ANCHORREASONING: A VISUAL GROUNDING AND CAUSAL REASONING DATASET IN LONG-TAIL AUTONOMOUS DRIVING SCENARIOS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) offer a promising approach to long-tail autonomous driving, but existing driving datasets provide limited supervision for connecting decision-critical visual evidence with reasoning and planning. We introduce AnchorReasoning, a visually grounded reasoning dataset built on WOD-E2E, containing 416,119 annotated frames and 395,379 decision-critical elements across four major categories and 19 fine-grained types. Each frame is organized as a visually grounded chain-of-thought (VG-CoT) that links decision-critical element identification and localization, element attributes and implications, driving-action rationale, and action and trajectory planning. We further develop a curriculum supervised fine-tuning strategy that progressively learns these hierarchical capabilities, together with an object-size-aware grounding metric for evaluating localization quality. Experiments across eight general-purpose, embodied-AI, and AV-specific backbones show that VG-CoT supervision improves grounded reasoning and trajectory prediction. Across models, 5-s ADE and FDE decrease by 71.4% and 60.6%, while RFS Frame and Cluster improve by 27.0% and 27.5%. These gains are achieved with 18.5 fewer reasoning tokens and 0.32 s/frame lower inference latency on average, demonstrating the value of visually grounded, decision-focused supervision for VLM reasoning and planning in long-tail autonomous driving. The code is available at `https://anonymous.4open.science/r/AnchorReasoning-6266`.

1 INTRODUCTION

Recent autonomous driving research has increasingly focused on addressing long-tail scenarios. These rare and complex situations often require a deeper understanding of the driving context and deliberate reasoning to derive appropriate behavioral plans. Vision-language models (VLMs), with their strong multimodal understanding and commonsense reasoning capabilities, have therefore emerged as a promising paradigm for handling such challenging scenarios (Wang et al., 2025; Zhou et al., 2026). However, effectively harnessing these capabilities for long-tail driving largely depends on how the training data are organized.

Early VLM-oriented driving datasets were primarily organized around scene descriptions or captioning annotations (Kim et al., 2018; Liu et al., 2023). Such annotations provide holistic supervision of the driving environment, enabling models to learn high-level scene semantics and contextual information. However, coarse scene-level descriptions provide limited supervision for fine-grained, object-centric understanding. More importantly, they provide little explicit guidance on how scene observations should be translated into driving decisions.

To provide more targeted supervision, recent datasets have increasingly adopted question-answering (QA) and chain-of-thought (CoT) annotations (Wu et al., 2025; Marcu et al., 2024; Qian et al., 2024). QA-based datasets decompose driving scenes into object-level questions, allowing models to learn individual visual concepts. Such supervision enables more fine-grained probing and learning of driving implications of individual elements. However, QA-based supervision primarily trains VLMs to answer isolated questions, providing limited guidance on how multiple pieces of evidence are connected into a coherent and structured reasoning process (Nie et al., 2024). CoT-based an-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

notations mitigate this limitation by modeling the intermediate reasoning steps, enabling VLMs to learn a coherent driving reasoning chain that connects scene observation, contextual interpretation, and decision-making (Ishaq et al., 2025; Arai et al., 2025). Despite these advances, most existing CoT supervision does not prioritize decision-critical evidence. Consequently, VLMs may incorporate decision-irrelevant elements into their reasoning, resulting in verbose or distracted outputs that compromise both quality and efficiency (Qian et al., 2025). In addition, current CoT annotations are predominantly textual and rarely ground critical elements to their corresponding visual regions. This missing connection may cause VLMs to rely on language priors rather than the actual scene, producing interpretations that are weakly supported by visual observations and increasing the risk of hallucination (Xie et al., 2025; Chang et al., 2026). Moreover, most existing datasets are dominated by routine driving scenarios and provide limited coverage of long-tail situations.

To address these limitations, we introduce AnchorReasoning, a visual grounding and causal reasoning dataset for VLMs in long-tail driving scenarios (Table 1). Built upon WOD-E2E (Xu et al., 2026), AnchorReasoning augments real-world long-tail driving clips with decision-critical visual grounding annotations and driving reasoning chains tailored for VLM training. For each frame, the annotations are organized as a visually grounded chain-of-thought (VG-CoT): observation and grounding, element implication, driving action rationale, and driving action and trajectory planning. First, it guides VLMs to identify and localize decision-critical elements, such as roadway users and traffic control devices. It then enables VLMs to interpret element attributes and infer their implications for ego driving. VLMs then consolidate multiple implications to derive the action plan and future trajectory, while providing a rationale. In addition, we develop a curriculum supervised fine-tuning (SFT) strategy to fully leverage the proposed dataset. The curriculum progressively equips VLMs with visually grounded causal reasoning and planning capabilities through the VG-CoT supervision. We further design a point-based visual-grounding metric to evaluate the localization accuracy of predicted decision-critical elements. Experimental results across different backbones demonstrate that VG-CoT supervision equips models with visual grounding capability, enabling them to incorporate visual evidence into higher-quality reasoning and more accurate trajectory prediction. It improves the Rater Feedback Score (RFS) Frame and Cluster by 27.0% and 27.5%. These gains are achieved with 18.5 fewer reasoning tokens on average and a 0.32 s/frame reduction in inference latency. The main contributions of this paper are summarized as follows:

- We introduce AnchorReasoning, a long-tail driving dataset for VLM training that provides decision-relevant visual grounding, fine-grained element attributes and semantics, and structured reasoning annotations. These annotations enable models to identify, localize, and correctly interpret the elements that truly influence driving decisions, thereby improving their trajectory planning performance in long-tail autonomous driving scenarios.
- We propose a curriculum SFT strategy that follows the structure of VG-CoT, allowing models to fully leverage the hierarchical supervision and progressively acquire visual grounding, element implication, driving action rationale, behavior planning, and trajectory generation capabilities.

