.614	0.500	0.317	14.368	29.943
InternVL2-8B	0.592	0.637	0.583	0.325	13.361	30.520
InternVL2-8B-tuned	0.820	0.657	0.517	0.367	13.109	27.873

Table 5. **Quantitative result of closed-loop evaluation.** Despite not being directly trained on the driving task, VLMs report improvements in closed-loop metrics after learning the MetaVQA Dataset, in addition to better VQA accuracy. This correlation suggests that the MetaVQA Dataset contains generalizable embodied knowledge that could be easily learned and transferred to the downstream application domain (in this case, self-driving). Note that the fine-tuning set is a condensed version of the training set specified in Sec. 4.1 to expedite experiments.

VLMs should learn generalizable embodied scene understanding for the unseen driving task, and models are tasked as self-driving planners in a closed-loop setting.

Metrics. For each evaluated VLM, we record the collision rate, defined as the ratio of scenarios in which the VLMs collided over the total number of scenarios, off-road rate, average displacement error (ADE), final displacement error (FDE), and route completion ratio. Furthermore, we append its accuracy on the test set mentioned in Sec. 4.1 to examine the correlation between open-loop performance and closed-loop performance of VLMs. Details on metrics implementation can be found in the supplementary materials.

Discussions. As presented in Tab. 5, all fine-tuned VLMs exhibit significantly better closed-loop performance in route completion, off-road ratio, and FDE, in addition to improved VQA accuracy. With a few exceptions, models also demonstrate improved ADEs and collision rates in general. We speculate these inconsistencies are a by-product of varied pre-training strategies adopted by each VLM, leading to differed learning of situational awareness. The improved zero-shot closed-loop performance of fine-tuned VLMs confirms our expectation: general-purposed VLMs become stronger adaptive embodied agents after learning the MetaVQA Dataset.

Fig. 6 provides case comparisons of VLMs in closed-loop evaluation, with the ego trajectory drawn in green while those of interested objects in orange or red. The decision-making of fine-tuned Llama3.2 [19], the safest driver after fine-tuning, is plotted on the left side in both cases. In case 1, Llama3.2’s behavior before fine-tuning is plotted against on the right. Without the open-loop VQA pre-training, naive Llama3.2 fails to realize the danger of

turning-left when a pickup was present on that side, resulting in two consecutive collisions. In comparison, fine-tuned Llama3.2 becomes more aware of its surroundings. It deduces that the safest action, when surrounded by obstacles on both sides, is to main direction, and it successfully merges into the unblocked left lane by keeping straight consistently. In case 2, the right side illustrates the trial for fine-tuned Qwen2 [17]. Both fine-tuned VLMs demonstrate situational awareness by merging into the left lane at wall time 8.5 second, when traffic cones are present ahead. However, they improve to varied degrees: Llama3.2 keeps going left until there it is sufficiently distant from the obstacles, upon when it makes a right turn to stay within the road. In comparison, Qwen2 becomes oblivious to the traffic cones to its right, and it decides to turn right when there is no space, leading collisions. In both comparisons, consisting of four trials, the VLMs receive the same navigation command “forward” throughout simulation (complying with the ground-truth trajectories), meaning that the evasions are made on-the-go without behavioral hints from the navigation command. We further supports from improved closed-loop evaluations, we are confident to claim that the MetaVQA Dataset contributes to general-purpose VLMs’ embodied scene understanding.

6. Conclusion

We present MetaVQA, a large-scale benchmark for evaluating and improving the embodied scene understanding of vision language models. Besides establishing the baseline performance of representative VLMs, we showcase the transferable knowledge learned from the MetaVQA Dataset across observation domains through sim-to-real VQA evaluations. Finally, we further evaluate the fine-tuned VLMs in the untrained task of closed-loop driving in simulation, and the significantly improved driving capabilities confirm the generalizability and robustness of learned embodied knowledge from the MetaVQA Dataset.

Limitations. Currently, MetaVQA contains only images as observations, but embodied agents might need multi-step historical information to make the optimal decision in complex situations. In addition, the MetaVQA Dataset contains only single-perspective observations captured from fixed angles, while multi-camera observations could provide more contextual information for better decision-making.

References

- [1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan

- Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspair Singh, Anikait Singh, Radu Soricu, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023. 1
- [2] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailev, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1
- [3] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Zhiyong Zhao, 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. 1, 2, 3
- [4] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelml: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023. 1, 2, 3
- [5] 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. 1, 3
- [6] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023. 3
- [7] Thierry Deruyttere, Simon Vandenhende, Dusan Grujacic, Luc Van Gool, and Marie Francine Moens. Talk2car: Taking control of your self-driving car. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2088–2098, 2019. 2
- [8] Shuo Xing, Hongyuan Hua, Xiangbo Gao, Shenzhe Zhu, Renjie Li, Kexin Tian, Xiaopeng Li, Heng Huang, Tianbao Yang, Zhangyang Wang, Yang Zhou, Huaxiu Yao, and Zhengzhong Tu. AutoTrust: Benchmarking Trustworthiness in Large Vision Language Models for Autonomous Driving. *arXiv*, December 2024. 3
- [9] Xianda Guo, Zhang Ruijun, Duan Yiqun, He Yuhang, Chenming Zhang, and Long Chen. Drivemllm: A benchmark for spatial understanding with multimodal large language models in autonomous driving. *arXiv preprint arXiv:2411.13112*, 2024. 1, 2, 3
- [10] Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2
- [11] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 2, 3, 4, 5, 15, 16, 17
- [12] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3, 4, 5, 8, 14, 19
- [13] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yuxin Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 2, 3, 4, 14
- [14] 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. 2
- [15] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 2, 5, 6, 7, 18, 19, 20
- [16] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 6, 19, 20
- [17] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6, 9, 15, 17, 19, 20
- [18] OpenAI. Chatgpt-4, 2024. Large Language Model developed by OpenAI. 6, 19, 20
- [19] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korvaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade

Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junting Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenying Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Couder, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel

Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Diadem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghatham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajai, Victoria Montanez, Vijai Mohan,

- Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaoqian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary De-Vito, Zef Rosnbrick, Zhao Duo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. [6](#), [9](#), [19](#), [20](#)
- [20] Zhe Chen, Jiannan Wu, Wenhui Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023. [2](#), [18](#), [20](#)
- [21] Dongming Wu, Wencheng Han, Tiancai Wang, Yingfei Liu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. *arXiv preprint arXiv:2309.04379*, 2023. [2](#)
- [22] 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 *CVPR*, 2020.
- [23] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring Multi-Object tracking. In *CVPR*, 2023.
- [24] Hongyang Li, Yang Li, Huijie Wang, Jia Zeng, Huilin Xu, Pinlong Cai, Li Chen, Junchi Yan, Feng Xu, Lu Xiong, Jingdong Wang, Futang Zhu, Chunjing Xu, Tiancai Wang, Fei Xia, Beipeng Mu, Zhihui Peng, Dahua Lin, and Yu Qiao. Open-sourced data ecosystem in autonomous driving: the present and future, 2024. [2](#)
- [25] Jinkyu Kim, Teruhisa Misu, Yi-Ting Chen, Ashish Tawari, and John Canny. Grounding human-to-vehicle advice for self-driving vehicles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [2](#)
- [26] Ali Keysan, Andreas Look, Eitan Kosman, Gonca Gürsun, Jörg Wagner, Yu Yao, and Barbara Rakitsch. Can you text what is happening? integrating pre-trained language encoders into trajectory prediction models for autonomous driving. *arXiv preprint arXiv:2309.05282*, 2023.
- [27] 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.
- [28] Jessica Echterhoff, An Yan, Kyungtae Han, Amr Abdelraouf, Rohit Gupta, and Julian McAuley. Driving through the concept gridlock: Unraveling explainability bottlenecks. *arXiv preprint arXiv:2310.16639*, 2023.
- [29] Xinpeng Ding, Jianhua Han, Hang Xu, Wei Zhang, and Xiaomeng Li. HiLM-D: Towards high-resolution understanding in multimodal large language models for autonomous driving. *arXiv preprint arXiv:2309.05186*, 2023.
- [30] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023.
- [31] Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. Adapt: Action-aware driving caption transformer. *arXiv preprint arXiv:2302.00673*, 2023.
- [32] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. [2](#)
- [33] 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. [2](#)
- [34] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. NuScenes-QA: A multi-modal visual question answering benchmark for autonomous driving scenario. *arXiv preprint arXiv:2305.14836*, 2023. [2](#)
- [35] Amine Elhafsi, Rohan Sinha, Christopher Agia, Edward Schmerling, Issa Nesnas, and Marco Pavone. Semantic anomaly detection with large language models, 2023. [3](#)
- [36] Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-agent motion forecasting as language modeling. In *ICCV*, 2023.
- [37] Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. LanguageMPC: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023. [3](#)
- [38] Wayve. Lingo-1. <https://wayve.ai/thinking/lingo - natural - language - autonomous - driving/>, 2023. [3](#)
- [39] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. GPT-Driver: Learning to drive with GPT. *arXiv preprint arXiv:2310.01415*, 2023. [3](#)
- [40] 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. [3](#)
- [41] Daocheng Fu, Wenjie Lei, Licheng Wen, Pinlong Cai, Song Mao, Min Dou, Botian Shi, and Yu Qiao. Limsim++: A closed-loop platform for deploying multimodal llms in autonomous driving. *arXiv preprint arXiv:2402.01246*, 2024. [3](#)
- [42] 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. [3](#)
- [43] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021. [3](#)