Table 1: **Comparison of representative VLM-oriented driving datasets.**

Dataset	# Ann. Frames	Long-tail	Decision-critical Element	Decision-critical Elements: Visual Grounding	Decision-critical Elements: Impact Rank	Decision-critical Elements: Implication	Reasoning	Action / Trajectory
nuScenes-QA (Qian et al., 2024)	34,149	✗	✗	✗	✗	✗	QA	–
LingoQA (Marcu et al., 2024)	28K clips	✗	✓	✗	✗	✗	QA	Action
Rank2Tell (Sachdeva et al., 2024)	3,833	✗	✓	✓	✓	✗	QA	–
DriveLM (Sima et al., 2024)	4,871	✗	✓	✗	✓	✓	Graph QA	Both
DriveLMM-o1 (Ishaq et al., 2025)	2,501	✗	✓	✗	✗	✗	CoT	Action
STRIDE-QA (Ishihara et al., 2026)	270K	✗	✗	✗	✗	✗	Grounded QA	–
CoVLA (Arai et al., 2025)	6.0M	✗	✗	✗	✗	✗	Description	Both
nuReasoning (Huang et al., 2026)	∼3.78M est.	✓	✓	△	✗	✓	QA & CoT	Both
AnchorReasoning (Ours)	**416K**	✓	✓	✓	✓	✓	**VG-CoT**	**Both**

Note. "Graph QA" organizes QA pairs as a dependency graph linking perception, prediction, and planning questions. "Grounded QA" associates QA pairs with corresponding visual regions or objects. △ denotes partial grounding: nuReasoning grounds objects in spatial reasoning with 2D image locations, but does not explicitly propagate this grounding into decision reasoning or trajectory generation.

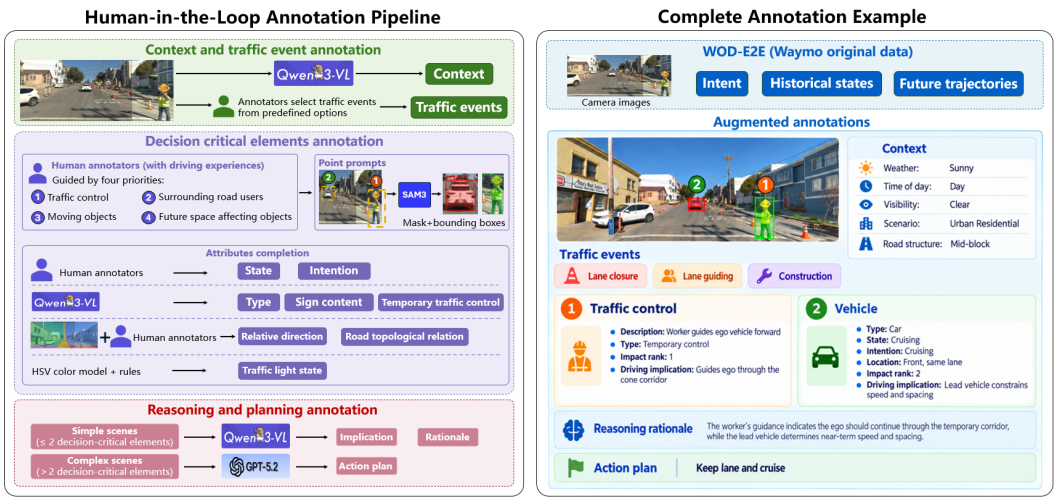


Figure 1: AnchorReasoning annotation pipeline, and example.

- We introduce an object-size-aware visual grounding metric that evaluates predicted grounding points using scale-adaptive point-to-mask distances, providing a consistent measure of localization quality across elements of different sizes.

2 Related Work

Language Description and Explanation. Early language-supervised driving datasets focus on describing driving behavior and explaining action decisions. BDD-X provides natural-language descriptions of driving actions and their justifications (Kim et al., 2018), while DRAMA further introduces localization of risk-relevant objects together with hazard explanations (Malla et al., 2023). These works establish language-based supervision for driving scene understanding and behavior explanation.

Question Answering. VQA-based datasets provide more structured supervision through targeted questions about driving scenes. NuScenes-QA covers object existence, attributes, states, and spatial relationships (Qian et al., 2024); LingoQA extends the scope to localization, anticipation, attention, and action justification (Marcu et al., 2024); and Rank2Tell identifies and ranks decision-relevant traffic participants with natural-language explanations (Sachdeva et al., 2024). DriveLM further introduces Graph VQA, connecting perception, prediction, and planning questions through logical dependencies (Sima et al., 2024). These datasets support fine-grained learning of object- and task-level driving knowledge.

Chain-of-Thought Reasoning. Recent datasets move toward structured reasoning chains for driving. Reason2Drive introduces reasoning across perception, prediction, and decision-making (Nie et al., 2024), while DriveLMM-o1 provides step-by-step rationales for perception, prediction, and planning (Ishaq et al., 2025). nuReasoning further extends reasoning supervision to real-world long-tail scenarios with spatial, decision, and counterfactual reasoning (Huang et al., 2026).

3 AnchorReasoning Dataset

3.1 Overview

AnchorReasoning is a visually grounded reasoning dataset designed to train and evaluate VLM in long-tail scenarios. Built upon WOD-E2E, it augments real-world long-tail driving clips with annotations centered on decision-critical elements. AnchorReasoning identifies and visually grounds the



Figure 2: Representative annotation examples across diverse driving scenarios. Decision-critical elements are highlighted with bounding boxes and category-specific labels.

elements that influence the ego driving, and further annotates their attributes and implications. These annotations are organized into a VG-CoT that connects grounding, scene understanding, causal reasoning, action and trajectory prediction. The dataset contains over **11** hours of driving data across **2492** clips, comprising **416,119** annotated frames and **395,379** decision-critical elements.