- [44] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 3
- [45] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021. 4, 8, 14
- [46] Quanyi Li, Zhenghao Peng, Lan Feng, Zhizheng Liu, Chenda Duan, Wenjie Mo, and Bolei Zhou. Scenarionet: Open-source platform for large-scale traffic scenario simulation and modeling. *Advances in Neural Information Processing Systems*, 2023. 4, 8, 14, 19
- [47] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018. 4
- [48] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 4
- [50] Zhe Chen, Jiannan Wu, Wenhui 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. 5, 6, 9, 19
- [51] Linrui Zhang, Zhenghao Peng, Quanyi Li, and Bolei Zhou. Cat: Closed-loop adversarial training for safe end-to-end driving. In *7th Annual Conference on Robot Learning*, 2023. 8, 21

A. VQA Generation Pipeline

A.1. Scenario Aggregation from Multiple Sources

Scene collection of nuScenes dataset. To collect nuScenes scenarios with the original observations (nuScenes-real), we use the Python implementation of nuscenes-devkit [13] to explore traffic scenarios. Following the naming paradigm provided in the official nuScenes documentation, for each sample in a scene, we extract its CAM_FRONT sampled_data. At this point, we can associate a recorded keyframe with its image observation. In the first filtering process, if an object is annotated with less than a “3” level of visibility or is scanned by less than five rays of Lidar, then it is considered “invisible,” and its information will not be recorded. However, this first pass doesn’t consider visual occlusion since the visibility of objects is annotated on the scene level instead of the frame level in nuScenes. Therefore, a second filtering pass is instigated. The nuscenes-devkit provides API to project 3D bounding boxes(8 vertices) of objects onto the 2D image observations, and we create the maximum enclosing 2D filled bounding boxes of the 8 vertices. Then, these filled rectangles are painted onto a black image following the distance order of objects to mimic the process of z-buffering, and boxes of distant objects will be overlayed by closer objects. Finally, we filter out objects with less than 50% of their 2D boxes visible in the composed image, completing the second filtering pass. The third and last filtering pass removes miscellaneous objects such as debris and vegetation from objects of interest. After these three filtering stages, the interested objects set will have the information mentioned in the color box to the right recorded. Notably, the nuScenes dataset doesn’t have the “color” annotation, and we leave this field empty while collecting the scenarios.

Scene reconstruction with simulator. Leveraging the MetaDrive [12] simulator and ScenarioNet [46] data platforms, we aggregate nuScenes [13] and Waymo [45]. For simulator-reconstructed traffic scenarios, we record frames every five steps (0.5 seconds wall time) until the end. We set a camera with a 60-degree field-of-view and 1920×1080 resolution to extract rendering. At each simulation step, we record the following information about the ego and objects

within 75 meters of the ego:

Information Recorded per Frame:

- id**, assigned by the simulator.
- color**, bound to the 3D asset.
- height**, bound to the 3D asset.
- type**, bound to the 3D asset.
- bounding box** in world coordinates.
- heading vector** in world coordinates.
- speed** of the object in meters per second (m/s).
- position** of the center point in world coordinates.
- ego camera that observes the vehicle** (if any).
- visibility** of the object to the ego vehicle.
- collided objects** (if any) at this moment.

Note that if an object is “visible,” the camera must capture at least 1,200 pixels of its body. This is implemented by assigning an ID color to each active object in the simulation, and we use a special instance segmentation camera (the same intrinsic and placement as the capturing camera) to capture the ID-color-based rendering. The traffic collected has the following statistics.

Construction of 3D scene graphs. Each scene graph comprises nodes connected by directed edges representing relative spatial relationships. Each node corresponds to a visible object from the frame information recorded from the previous step, and intrinsic properties (*e.g.* color, height) are contained in the node. Given a reference vector V , we determine the relative spatial relationships between current node A and node B by:

Relative Spatial Relationships (box A, B; front vector V):

left or right. Refer to Fig. 9, and we determine the leftmost and rightmost vertices of bounding box A using the reference vector V as the front direction. Then, if all vertices of bounding box B are to the left of the leftmost vertex of A, then we consider B’s sidedness to be “left” (and similarly for sidedness to be “right”). If bounding box B satisfies neither of the two conditions, then we consider B’s “sidedness” to be “none”.

front or back. We determine this relationship similarly to determining “left” or “right”, with the modification that V is the left direction.

This reference vector V is the heading of the ego vehicle when determining “left or right”, and it’s rotated 90 degrees counterclockwise with respect to the yaw axis when determining “front or back”. Once we have the two values for

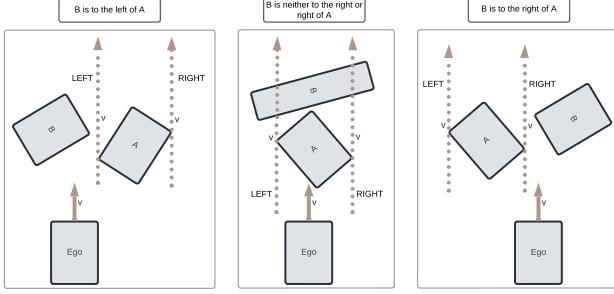


Figure 9. **Top-down illustration of sidedness.** We demand all vertices of box B reside in the “LEFT” region of A for B to be considered “to the left of” (and similarly for “to the right of”) A.

“left or right” and “front or back”, we draw the corresponding directed edge from A to B from the following:

Named Spatial Edges:

- l**, corresponding to “to the left of.”
- lb**, corresponding to “to the left and behind.”
- lf**, corresponding to “to the left and in front of.”
- b**, corresponding to “behind.”
- f**, corresponding to “in front of.”
- r**, corresponding to “to the right.”
- rb**, corresponding to “to the right and behind.”
- rf**, corresponding to “to the right and in front of.”

For example, if “l” edge is chosen, this means “B is to the left of A.”

A.2. Set-of-Mark Prompting



Figure 10. **Instance segmentation masks.** Approximated instance segmentation is generated for real images from the nuScenes dataset. Simulated images are paired with precise instance segmentation.

From Appendix A.1, we have collected image observations and the corresponding instance segmentation in approximated boxes(nuScenes images) or shape-precise masks(simulated images), as shown in Fig. 10. Then, we

run the algorithm illustrated in Fig. 11 adopted from the original Set-of-Mark paper [11] to determine the appropriate position for object labels:

```
// Find center for a region
def Find_Center(r)
    D = DT(r) // Run distance transform
    c = arg max(D) // Find maximum location
    return c

// The main function
def Mark_Allocation(R):
    R̂ = Sorted(R) // Sort regions in ascending
                    order of areas
    for k in range(K): do
        r_k = R̂[k] & ¬R̂[:k - 1].sum(0) // Exclude
                                         k - 1 regions
        C[k] = Find_Center(r_k)
    end
    return C
```

Figure 11. **Labeling algorithm adopted from the Set-of-Mark paper.** Credit to the Set-of-Mark authors.