3.2 VG-CoT Annotation Schema

VG-CoT is designed to guide VLMs through a hierarchical and evidence-grounded driving reasoning process. It encourages models to reason from the observed scene instead of relying primarily on language priors, thereby improving reasoning fidelity and reducing hallucinations. VG-CoT first provides scene-level context and event-level traffic information. The context annotations describe global driving conditions (Fig. 3), providing the background needed to interpret appropriate driving behavior under different environments. Traffic event annotations further capture events such as roadside parking and construction. These annotations provide event-level constraints that may not be fully represented by individual objects.

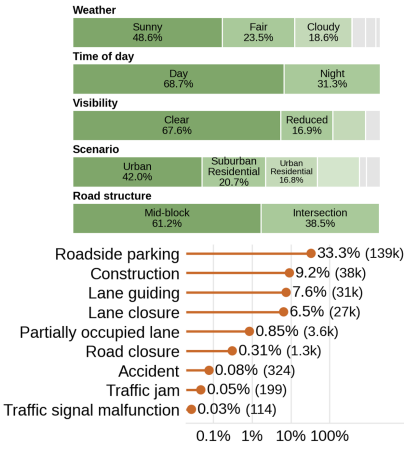


Figure 3: Context and traffic event distribution. Not labeled: Weather: Rainy 4.4%, Unknown 3.4%, Fog 1.5%; Visibility: Limited 10.6%, Poor 4.9%; Scenario: Suburban 13.7%, Urban Highway 2.3%, Other 4.5%; Road structure: Other 0.3%.

The core of VG-CoT is the annotation of decision-critical elements, which directs the model toward visual evidence that influences the current driving decision (Fig. 2). Each element is annotated with both visual grounding (bounding boxes and segmentation masks), and structured semantic information. We group decision-critical elements into four major categories: vehicles, vulnerable road users (VRUs), obstacles, and traffic-control elements. For vehicles, VRUs, and obstacles, the annotated attributes include type, location, state, intention, and impact rank. Location describes both the relative direction of the element and its road-topological position. State represents the currently observed condition, whereas intention describes the likely short-term behavior, such as turning, or crossing the road. Impact rank indicates the relative priority of each element in the current decision, where a smaller rank number denotes a stronger influence on ego driving. This ranking guides the model to allocate attention among multiple critical elements in a human-like manner. For traffic-

control elements, other than the impact rank, the annotations capture information such as traffic-light states, sign semantics, and descriptions of temporary traffic control. Each decision-critical element is further associated with an implication, which explains how the observed element may affect the ego vehicle and what action responses are appropriate. This encourages the model to reason about one critical factor at a time before integrating multiple factors, thereby reducing interference from unrelated information and facilitating the subsequent global reasoning process.

Finally, the reasoning and planning stage consolidates multiple element-level implications to derive the final driving decision. The rationale weighs these cues according to their relevance and impact, producing longitudinal and lateral actions. Because high-level actions can serve as an intermediate representation between semantic reasoning and continuous trajectory, AnchorReasoning includes both action annotations and future trajectories, supporting models that predict either meta-actions or trajectories directly.

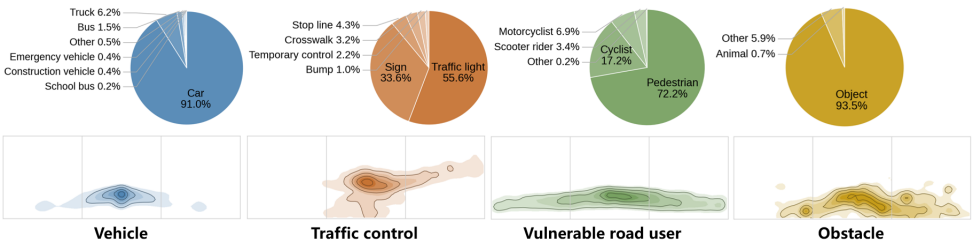


Figure 4: Decision-critical element composition and spatial distribution.

3.3 Human-in-the-Loop Annotation and Quality Control

We adopt a hybrid human-in-the-loop annotation pipeline that combines human judgment, VLM-assisted annotation, promptable segmentation, and rule-based methods. At the scene level, Qwen3-VL generates contextual labels, while human annotators identify traffic events. For decision-critical elements, annotators with driving experience first identify and rank the elements they would prioritize under the same driving situation. To improve consistency, they follow four guidelines: prioritizing traffic-control elements, nearby elements, moving elements, and elements likely to affect the ego vehicle's future driving space. At most five decision-critical elements are retained per frame to keep the supervision concise. Annotators then provide point prompts for each selected element, which are passed to SAM3 to generate instance masks and corresponding bounding boxes.



Figure 5: Predefined spatial regions.

Element attributes are subsequently obtained through human and automatic annotation. Human annotators provide element states and short-term intentions, while Qwen3-VL generates element types, sign content, and temporary traffic-control descriptions. Element location combines manually annotated road-topological relations with zone-based relative directions inferred from bounding-box positions and their overlap with predefined regions (Fig. 5). Traffic-light states are determined using Hue–Saturation–Value-based rules within the detected traffic-light regions (Appendix B.2).

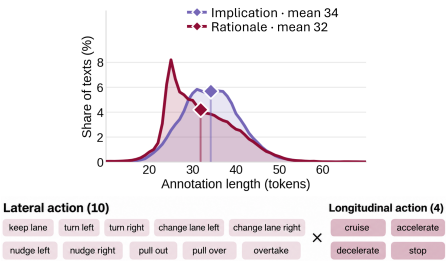


Figure 6: Free-text annotation length and action space.

Based on these grounded element annotations, implications are generated using a complexity-aware strategy. Scenes with no more than two decision-critical elements are processed using a locally deployed Qwen3-VL, whereas scenes with richer interactions are handled by GPT-5.2. The same strategy is applied to the final reasoning and planning stage. This tiered pipeline can balance annotation quality and cost.

Quality Control. Before large-scale annotation, each automatic annotation tool is evaluated to verify its reliability; the evaluation method and quantitative results are provided in Appendix B. After annotation is completed, we employ a multi-stage quality-control procedure. For context and traffic event annotations, 50% of the samples are randomly reviewed and corrected by human annotators. Decision-critical element selection and impact ranking are further verified through cross-annotation review. Each sample is checked by a second annotator, and substantial disagreements are resolved by a third annotator through majority voting. For long-form annotations, including implications, rationales, and action plans, recurrent error patterns are summarized and used to refine the system prompts, after which the corresponding annotations are regenerated. We additionally apply a rule-based consistency check between action plans and future trajectories to enforce expected geometric and kinematic consistency (Appendix B.6). Inconsistent samples are flagged for manual review and correction.

3.4 Dataset Statistics

The dataset contains approximately **395K** decision-critical elements across **4** high-level categories and **19** fine-grained element types, providing extensive object-level supervision for identifying driving-relevant evidence. Their grounded locations span the three-camera images and exhibit different spatial distributions across element classes, enabling models to associate heterogeneous visual evidence with its semantic role in driving (Fig. 4). The free-text annotations remain compact, with mean lengths of **34** tokens for element-level implications and **32** tokens for rationales, supporting concise inference from individual element effects to integrated driving decisions. Finally, the action space comprises **4** longitudinal and **10** lateral actions, providing semantic planning supervision that connects grounded reasoning with motion planning (Fig. 6).

4 Curriculum-Based Supervised Fine-Tuning

Stage I: Perception and Grounding. Stage I focuses on learning visual perception and proceeds in two phases. In phase a, supervision is limited to scene context, traffic events, and binary decision-critical element presence, establishing coarse scene understanding before element-level reasoning. Phase b introduces element identification, visual grounding, impact ranking, and attribute understanding. Throughout both phases, the complete VG-CoT target is retained, but locked fields are masked from the loss and contribute no gradient. This preserves a consistent output structure while allowing the model to progressively acquire the capabilities required for finer-grained perception.

Stage II: Reasoning and Planning. Stage II is initialized from the best Stage-I checkpoint and supervises the complete VG-CoT chain, including element-level implication, driving action rationale, action and trajectory generation. At this stage, the visual encoder is frozen to preserve the grounding capability learned in Stage I while the model learns higher-level reasoning and planning. We additionally apply trajectory-only supervision to a subset of training samples, where only trajectory tokens contribute to the loss, to preserve a direct perception-to-action pathway. We also increase the sampling probability of frames containing longitudinal action transitions and rare element states to improve the coverage of underrepresented cases.

Training Objective. For each target token t, we define a curriculum-aware weight $w_t = m_t^{(s)} W_{\phi(t)} \rho_t$, where $m_t^{(s)}$ is the stage-dependent supervision gate that determines which VG-CoT fields contribute to the loss at each curriculum. $W_{\phi(t)}$ is the base weight of the annotation field, and ρ_t adjusts the contribution of informative or underrepresented targets. In particular, rare attribute values are upweighted, while trajectory waypoints that deviate more from constant-velocity motion receive greater emphasis.

The overall training objective combines a VG-CoT chain loss and a trajectory loss:

$$\mathcal{L} = \mathcal{L}_{\text{chain}} + \lambda \mathcal{L}_{\text{traj}}, \quad \mathcal{L}_{\text{chain}} = \frac{\sum_{t \in \mathcal{C}} w_t \ell_t}{\sum_{t \in \mathcal{C}} w_t}, \quad \mathcal{L}_{\text{traj}} = \frac{\sum_{t \in \mathcal{J}} w_t \ell_t}{\sum_{t \in \mathcal{J}} w_t}. \tag{1}$$

Here, $\ell_t = -\log p_\theta(y_t \mid y_{<t}, x)$ denotes the token-level cross-entropy loss. $\mathcal{C}$ and $\mathcal{J}$ denote the supervised VG-CoT chain tokens and trajectory tokens, respectively. $\mathcal{L}_{\text{chain}}$ supervises the perception-to-action reasoning chain, while $\mathcal{L}_{\text{traj}}$ optimizes future trajectory generation. The coefficient λ con-

trols the balance between the two objectives and is set to 1 in all experiments. We normalize the two terms independently so that their relative weighting is not determined by differences in token sequence length. Stage I optimizes only $\mathcal{L}_{\text{chain}}$, whereas Stage II jointly optimizes both objectives.

5 EXPERIMENTS

5.1 MODELS

We evaluate eight representative models to assess the effectiveness of AnchorReasoning across diverse model foundations. **(i) General-purpose VLMs.** Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B (Bai et al., 2025a) serve as general-purpose baselines.**(ii) Embodied-AI foundation models.** Cosmos-Reason2-2B and Cosmos-Reason2-8B (NVIDIA, 2025) provide embodied-reasoning baselines without driving-specific adaptation. **(iii) AV-specific VLMs.** Alpamayo-R1-10B and Alpamayo-1.5-10B (Wang et al., 2025) are driving-specialized models capable of joint reasoning and trajectory prediction. AutoVLA-3B (Zhou et al., 2026) and Impromptu-VLA-7B (Chi et al., 2026) are included as complementary open-source driving baselines. Detailed training and implementation settings are provided in Appendix C.1, C.2.