The Set-of-Mark paper suggested various schemes to perform the visual prompting. For example, using instance segmentation masks and contours are both valid schemes to improve the visual grounding capabilities of vision-language models (VLMs). As mentioned in the main paper, we conducted an ablation study on different prompting schemes to determine the optimal scheme for referal clarity using labels. Using Qwen2 [17] as the zero-shot evaluating model, we fix the prompting scheme with bounding-box annotations, black text background color, and a text size of 1.00 (to reduce label occlusions). The bounding boxes and texts use colors identical to that of the instance segmentation masks of corresponding objects. We use `cv2.rectangle` to draw the bounding boxes onto original images with `thickness = 2`, and we use `cv2.putText` with `font_size = 1` and `thickness = 2`. In addition, we slightly relocate the labels if their corresponding 2D bounding boxes enclose regions less than 1,600 pixels. This is to ensure the visibility of highlighted objects after the visual prompting. The concrete code implementations will be released.

A.3. Question-Answer Generation

A.3.1. Question Generation

MetaVQA adopts a template-based question generation process. Each type of question is bonded to a single template with varying numbers and types of parameters to be replaced with concrete values. We categorize questions into “non-parameterized” and “parameterized” based on the number of parameter types in the template.

Parameterized question generation. Parameters are present for the templates of these questions. These parameters will be replaced upon question generation with concrete values selected from corresponding parameter spaces, the summary of which is provided in Fig. 12. The generation process for a parameterized question is illustrated in Fig. 13: the template of `identify_distance` contains a single `<id1>` parameter, the parameter space of which is all valid labels generated from the Set-of-Mark prompting. In this example, `<id1>` is replaced by the randomly selected label `<0>`. Additionally, multiple parameters belonging to different types can co-exist in a single-question template. Refers to Fig. 14 for an illustration. Observe how concrete values for parameters are sampled from the parameter spaces.

<speed> space: slow (0-10 mph) moderate (10-30 mph) fast (30-50 mph) very fast (50+ mph)	<action> space: TURN_LEFT TURN_RIGHT SLOW_DOWN BRAKE KEEP_STRAIGHT	<pos> space: left-front right-front left-back right-back front back left right next-to
<dist> space: very close (0-2m) close (2-10m) medium (10-30m) far (30m-)	<duration> space: 0.5 seconds 1.0 seconds 1.5 seconds 2.0 seconds	

Figure 12. **Parameter space summary.** Note that space of `<id*>` is scenario-dependent, namely, all valid labels.

Non-parameterized question generation. These questions don’t have any parameters in their templates, as they demand the VLMs to examine all present objects in observations before answering. An example can be found in Fig. 15. Therefore, no computation is done in the question generation phase.

A.3.2. Answer Generation

A unique query program is selected to generate answers for each type of question. Refer to Fig. 13, Fig. 14, and Fig. 15 for examples. Upon the execution of these query programs, the concrete answers are extracted utilizing scenario information for simulated dynamics. Note that both the question-answer pairs at this stage are not formulated in the multiple-choice setting, and the next stage will reformat the pairs.

A.3.3. Post-processing

At this point, question-answer pairs are already generated. The remaining works are (1) the generation of non-answer candidates for multiple-choice setup (2) the creation of the multiple-choice description strings which map choices with concrete answer candidates (3) the creation of optional “explanation” strings to elevate VLMs’ learning. Each ques-

tion has different search spaces for non-answer candidates. As shown in Fig. 13, `identify_distance`’s candidate space is the `<dist>` space listed in Fig. 12, while that of `embodied_sideness` is a subset of `<pos>` space, shown in Fig. 14. When applicable, non-answer candidates are selected to challenge the evaluated VLMs maximally. For example, candidate generation in question `identify_type` prioritizes ones present in the scenarios on which the question is constructed. After the candidates’ generation, they are put into multiple-choice format as suffixes to the original question, and the answer is replaced by the answer choice. The optional “explanation” strings (used interchangeably with “reasoning”) are also programmatically created, depending on the choice-candidate mapping. Complete implementation will be included in the released codebase.

B. MetaVQA Dataset

B.1. Dataset Composition

Fig. 16 list all question types divided along two dimensions. The horizontal dimension indicates the objects that need to be analyzed to answer the question successfully, and the vertical dimension indicates which facet of embodied scene understanding is evaluated. Detailed descriptions—along with two examples using both simulated and real observations—for each question type can be found at the end of this document in Appendix C.3.2.

B.2. Zero-shot Answerability with Set-of-Mark Prompting

B.2.1. Human Evaluation

Before large-scale dataset generation, we first prepare a small questionnaire to examine the answerability and the quality of the MetaVQA Dataset. Since this is a pilot study, we utilize a Set-of-Mark prompting scheme slightly different from the final MetaVQA Dataset: contours are drawn around objects, and the background color of each label is determined by the corresponding text color following the original paper [11]. We sampled 35 questions with distinct types generated from a single keyframe to speed up the evaluation process. Six participants report an average accuracy of 88.05% on the 35 questions with a standard deviation of 7.54%. The best-performing participant achieves a 94.2% accuracy, while the worst-performing participant reports a 74.2% accuracy. An example question from the questionnaire is illustrated in Fig. 17.

Noticeably, participants struggle with question 19 (5 out of 6 wrong) and question 29 (4 out of 6 wrong), zero-indexed. The former is of type “order_leftmost”, while the latter is of type “describe_distance.” For question 19 illustrated in Fig. 18, the participants report—after questionnaire submission—confusion on whether the answer should

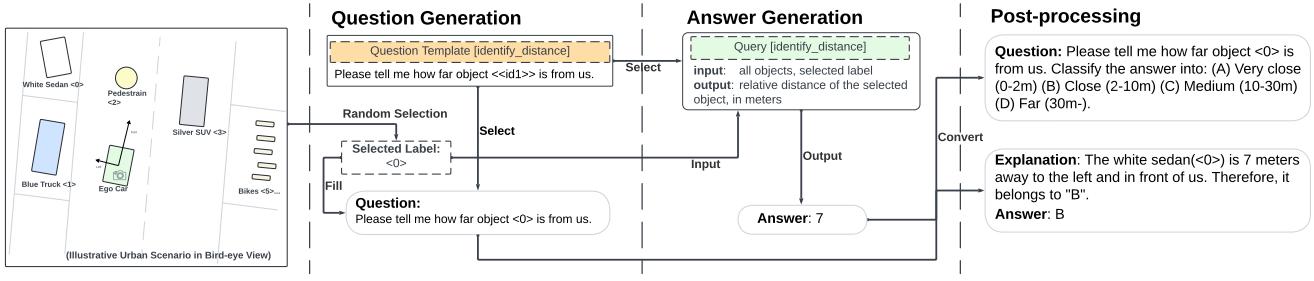


Figure 13. Question-Answer generation of parameterized questions with only one type of parameter.

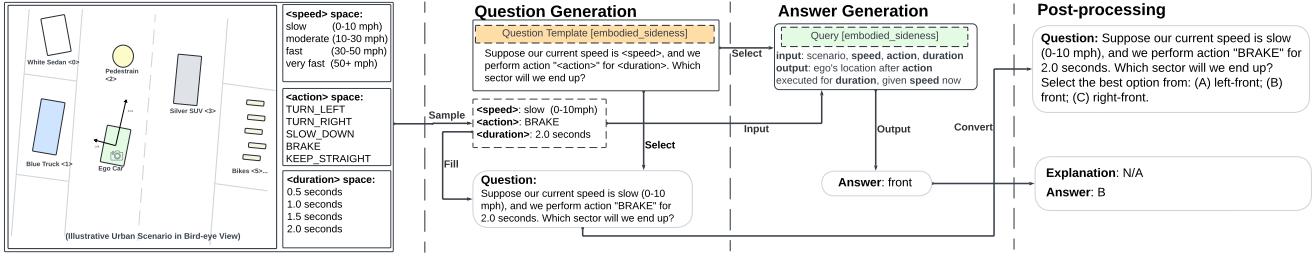


Figure 14. Question-Answer generation of parameterized questions with distinct parameters.

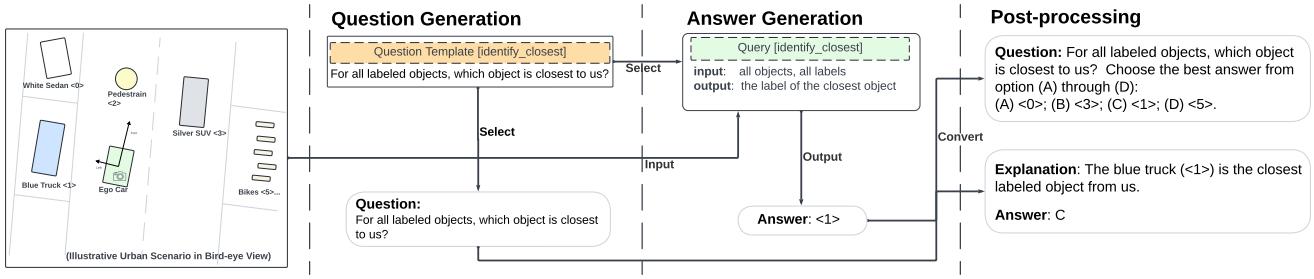


Figure 15. Question-Answer generation of non-parameterized questions.

be deduced using pixel-position ordering of the labels or the world-position ordering of objects. We speculate this confusion leads to the participants’ overwhelming mistakes on this question. In addition, since this question involves objects very distant from the ego vehicle, the question is challenging due to the linear perspective. This might also cause conflicted participants’ responses to question 29 shown in Fig. 19. Accounting for these factors, we refine the generation process for the final version of MetaVQA Dataset by choosing clearer phrasing and enforcing better visibility constraints on objects (for example, increasing the minimum observable pixels). Despite these issues, novice participants still report high test accuracies, and we conclude that the MetaVQA Dataset is intuitive to answer and clear in answering guideline. Therefore, we argue that the MetaVQA Dataset is suitable for zero-shot plug-in-and-play evaluating the embodied scene understanding entertained by general-purpose vision language models.

B.3. Effect of Set-of-Marks Prompting Scheme

The Set-of-Mark [11] paper proposes numerous prompting schemes, from using instance-segmentation masks to bounding boxes. In addition, the text size and background colors are also varied. We perform a grid search with observation generated using different prompting schemes while keeping the base images and object-to-label mapping identical across sets, and we use Qwen2 [17]—the VLM with the best grounding capability as discussed in the main paper. Referring to Tab. 6, Qwen2 achieves the best overall and grounding performance on images annotated with bounding boxes with labels of text size 1.25 and black for background colors. In addition, we observe that text size seems to have a trivial impact on the final performance. Based on these observations, we fixed the annotation style of MetaVQA with bounding-box annotations, black text background color, and a text size of 1.00 (to reduce label occlusions).

Ego Agent	Ego & A Non-ego Agent	A Non-ego Agent	2 Non-ego Agents	N Non-ego Agents	All Non-ego Agents
embodied_distance	embodied_collision				
embodied_sideness	predict_crash_ego_still predict_crash_ego_dynamic				
		identify_distance relative_distance order_closest identify_closest	relative_position order_leftmost order_rightmost order_frontmost order_backmost	identify_heading relative_heading relative_predict_crash_still relative_predict_crash_dynamic pick_closer describe_sector describe_distance	identify_color identify_type identify_leftmost identify_rightmost identify_frontmost identify_backmost describe_scenario
Spatial Questions					
Grounding Question		grounding			

Figure 16. **Question taxonomy** of MetaVQA Dataset. Notice that questions are further blocked by black dotted contours to denote similarity in the formulation (illustrated collectively in Appendix C.3.2).



Figure 17. **Sample question from the questionnaire.** The answer is (C).

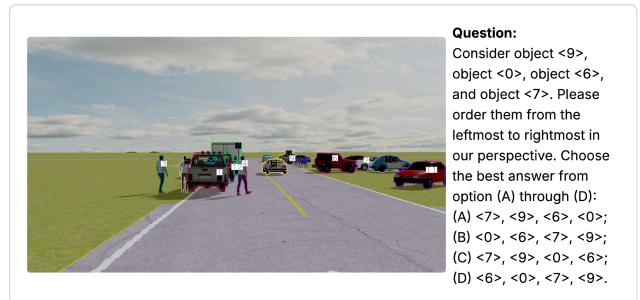


Figure 18. **Question 19 from the questionnaire.** The answer is (A). Ambiguous wording and distant objects lead to common mistakes by participants.

C. Benchmark Results

C.1. Definitions

We present the naming conventions used in this work in this subsection.

C.2. Visual Question Answering

We benchmark the performance of various baselines [15–20] on the withheld test set (“overall”) mentioned in the main paper. Furthermore, We provide detailed performances of baselines on (1) test questions with simulated

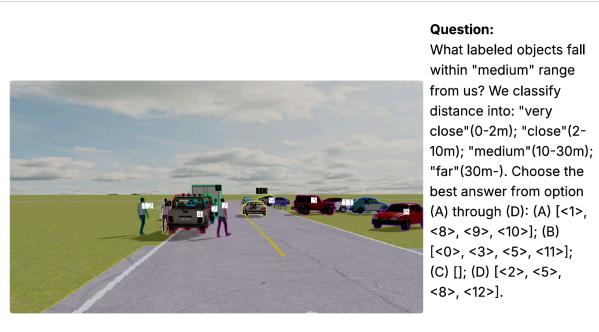


Figure 19. **Question 29 from the questionnaire.** The answer is (D). Some referred objects show limited visibility, leading to common errors.

Text Size	Form	Background	Overall	Grounding
0.75	box	white	0.440	0.867
0.75	box	black	0.457	0.933
0.75	mask	white	0.422	0.467
0.75	mask	black	0.420	0.533
0.75	contour	white	0.430	0.467
0.75	contour	black	0.420	0.733
1.25	box	white	0.437	0.800
1.25	box	black	0.472	0.933
1.25	mask	white	0.440	0.333
1.25	mask	black	0.437	0.333
1.25	contour	white	0.437	0.400
1.25	contour	black	0.422	0.600

Table 6. **Effect of Set-of-Marks Annotations.** We tested different annotation styles, text sizes, and background colors while fixing the model (Qwen2) and the numerical labeling, base images, and grounding questions.

Abbreviation	Checkpoint
LLaVA-NeXT	llava-1.6-vicuna-7b [15]
LLaVA-OneVision	llava-onevision-7b-ov [16]
GPT-4o	GPT-4o [18]
Qwen2	qwen2-vl-7b-instruct [17]
Llama3.2	llama-3.2-11B-Vision-Instruct [19]
InternVL2-4B	InternVL2-4B [50]
InternVL2-8B	InternVL2-8B [50]

Table 7. **Model Abbreviations.** These mappings are used consistently throughout the main paper and the supplementary materials.

observations (“sim” split) (2) test questions with real observations (“real” split). To save spaces, we used abbreviations illustrated in Tab. 7 for baselines in these benchmark tables.

C.2.1. Response Parsing

We establish a unified parsing standard using regular expression (regex) matching for the token sequences generated by all VLMs. If only a singular token is generated, we use this character as the option. If this is not the case, we search for option keywords provided in the multiple-choice questions. In cases of multiple matches, We select the last matched string as the model’s output upon empirical examinations of the VLMs’ raw outputs. If still no match, the parser will look for single characters enclosed by paren-

theses. If all searches returns ill-composed results (empty match or illegal character), we consider the parsing to be a failed case. In the closed-loop evaluations, if a parse failure happens, a randomized action is taken. Code implementation will be available in the Github repository.

C.2.2. Benchmarks on Test Set

Tab. 8 presents the performance in “spatial reasoning” of the baseline VLMs on the withheld test set. Tab. 11 presents the performance in “embodied understanding” on the withheld test set. Tab. 12 presents the grounding performance of baseline VLMs, categorized according to the test set compositions.