5.2 EVALUATION METRICS

Scene-Level Semantics. Context Acc. measures the average accuracy of scene-context predictions, while Traffic Event F1 evaluates the detection of traffic events.

Decision-Critical Elements. Presence Acc. measures whether the model correctly predicts the presence of decision-critical elements in a frame. For the grounding measurement, because the model predicts a point rather than a bounding box, we match predicted elements to ground-truth elements using a one-to-one, scale-adaptive point-to-mask criterion. For each ground-truth element, the point-to-mask distance is zero when the predicted point falls inside its segmentation mask and otherwise equals the minimum Euclidean distance to the mask. A prediction is considered matched when this distance falls within a scale-adaptive assignment threshold determined by the ground-truth element size.

We report Element Recall $R_{\text{det}} = |\mathcal{M}|/|\mathcal{G}|$, where $\mathcal{G}$ is the set of ground-truth decision-critical elements and $\mathcal{M}$ is the subset successfully matched to model predictions. We also report Rank-1 Recall to evaluate whether the most influential element is retrieved, Element Type Acc. to assess type prediction among matched elements, and Point-in-Mask Rate to measure localization precision:

$$R_{\text{in}} = \frac{1}{|\mathcal{M}|} \sum_{g \in \mathcal{M}} \mathbf{1}\left[x_{\pi(g)} \in M_g\right], \tag{2}$$

where $\pi(g)$ denotes the prediction matched to ground-truth element g, $x_{\pi(g)}$ is its predicted image point, M_g is the segmentation mask of g, and $\mathbf{1}[\cdot]$ is the indicator function. For matched elements, Element State Recall, Element Intention Recall, and Sign-Content Recall evaluate the corresponding semantic attributes, while R/G Conf. measures the red–green traffic-light confusion rate.

Reasoning and Trajectory Prediction. We use GPT-5.5 to evaluate element-level implications and frame-level rationales. Element Implication measures whether the model correctly describes each decision-critical element and its effect on ego driving, while Driving Action Rationale measures whether it correctly consolidates multiple factors into the final driving decision. For trajectory prediction, we report Average Displacement Error (ADE) and Final Displacement Error (FDE). We additionally report the official WOD-E2E RFS (Xu et al., 2026). Detailed definitions and implementation protocols for the metrics are provided in Appendix C.3.

5.3 RESULTS

To evaluate the effectiveness of VG-CoT supervision, we compare the Base and VG-CoT models across scene-level semantic understanding, decision-critical element understanding and reasoning, trajectory prediction, and inference latency. We further conduct ablation studies to assess the contribution of visual grounding to trajectory prediction. Training details are provided in Appendix C.2.

Table 2: **Scene-level semantics and decision-critical elements.**

Model	Scene-Level Semantics		Decision-Critical Elements								
	Context Acc. ↑	Traffic Event F1 ↑	Presence Acc. ↑	Element Recall ↑	Rank-1 Recall ↑	Element Type Acc. ↑	Point-in-Mask Rate ↑	Element State Recall ↑	Element Intention Recall ↑	Traffic Control -Content Recall ↑	R/G Conf. ↓
Qwen2.5-VL-7B	– / 0.83	– / 0.72	0.80 / 0.87	0.43 / 0.74	0.38 / 0.79	0.89 / 0.96	0.57 / 0.85	0.25 / 0.46	0.22 / 0.46	– / 0.95	0.042 / 0.029
Qwen3-VL-8B	– / 0.82	– / 0.71	0.64 / 0.88	0.10 / 0.79	0.10 / 0.83	0.63 / 0.98	0.18 / 0.91	0.13 / 0.47	0.00 / 0.47	– / 0.95	0.037 / 0.026
Cosmos-Reason2-2B	– / 0.83	– / 0.75	0.81 / 0.88	0.10 / 0.78	0.10 / 0.81	0.71 / 0.95	0.31 / 0.86	0.15 / 0.46	0.02 / 0.46	– / 0.98	0.029 / 0.025
Cosmos-Reason2-8B	– / 0.82	– / 0.71	0.67 / 0.89	0.17 / 0.78	0.17 / 0.82	0.62 / 0.97	0.63 / 0.85	0.01 / 0.48	0.00 / 0.48	– / 0.95	0.034 / 0.013
Alpamayo-R1-10B	– / 0.82	– / 0.74	0.83 / 0.89	0.49† / 0.62	0.67 / 0.69	– / 0.90	– / 0.40	– / 0.45	– / 0.42	– / 0.92	– / 0.044
Alpamayo-1.5-10B	– / 0.82	– / 0.70	0.84 / 0.91	0.46† / 0.79	0.66 / 0.82	– / 0.98	– / 0.92	– / 0.71	– / 0.58	– / 0.97	– / 0.028
AutoVLA-3B	– / 0.82	– / 0.72	– / 0.86	– / 0.79	– / 0.81	– / 0.95	– / 0.80	– / 0.46	– / 0.47	– / 0.97	– / 0.014
Impromptu-VLA-7B	– / 0.83	– / 0.71	– / 0.86	– / 0.67	– / 0.76	– / 0.96	– / 0.85	– / 0.47	– / 0.48	– / 0.96	– / 0.028

Note. Values are reported as Base / VG-CoT. Base denotes the domain-adapted model trained on the original WOD-E2E data, whereas VG-CoT denotes the same model trained with VG-CoT supervision from AnchorReasoning. "–" indicates that the capability is unavailable or cannot be evaluated. † indicates type-only matching because grounding ability is unavailable. ↑/↓ indicates that higher/lower values are better, respectively.

Scene-Level Semantics and Decision-Critical Elements. Table 2 shows that VG-CoT supervision consistently improves decision-critical element understanding across different model families, with the clearest gains in element retrieval, prioritization, spatial grounding, and semantic interpretation. It also enables scene- and element-level predictions that are unavailable in several Base models. More importantly, VG-CoT allows models not only to state decision-critical elements in text but also to localize them in the image (Fig. 7), creating a direct connection between semantic reasoning and visual evidence. For the general-purpose Qwen models and the Cosmos embodied-AI foundation models, the Base models can often recognize the types of relevant elements but tend to retrieve only a limited subset of the elements that influence the driving decision. After VG-CoT training, both Element Recall and Rank-1 Recall improve consistently. Rank-1 Recall is also higher than overall Element Recall for every trained backbone, indicating that the supervision is particularly effective at capturing the elements with the strongest influence on the driving decision, even when not all relevant elements are retrieved. The AV-specific models exhibit a different pattern. The Base Alpamayo models already show stronger decision-critical element retrieval than most general-purpose models, but their native outputs lack explicit grounding and several structured semantic attributes. VG-CoT adds these capabilities while preserving strong retrieval performance. At the same time, the contrast between the Alpamayo variants shows that identifying an important element does not necessarily imply precise visual localization, reinforcing the need to evaluate retrieval and grounding separately. Across all model families, VG-CoT also strengthens understanding of element type, state, intention, and traffic-control content while reducing red–green traffic-light confusion. Element Type Accuracy and Traffic Control-Content Recall are generally stronger than Element State and Intention Recall, suggesting that recognizing element identity and relatively static traffic-control semantics is easier than inferring dynamic behavior and intent from a single frame. Overall, the results indicate that VG-CoT provides more integrated supervision, connecting scene understanding, decision-critical element retrieval, visual grounding, and decision-relevant semantic interpretation rather than improving only a single perception task.



Figure 7: Example VG-CoT output of Alpamayo-1.5-10B.

Reasoning, Trajectory Prediction, and Inference Efficiency. Table 3 shows that VG-CoT supervision improves both Element Implication and Driving Action Rationale across all model families, with average relative gains of 85.8% and 77.9%, respectively. Element Implication consistently

Table 3: **Reasoning, trajectory prediction, and inference efficiency.**

Model	Reasoning		Trajectory Prediction						Inference Efficiency	
	Element Implication ↑	Driving Action Rationale ↑	ADE@1s ↓	ADE@3s ↓	ADE@5s ↓	FDE@5s ↓	RFS Frame ↑	RFS Cluster ↑	Avg. Tokens	Latency (s/frame) ↓
Qwen2.5-VL-7B	0.05 / 0.78	0.47 / 0.69	4.39 / **0.25**	8.81 / **1.48**	12.99 / **3.35**	24.15 / **8.23**	5.24 / **7.33**	5.27 / **7.39**	378 / 365	4.76 / 4.62
Qwen3-VL-8B	0.54 / 0.79	0.36 / 0.72	3.72 / **0.22**	8.44 / **1.34**	12.95 / **3.04**	23.95 / **7.54**	5.37 / **7.54**	5.38 / **7.56**	384 / 398	4.77 / 4.90
Cosmos-Reason2-2B	0.07 / 0.75	0.05 / 0.71	16.36 / **0.22**	24.03 / **1.35**	27.88 / **3.06**	36.27 / **7.58**	5.21 / **7.56**	5.20 / **7.59**	617 / 405	7.68 / 5.07
Cosmos-Reason2-8B	0.35 / 0.80	0.22 / 0.72	5.84 / **0.22**	11.36 / **1.34**	16.23 / **3.02**	28.11 / **7.43**	5.28 / **7.56**	5.32 / **7.59**	529 / 389	9.12 / 6.74
Alpamayo-R1-10B	0.61 / 0.70	0.60 / 0.61	0.24 / **0.23**	1.46 / **1.38**	3.37 / **2.97**	8.45 / **7.40**	7.05 / **7.66**	7.06 / **7.69**	– / 403	– / 7.00
Alpamayo-1.5-10B	0.69 / 0.84	0.61 / 0.80	0.26 / **0.23**	1.55 / **1.34**	3.49 / **2.94**	8.56 / **7.06**	6.66 / **7.81**	6.69 / **7.83**	– / 359	– / 6.24
AutoVLA-3B	0.48 / 0.76	0.32 / 0.69	0.62 / **0.30**	2.49 / **1.64**	5.01 / **3.60**	11.49 / **8.69**	6.18 / **7.17**	6.10 / **7.24**	341 / 373	5.68 / 6.21
Impromptu-VLA-7B	0.53 / 0.75	0.54 / 0.70	0.23 / **0.22**	2.24 / **1.40**	5.96 / **3.17**	15.68 / **7.84**	6.31 / **7.45**	6.32 / **7.47**	144 / 352	1.91 / 4.46
Avg. Improvement (%)	85.8%	77.9%	94.0%	81.3%	71.4%	60.6%	27.0%	27.5%	4.6%	5.7%

Note. Values are reported as Base / VG-CoT. For Alpamayo models, efficiency metrics are marked as "-" because trajectory generation relies on a separate diffusion module and is therefore not directly comparable with the other models.

scores higher than Driving Action Rationale, indicating that understanding the effect of an individual decision-critical element is easier than consolidating multiple factors into a final driving decision. These reasoning gains are accompanied by improved trajectory prediction: on average, 5-s ADE and FDE decrease by 71.4% and 60.6%, while RFS Frame and Cluster improve by 27.0% and 27.5%, respectively.