C.2.3. Benchmarks on Real Test Split

Tab. 9 presents the performance in “spatial reasoning” of the baseline VLMs on the “real” split of withheld test set. Tab. 13 presents the performance in “embodied understanding” on the “real” split.

C.2.4. Benchmarks on Simulated Test Split

Tab. 10 presents the performance in “spatial reasoning” of the baseline VLMs on the “sim” split of withheld test set. Tab. 14 presents the performance in “embodied understanding” on the “sim” split.

C.3. Closed-loop Evaluation

C.3.1. Task Formulation

Interaction Paradigm. We use the MetaDrive [12] simulator, which provides accurate vehicle dynamics simulation for closed-loop evaluations. VLMs are deployed as driving agents in imported scenarios using [46]. At every five simulation steps (0.5 seconds wall time), the tested VLM is provided with (1) a Set-of-Mark annotated observation captured from the ego’s front camera in 1600×900 resolution and (2) a driving prompt containing current navigation command and allowed discrete action space. The model will analyze the combined input and select the best action from available options. The chosen action will be fed into the simulation, and it will be repeated for the next 0.5 seconds (5 steps in simulation time) until the next inference step. Fig. 20 illustrates this process. The simulations terminate when their time horizons are reached or when the ego vehicle wanders off drivable regions.

Very rarely, the tested VLM will generate an invalid response according to the parser mentioned in Appendix C.2.1. In this situation, we fix the chosen action as “KEEP_STRAIGHT” such that the speed and the heading of the ego vehicle will remain roughly identical.

Navigation command. At each inference step, the navigation command is recomputed to adjust for the current position of the ego vehicle. The command follows the following form:

Model	Embodied Questions (Overall)					
	Overall	embodied_distance	embodied_collision	predCrash ego_still	embodied_sideness	predCrash ego_dynamic
random	0.382	0.255	0.498	0.521	0.348	0.504
LLaVA-NeXT	0.419	0.159	0.489	0.303	0.652	0.384
LLaVA-OneVision	0.746	0.442	0.923	0.976	0.794	0.961
GPT-4o	0.764	0.785	0.719	0.893	0.732	0.873
Qwen2	0.649	0.451	0.836	0.259	0.804	0.482
Qwen2-finetuned	0.948	0.998	0.879	0.817	1.000	0.894
Llama3.2	0.536	0.332	0.650	0.517	0.574	0.849
Llama3.2-finetuned	0.944	0.962	0.846	0.997	0.999	0.961
InternVL2-8B	0.711	0.620	0.923	0.914	0.509	0.961
InternVL2-8B-finetuned	0.926	0.807	0.953	0.997	1.000	0.961

Table 11. **VQA benchmarks (Overall-Embodied).** Per-question accuracies are evaluated on the withheld test set.

Model	Embodied Questions (Real)					
	Overall	Emb.Dist	Emb.Coll	PredCrash EgoStill	Emb.Side	PredCrash EgoDyn
random	0.372	0.248	0.498	0.487	0.345	0.467
LLaVA-NeXT	0.414	0.189	0.445	0.342	0.647	0.327
LLaVA-OneVision	0.735	0.430	0.905	0.980	0.795	0.980
GPT-4o	0.762	0.784	0.706	0.895	0.739	0.873
Qwen2	0.653	0.446	0.852	0.322	0.796	0.453
Qwen2-finetuned	0.946	0.999	0.875	0.789	1.000	0.887
Llama3.2	0.542	0.339	0.654	0.566	0.580	0.867
Llama3.2-finetuned	0.947	0.969	0.844	0.993	0.999	0.980
InternVL2-8B	0.720	0.615	0.905	0.895	0.576	0.980
InternVL2-8B-finetuned	0.919	0.780	0.956	0.993	1.000	0.980

Table 13. **VQA benchmarks (Real-Embodied).** Per-question accuracies are evaluated on the “real” split of the withheld test set. Question types are shortened for formatting.

wherein S_{max} (degree) is the maximal steering angle, F_{max} (hp) is the maximal engine force, and B_{max} (hp) is the maximal brake force. For fair and replicable experiments, we use identical vehicle configurations(for example, maximum engine force) across different trials.

We conducted grid searches to fix the suitable set of actions. For each candidate, we reconstruct real-world driving trajectories as action sequences with only allowed action provided by the candidate. These sequences are computed greedily (and repeated) at every five simulation steps, following the same inference frequency as the closed-loop evaluation. The optimal action at a particular step is decided according to the resulting deviation from the original trajectories if the action is executed. This sequence-building is autoregressive, meaning that previous optimal actions(and their generated trajectories) affect the decision on later optimal actions. We fix the current action space as it leads to the best reconstruction quality.

Test scenarios. We tailor 120 diverse scenarios to evaluate VLMs’ embodied scene understanding holistically.

Model	Grounding Questions		
	Overall	Real	Sim
random	0.268	0.257	0.280
LLaVA-NeXT	0.248	0.229	0.271
LLaVA-OneVision	0.728	0.827	0.615
GPT4-o	0.831	0.888	0.766
Qwen2	0.874	0.859	0.890
Qwen2-finetuned	0.972	0.992	0.950
Llama3.2	0.790	0.855	0.716
Llama3.2-finetuned	0.923	0.944	0.899
InternVL2-8B	0.702	0.783	0.610
InternVL2-8B-finetuned	0.916	0.948	0.881

Table 12. **VQA benchmarks (Grounding).** Per-question accuracies are evaluated on the withheld whole test set, “real” split of the test set, and “sim” split of the test set.

Model	Embodied Questions (Sim)					
	Overall	Emb.Dist	Emb.Coll	PredCrash EgoStill	Emb.Side	PredCrash EgoDyn
random	0.395	0.264	0.497	0.558	0.353	0.545
LLaVA-NeXT	0.426	0.12	0.542	0.261	0.660	0.448
LLaVA-OneVision	0.760	0.457	0.945	0.971	0.793	0.940
GPT-4o	0.767	0.786	0.734	0.891	0.722	0.873
Qwen2	0.645	0.457	0.817	0.188	0.815	0.515
Qwen2-finetuned	0.949	0.998	0.883	0.848	1.000	0.903
Llama3.2	0.527	0.321	0.644	0.464	0.565	0.828
Llama3.2-finetuned	0.939	0.952	0.847	1.000	1.000	0.940
InternVL2-8B	0.699	0.627	0.945	0.935	0.418	0.940
InternVL2-8B-finetuned	0.936	0.843	0.949	1.000	1.000	0.940

Table 14. **VQA benchmarks (Sim-Embodied).** Per-question accuracies are evaluated on the “sim” split of the withheld test set. Question types shortened for formatting.

These scenarios include 60 from the nuScenes dataset and the other 60 selected from a corpus of safety-critical situations generated using CAT [51]. For each of the 60 safety-critical scenarios, an adversarial agent will attempt to run into the ego vehicle, and we ensure the observability of adversarial agents.

C.3.2. Metrics

Route Completion The ratio of the traveled distance against the length of the complete route averaged across scenarios.

Collision Rate The ratio of scenarios where the ego vehicle collides with any other object.

Off-Road Rate The ratio of scenarios where the ego vehicle leaves drivable regions.

Final Displacement Error (FDE) The L2 distance between the final position of the ego vehicle from the final destination averaged across scenarios.

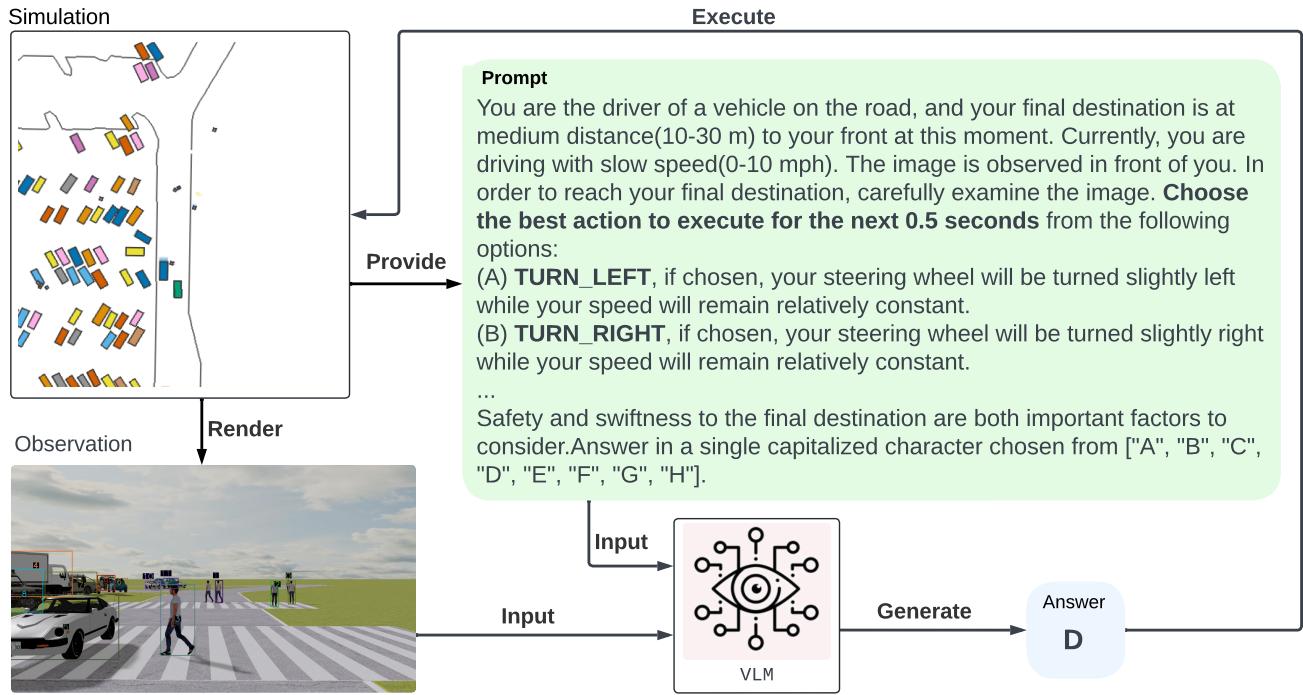


Figure 20. **Closed-loop evaluation paradigm.**

Average Displacement Error (ADE) The mean per-step L2 distance between the ground-truth trajectories and the VLM-driven trajectories averaged across scenarios. If a simulation terminates prematurely(due to VLMs driving off-road), the last ego vehicle position is appended to align the length of the ground-truth trajectory and the VLM-driven trajectory.

Embodied Questions:

embodied_distance. This question examines how far the ego will move from the current position, assuming that <action> is executed over the next <duration> period and the ego's current speed is <speed>.



Question:

Suppose our current speed is fast(30-50 mph), and we perform action "BRAKE" for 1.5 seconds. How far will we end up from our current position? Select the best option from: (A) Very close(0-2m); (B) Close(2-10m); (C) Medium(10-30m); (D) Far(30m-)

Explanation: N/A

Answer: C

Question:

Suppose our current speed is moderate(10-30 mph), and we perform action "SLOW_DOWN" for 1.0 seconds. How far will we end up from our current position? Select the best option from:(A) Very close(0-2m); (B) Close(2-10m); (C) Medium(10-30m); (D) Far(30m-)

Explanation: N/A

Answer: B

embodied_sideness. This question examines how whether the ego will move to its left or its right(in the current frame), assuming that <action> is executed over the next<duration> period and the ego's current speed is <speed>.



Question:

Suppose our current speed is moderate(10-30 mph), and we perform action "KEEP_STRAIGHT" for 1.0 seconds. Which sector will we end up? Select the best option from: (A) left-front; (B) front; (C) right-front.

Explanation: N/A

Answer: B

Question:

Suppose our current speed is moderate(10-30 mph), and we perform action "TURN_LEFT" for 1.5 seconds. Which sector will we end up? Select the best option from: (A) left-front; (B) front; (C) right-front.

Explanation: N/A

Answer: A

Embodied Questions:

embodied_collision. This question examines whether the ego will collide into selected object <id1>, assuming that <action> is executed over the next <duration> period and the ego's current speed is <speed>.



Question:

Suppose our current speed is slow(0-10 mph), and we perform action "BRAKE" for 0.5 seconds. Will we run into object <0>, provided that it remains still? Select the best option from: (A) Yes; (B) No.

Explanation:

We will not run into object <0>, even though we both end in our front sector.

Answer: B



Question:

Suppose our current speed is fast(30-50 mph), and we perform action "SLOW_DOWN" for 1.0 seconds. Will we run into object <0>, provided that it remains still? Select the best option from: (A) Yes; (B) No.

Explanation:

We will not run into object <0>. Object <0> is located in the right-front sector, but we will end in the front sector.

Answer: B

predict_crash_ego_*. This family of questions examines how whether the selected object <id1> will collide with the ego under various conditions.

predict_crash_ego_still



Question:

Suppose object <0> proceed along its current heading. Will it collides into us if we stay still? Choose the best answer between option (A) and (B): (A) Yes; (B) No.

Explanation:

No, this bus(<0>) to the right and in front of us and heading toward our front direction will not run into us if it drives along its current heading.

Answer: B

predict_crash_ego_dynamic



Question:

Suppose both object <5> and us proceed along our corresponding current headings with the same speed. Will we collide into each other? Choose the best answer between option (A) and (B): (A) Yes; (B) No.

Explanation:

No, this gray pickup(<5>) directly in front of us and heading toward our front direction will not run into us.

Answer: B

Spatial Questions:

identify_distance. This question prompts VLMs to estimate the distance of the selected object <id1> from the ego.



Question:

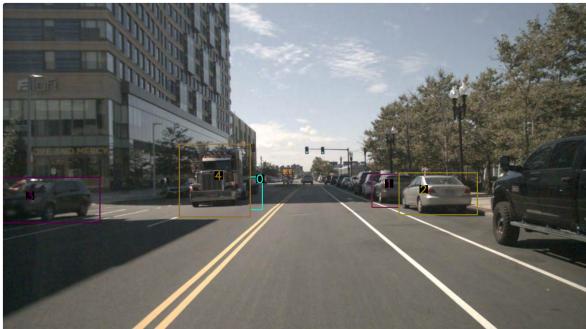
Please tell me how far object <0> is from us. Classify the answer into: (A) Very close(0-2m) (B) Close(2-10m) (C) Medium(10-30m) (D) Far(30m-).

Explanation:

The truck(<0>) is 28 meters to the left and in front of us. Therefore, it belongs to "medium".

Answer: C

identify_position. This question prompts VLMs to estimate the direction of the selected object <id1> from the ego.



Question:

Please tell me the relative position of <3> with respect to us. Choose the best answer from option (A) through (D): (A) next-to; (B) back; (C) left-front; (D) right-front.

Explanation:

The car(<3>) is to the left and in front of us.

Answer: C

Question:

Please tell me the relative position of <8> with respect to us. Choose the best answer from option (A) through (D): (A) right-front; (B) front; (C) next-to; (D) left.

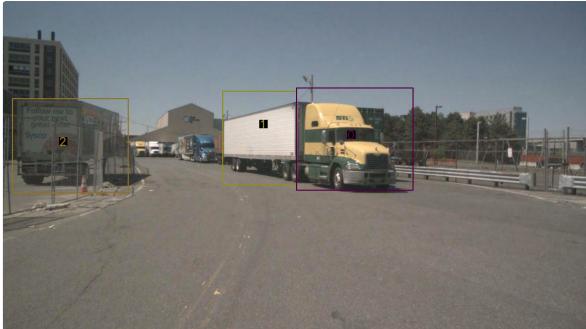
Explanation:

The white sedan(<8>) is directly in front of us.

Answer: B

Spatial Questions:

identify_heading. This question prompts models to estimate the heading angle of the selected object `<id1>`, expressed relative to the ego's front direction. The provided options are sufficiently distinct to avoid ambiguity.



Question:

Please describe the heading direction of object `<0>`. Choose the best answer from option (A) through (D): (A) left-front; (B) right-front; (C) right-back; (D) left-back.

Explanation:

The truck(`<0>`) directly in front of us is facing our right-back direction.

Answer: C



Question:

Please describe the heading direction of object `<1>`. Choose the best answer from option (A) through (D): (A) front; (B) right; (C) left; (D) back.

Explanation:

The red sports car(`<1>`) to the left and in front of us is facing our right direction.

Answer: B

identify_color This question prompts models to select the color of object `<id1>`. Note that it is generated only with simulated observations, as “color” is not annotated in the nuScenes dataset.



Question:

Specify the color of object `<1>`. Choose the best answer from option (A) through (D): (A) White; (B) Black; (C) Grey; (D) Yellow.

Explanation:

The color of this hatchback (`<1>`) to the right and in front of us is yellow.

Answer: D



Question:

Specify the color of object `<1>`. Choose the best answer from option (A) through (D): (A) White; (B) Black; (C) Grey; (D) Yellow.

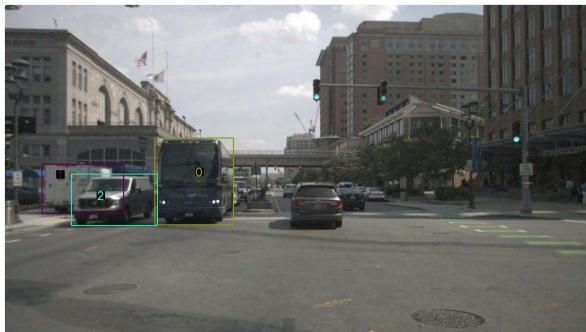
Explanation:

The color of this hatchback (`<1>`) to the right and in front of us is yellow.

Answer: D

Spatial Questions:

identify_type. This question prompts VLMs to select the most descriptive type of the selected object <id1>.



Question:

Specify the type of object <0>. Choose the best answer from option (A) through (D): (A) Car; (B) Hatchback; (C) Truck; (D) Bus.

Explanation:

The type of this object(<0>) to the left and in front of us is "bus".

Answer: D



Question:

Specify the type of object <4>. Choose the best answer from option (A) through (D): (A) Hatchback; (B) Sedan; (C) Suv; (D) Sports Car.

Explanation:

The type of this white object(<4>) to the left and in front of us is "sedan".

Answer: B

relative_distance. This question prompts VLMs to select the relative distance between two objects <id1> and <id2>.



Question:

How close are object <1> and object <3> positioned? Classify the answer into: (A) Very close(0-2m); (B) Close(2-10m); (C) Medium(10-30m); (D) Far(30m-).

Explanation:

Object <1>, a car to the left and in front of us, is directly in front of object <3>, a car to the left and in front of us, at a close distance.

Answer: B



Question:

How close are object <7> and object <0> positioned? Classify the answer into: (A) Very close(0-2m); (B) Close(2-10m); (C) Medium(10-30m); (D) Far(30m-).

Explanation:

Object <7>, a gray pickup to the right and in front of us, is to the right of object <0>, a yellow hatchback to the left and in front of us, at a medium distance.

Answer: C

Spatial Questions:

relative position. This question prompts VLMs to evaluate how object <id1> related spatially with object <id12>, expressed in the ego perspective.



Question:

How is object <6> positioned relative to object <2>? Choose the best answer from option (A) through (D): (A) next-to; (B) right-back; (C) right; (D) left-front.

Explanation:

Object <6>, a traffic barrier to the right and in front of us, is to the right of object <2>, a traffic barrier to the right and in front of us.

Answer: C



Question:

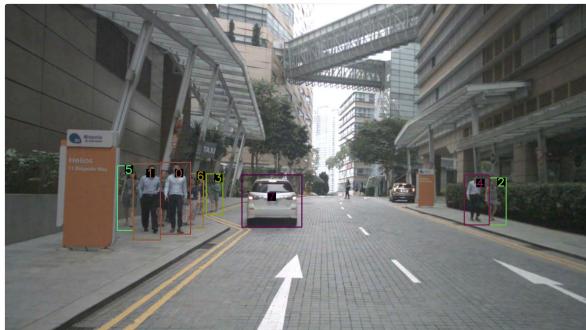
How is object <4> positioned relative to object <7>? Choose the best answer from option (A) through (D): (A) right-back; (B) right-front; (C) next-to; (D) left-back.

Explanation:

Object <4>, a blue hatchback to the right and in front of us, is to the right and in front of object <7>, a gray sedan directly in front of us.

Answer: B

relative heading. This question prompts VLMs to determine if object <id1> and <id12> are heading towards roughly the same direction.



Question:

Is object <7> heading toward roughly the same direction as object <1>? Choose the best answer between option (A) and (B): (A) Yes; (B) No.

Explanation:

No. Object <7>, a car directly in front of us, is not heading toward the same direction as object <1>, a pedestrian located to the left and in front of us. In particular, object <1>'s heading differs by 194 degrees counterclockwise from that of object <7>.

Answer: B



Question:

Is object <2> heading toward roughly the same direction as object <1>? Choose the best answer between option (A) and (B): (A) Yes; (B) No.

Explanation:

No. Object <2>, a blue hatchback to the right and in front of us, is not heading toward the same direction as object <1>, a white bike located directly in front of us. In particular, object <1>'s heading differs by -81 degrees counterclockwise from that of object <2>.

Answer: B

Spatial Questions:

relative_predict_crash_*. This family of questions prompts VLMs to infer whether two objects <id1> and <id2> will collide under varying assumptions.

relative_predict_crash_dynamic



Question:

Suppose object <0> and object <1> proceed along their current directions with the same speed. Will they collide into each other? Choose the best answer between option (A) and (B): (A) No; (B) Yes.

Explanation:

No, object <0> will not run into object <1>.

Answer: A

relative_predict_crash_still



Question:

Suppose object <1> proceed along its current heading. Will it collide into object <0> if object <0> stays still? Choose the best answer between option (A) and (B): (A) No; (B) Yes.

Explanation:

Yes, object <1> will run into object <0>.

Answer: B

pick_closer. This question asks the VLM to select the closer object from two candidates.



Question:

Which object is closer to me, <4> or <3>? Choose the best answer from option (A) through (C): (A) <4> is closer; (B) <3> is closer; (C) <4> and <3> are about the same distance.

Explanation:

Object <3>, a truck to the right and in front of us, is closer to us than object <4>, a traffic barrier directly in front of us.

Answer: C



Question:

Which object is closer to me, <2> or <11>? Choose the best answer from option (A) through (C): (A) <2> is closer; (B) <11> is closer; (C) <2> and <11> are about the same distance.

Explanation:

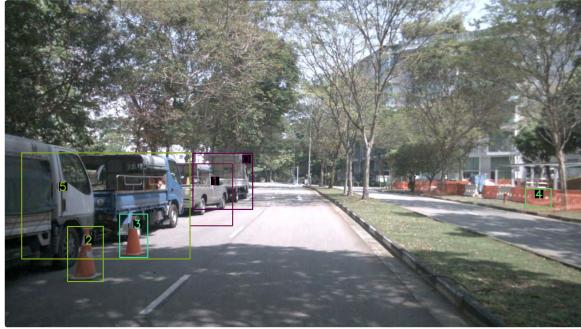
Object <11>, a yellow sports car to the right of us, is closer to us than object <2>, a white sedan to the left and in front of us.

Answer: C

Spatial Questions:

order_star. This family of questions asks the VLM to attend to multiple objects and sort their relevance by some spatial ordering in top-down world coordinates.

order_leftmost



Question:

Consider object <5>, object <2>, object <3>, and object <0>. Please order them from leftmost to rightmost in our coordinate system. Choose the best answer from option (A) through (D):
 (A) <3>, <0>, <2>, <5>
 (B) <5>, <3>, <2>, <0>
 (C) <0>, <5>, <3>, <2>
 (D) <2>, <3>, <5>, <0>

Explanation:

The truck(<0>) is at the far left, and the traffic cone(<2>) is at the far right. The truck(<5>) and the traffic cone(<3>) are in between.

Answer: C

order_frontmost



Question:

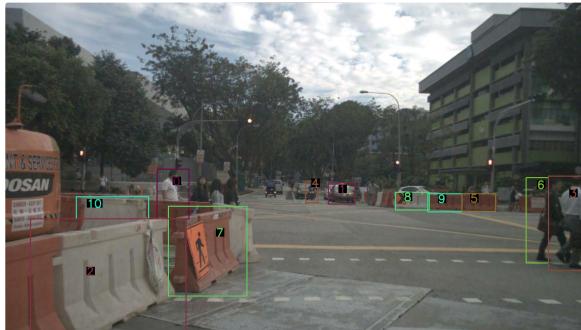
Consider object <0>, object <2>, object <5>, and object <4>. Please order them from furthest to the closest along our front direction. Choose the best answer from option (A) through (D):
 (A) <5>, <0>, <2>, <4>
 (B) <2>, <4>, <0>, <5>
 (C) <4>, <5>, <0>, <2>
 (D) <2>, <5>, <0>, <4>

Explanation:

The white sedan(<5>) is at the furthest along our heading direction, and the black sedan(<4>) is the closest. The yellow hatchback(<0>) and the white sedan(<2>) are in between.

Answer: A

describe_sector. This question asks the VLM to attend to all observable objects and select the maximal object set such that all of its members are in the specified ego's direction from the question body.



Question:

What labeled objects fall into our left-front sector? Choose the best answer from option (A) through (D):
 (A) []
 (B) [<0>, <2>, <7>, <10>]
 (C) [<1>, <3>, <7>, <10>]
 (D) [<3>, <5>, <7>, <8>]

Explanation:

Option (A) is wrong since there exists at least 4 objects(<0>, <2>, <7>, <10>) in the specified(left-front) sector;
 Option (C) is wrong since there exists 2 objects(<1>, <3>) in the right-front sector;
 Option (D) is wrong since there exists 3 objects(<3>, <5>, <8>) in the right-front sector.
Answer: B



Question:

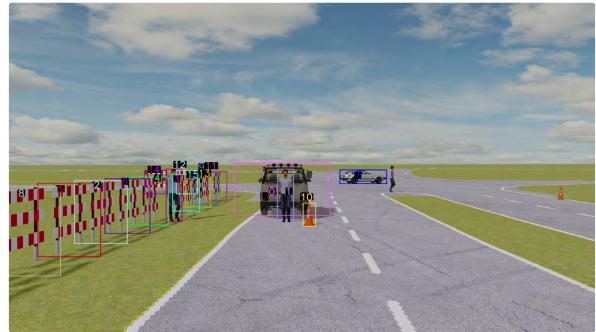
What labeled objects fall into our front sector? Choose the best answer from option (A) through (D):
 (A) []
 (B) [<3>, <4>, <8>]
 (C) [<3>, <4>, <9>]
 (D) [<5>, <9>, <10>]

Explanation:

Option (A) is wrong since there exists at least 3 objects(<5>, <9>, <10>) in the specified(front) sector;
 Option (B) is wrong since there exists 3 objects(<3>, <4>, <8>) in the left-front sector;
 Option (C) is wrong since there exists 2 objects(<3>, <4>) in the left-front sector.
Answer: D

Spatial Questions:

describe_distance. This question asks VLMs to attend to all observable objects and select the maximal object set such that all of its members are located away from the ego by the specified distance from the question body.



Question:

What labeled objects fall within "medium" range from us? We classify distance into: "very close"(0-2m); "close"(2-10m); "medium"(10-30m); "far"(30m-). Choose the best answer from option (A) through (D):
(A) [<0>, <1>, <2>]
(B) [<1>, <2>, <3>]
(C) []
(D) [<0>, <2>, <3>]

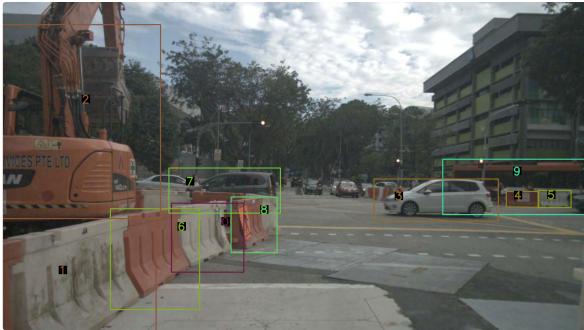
Explanation:

Option (A) is wrong since there exists 1 object(<1>) positioned at far distance from us; Option (B) is wrong since there exists 1 object(<1>) positioned at far distance from us; Option (C) is wrong since there exists at least 3 objects(<0>, <2>, <3>) positioned at specified(medium) distance from us.

Answer: D

identify_star. This question asks the VLM to attend to all observable objects and select the leading object according to some ordering specified in the question body.

identify_frontmost



Question:

For all labeled objects, which object is furthest along our front direction? Choose the best answer from option (A) through (D): (A) <3>; (B) <4>; (C) <5>; (D) <9>.

Explanation:

The bus(<9>) is the object furthest along front direction us.
Answer: D

Question:

What labeled objects fall within "medium" range from us? We classify distance into: "very close"(0-2m); "close"(2-10m); "medium"(10-30m); "far"(30m-). Choose the best answer from option (A) through (D):
(A) [<1>, <5>, <10>, <15>]
(B) []
(C) [<0>, <8>, <11>, <15>]
(D) [<5>, <9>, <13>, <14>]

Explanation:

Option (B) is wrong since there exists at least 4 objects(<1>, <5>, <10>, <15>) positioned at specified(medium) distance from us; Option (C) is wrong since there exists 1 object(<11>) positioned at far distance from us; Option (D) is wrong since there exists 1 object(<14>) positioned at far distance from us.

Answer: A

identify_closest



Question:

For all labeled objects, which object is closest to us? Choose the best answer from option (A) through (D): (A) <1>; (B) <0>; (C) <9>; (D) <3>.

Explanation:

The gray pickup(<1>) is the closest labeled object from us.
Answer: A

Spatial Questions:

describe_scenario. This question prompts the VLM to examine all labeled objects in the scenario. It is a train-only question designed to boost learning performance and avoid VLM collapse.



Question:

Please describe all labeled objects in the observation, in the order of their labels...

Explanation:

- <0>: A car positioned in our front sector at medium distance. It heads toward our front direction.
- <1>: A pedestrian positioned in our right-front sector at medium distance. It heads toward our back direction.
- <2>: A car positioned in our right-front sector at far distance. It heads toward our front direction.
- <3>: A pedestrian positioned in our left-front sector at medium distance. It heads toward our right-back direction.
- <4>: A pedestrian positioned in our left-front sector at medium distance. It heads toward our right-back direction.
- <5>: A pedestrian positioned in our left-front sector at medium distance. It heads toward our back direction.
- <6>: A pedestrian positioned in our left-front sector at medium distance. It heads toward our right-back direction.
- <7>: A pedestrian positioned in our left-front sector at medium distance. It heads toward our back direction.
- <8>: A pedestrian positioned in our right-front sector at medium distance. It heads toward our back direction.
- <9>: A pedestrian positioned in our left-front sector at medium distance. It heads toward our right-back direction.

Answer: N/A



Question:

Please describe all labeled objects in the observation, in the order of their labels...

Explanation:

- <0>: A yellow sports car positioned in our left-front sector at medium distance. It heads toward our front direction.
- <1>: A blue SUV positioned in our right-front sector at medium distance. It heads toward our front direction.
- <2>: A white sedan positioned in our front sector at far distance. It heads toward our front direction.
- <3>: A gray pickup positioned in our right-front sector at medium distance. It heads toward our front direction.
- <4>: A blue SUV positioned in our left-front sector at far distance. It heads toward our front direction.

Answer: N/A

Grounding Questions:

grounding. This question examines the visual grounding ability of the tested VLM. All non-answer options are selected from valid labels to challenge the model maximally.



Question:

What is the numerical label associated with the highlighted area?
Choose the best answer from option (A) through (D): (A) 1; (B) 26; (C) 30; (D) 28.

Explanation: N/A

Answer: B



Question:

What is the numerical label associated with the highlighted area?
Choose the best answer from option (A) through (D): (A) 10; (B) 25; (C) 35; (D) 29.

Explanation: N/A

Answer: D