The visual-grounding ablation in Table 4 further shows that VG-CoT consistently outperforms its non-grounded variant across all trajectory metrics for the three evaluated backbones, indicating that visually grounded CoT provides additional benefits for trajectory prediction. Despite the additional reasoning structure, VG-CoT does not require longer outputs. The average output length decreases by 18.5 tokens and latency by 0.32 s/frame while trajectory accuracy improves, suggesting that VG-CoT can provide more compact reasoning without sacrificing planning performance.

Table 4: Ablation results on trajectory prediction.

Model	ADE @1s ↓	ADE @3s ↓	ADE @5s ↓	FDE @5s ↓	RFS Frame ↑	RFS Cluster ↑
Alpamayo-1.5	0.26 / **0.23**	1.47 / **1.34**	3.26 / **2.94**	7.97 / **7.06**	7.10 / **7.81**	7.13 / **7.83**
Qwen3-VL	0.32 / **0.22**	1.48 / **1.34**	3.21 / **3.04**	7.87 / **7.54**	7.44 / **7.54**	7.45 / **7.56**
Cosmos-R2-2B	0.33 / **0.22**	1.53 / **1.35**	3.32 / **3.06**	8.00 / **7.58**	7.35 / **7.56**	7.39 / **7.59**

Note. Values are reported as Ablation / VG-CoT.

6 Limitations and Future Work

The current annotations cover only the front-left, front, and front-right camera views, leaving the side and rear views unannotated. AnchorReasoning could be extended toward full-surround perception, and the annotated data could be used to train models that facilitate and scale future VG-CoT dataset annotation. The dataset also does not currently include explicit depth supervision. Incorporating depth and depth-change information could further strengthen spatial grounding and the identification of decision-critical elements.

7 Conclusion

We introduce AnchorReasoning, a long-tail autonomous-driving dataset for VLMs that provides visually grounded supervision over decision-critical elements, their attributes and implications, driving-action rationale, and action and trajectory planning. Its VG-CoT structure explicitly connects visual evidence with reasoning and planning, while the curriculum SFT strategy enables models to progressively acquire these capabilities. We further introduce an object-size-aware grounding metric for evaluating decision-critical element localization. Experiments across eight general-purpose, embodied-AI, and AV-specific backbones show that VG-CoT supervision consistently improves grounded reasoning and trajectory prediction. Ablation studies further demonstrate that incorporating explicit visual grounding into CoT supervision contributes to improved trajectory prediction performance. Across models, VG-CoT reduces 5-s ADE and FDE by 71.4% and 60.6% on average, respectively, while improving RFS Frame and Cluster by 27.0% and 27.5%. These gains are achieved with 18.5 fewer reasoning tokens and a 0.32 s/frame reduction in inference latency. These results demonstrate the effectiveness of visually grounded, decision-focused supervision in improving both reasoning and planning for long-tail autonomous driving scenarios.

AI USE STATEMENT

Generative AI tools were used to assist with dataset annotation and data cleaning and reformatting. AI tools were also used for code writing and modification and for language editing of the manuscript to improve clarity and readability. All AI-assisted annotations, data processing outputs, code, and text were reviewed and verified by the authors before use. The authors take full responsibility for the final content, results, and claims of this work.

ETHICS STATEMENT

This work is built upon the Waymo Open Dataset. We do not redistribute the original Waymo data; only the annotations introduced in this work are released. Users must obtain the corresponding source data separately under the terms of the Waymo Open Dataset license.

REPRODUCIBILITY STATEMENT

Additional implementation and experimental details are provided in the appendix after the references. The code is included in the supplementary material. The dataset, model weights, and code will be publicly released upon acceptance.

ACKNOWLEDGMENTS

We thank XX for their contributions to data annotation. We also thank XX for their valuable input on experimental design and manuscript development. We thank XX for providing the computational resources used in this work.

REFERENCES

Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. CoVLA: Comprehensive vision-language-action dataset for autonomous driving. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1933–1943. IEEE, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

Chun-Peng Chang, Chen-Yu Wang, Holger Caesar, and Alain Pagani. Probing the reliability of driving VLMs: From inconsistent responses to grounded temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 644–654, 2026.

Haohan Chi, Huan-ang Gao, Ziming Liu, Jianing Liu, Chenyu Liu, Jinwei Li, Kaisen Yang, Yangcheng Yu, Zeda Wang, Wenyi Li, et al. Impromptu VLA: Open weights and open data for driving vision-language-action models. *Advances in Neural Information Processing Systems*, 38, 2026.

Zhiyu Huang, Johnson Liu, Rui Song, Zewei Zhou, Ruining Yang, Yun Zhang, Tianhui Cai, Hanyin Zhang, Mingxuan Gao, Valeria Xu, et al. nuReasoning: A reasoning-centric dataset and benchmark for long-tail autonomous driving. *arXiv preprint arXiv:2605.31572*, 2026.

Ayesha Ishaq, Jean Lahoud, Ketan More, Omkar Thawakar, Ritesh Thawkar, Dinura Dissanayake, Noor Ahsan, Yuhao Li, Fahad Shahbaz Khan, Hisham Cholakkal, et al. DriveLMM-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20501–20508. IEEE, 2025.

Keishi Ishihara, Kento Sasaki, Tsubasa Takahashi, Daiki Shiono, and Yu Yamaguchi. STRIDE-QA: Visual question answering dataset for spatiotemporal reasoning in urban driving scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 5257–5266, 2026.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *European Conference on Computer Vision*, pp. 577–593. Springer, 2018.

Chunsheng Liu, Xiao Zhang, Faliang Chang, Shuang Li, Penghui Hao, Yansha Lu, and Yinhai Wang. Traffic scenario understanding and video captioning via guidance attention captioning network. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):3615–3627, 2023.

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*, pp. 1043–1052, 2023.

Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. LingoQA: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pp. 252–269. Springer, 2024.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2Drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pp. 292–308. Springer, 2024.

NVIDIA. Cosmos-Reason2: Reasoning vision-language model for physical AI and robotics, 2025. URL `https://github.com/nvidia-cosmos/cosmos-reason2`. Hugging Face / NGC Model Family (2B, 8B, 32B variants).

Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, et al. AgentThink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. In *EMNLP (Findings)*, pp. 10663–10682, 2025.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. NuScenes-QA: A multimodal visual question answering benchmark for autonomous driving scenario. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4542–4550, 2024.

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*, pp. 7513–7522, 2024.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *Proceedings of the European Conference on Computer Vision*, pp. 256–274. Springer, 2024.

Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-R1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

Dongming Wu, Wencheng Han, Yingfei Liu, Tiancai Wang, Cheng-zhong Xu, Xiangyu Zhang, and Jianbing Shen. Language prompt for autonomous driving. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 8359–8367, 2025.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. Are VLMs ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6585–6597, 2025.

Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Kate Tolstaya, Sarah Tang, Brandyn White, et al. WOD-E2E: Waymo open dataset for end-to-end driving in challenging long-tail scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3709–3718, 2026.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. AutoVLA: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026.

A Dataset Details

A.1 Dataset Composition and WOD-E2E Splits

AnchorReasoning is constructed on top of the Waymo Open Dataset for End-to-End Driving (WOD-E2E) without modifying the original sensor data or dataset splits. We add semantic and reasoning annotations directly to the decoded WOD-E2E frames. Table 5 summarizes the correspondence between our annotations and the decoded WOD-E2E splits.

Table 5: Composition of AnchorReasoning with respect to our decoded copy of the WOD-E2E splits.

Split	# Decoded Segments	# Annotated Segments	# Annotated Frames
Train	2,037	2,037	415,663
Validation	479	455	456
Test	1,505	0	0
Total annotated	–	**2,492**	**416,119**

For the training split, all 415,663 decoded frames from 2,037 segments are annotated. For validation, we annotate the 456 frames for which WOD-E2E provides rater-feedback trajectories. These frames come from 455 of the 479 decoded validation segments and are used for both model evaluation and Rater Feedback Score (RFS) computation. Each frame is associated with three rated candidate trajectories, resulting in 1,368 human-rated trajectories in total. The remaining decoded validation frames are not included in the annotation statistics. The decoded test split is retained for completeness but is not annotated. Overall, AnchorReasoning contains 416,119 annotated frames from 2,492 WOD-E2E segments and 395,379 decision-critical elements.

A.2 Impact-Rank Distribution

The distribution is strongly concentrated at the highest ranks: 67.9% of the annotated elements are assigned rank 1, 25.3% rank 2, 5.6% rank 3, and 1.0% rank 4, while only 0.2% have rank 5 or higher. 93.2% of all decision-critical elements fall within the first two ranks and 98.8% within the first three, as shown in Fig. 8. This distribution reflects the annotation principle of retaining only elements that materially influence the current driving decision rather than exhaustively labeling all visible objects.

A.3 Action-Plan Distribution

Figure 9 shows the joint distribution of lateral and longitudinal actions. The distribution is naturally imbalanced, with routine lane-keeping action accounting for most frames and more complex lateral maneuvers forming a long tail. Importantly, lateral and longitudinal actions are not independent: the same lateral maneuver can be associated with substantially different longitudinal responses depending on the surrounding constraints and interaction state. The dataset therefore provides supervision not only for recognizing individual actions, but also for learning their joint behavioral structure.

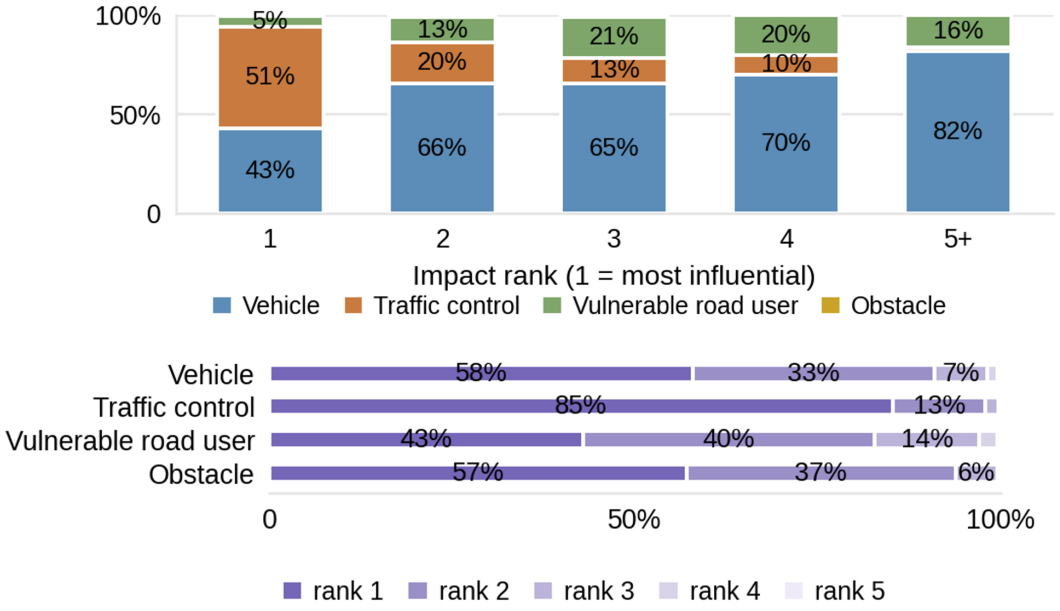


Figure 8: Distribution of Impact Ranks

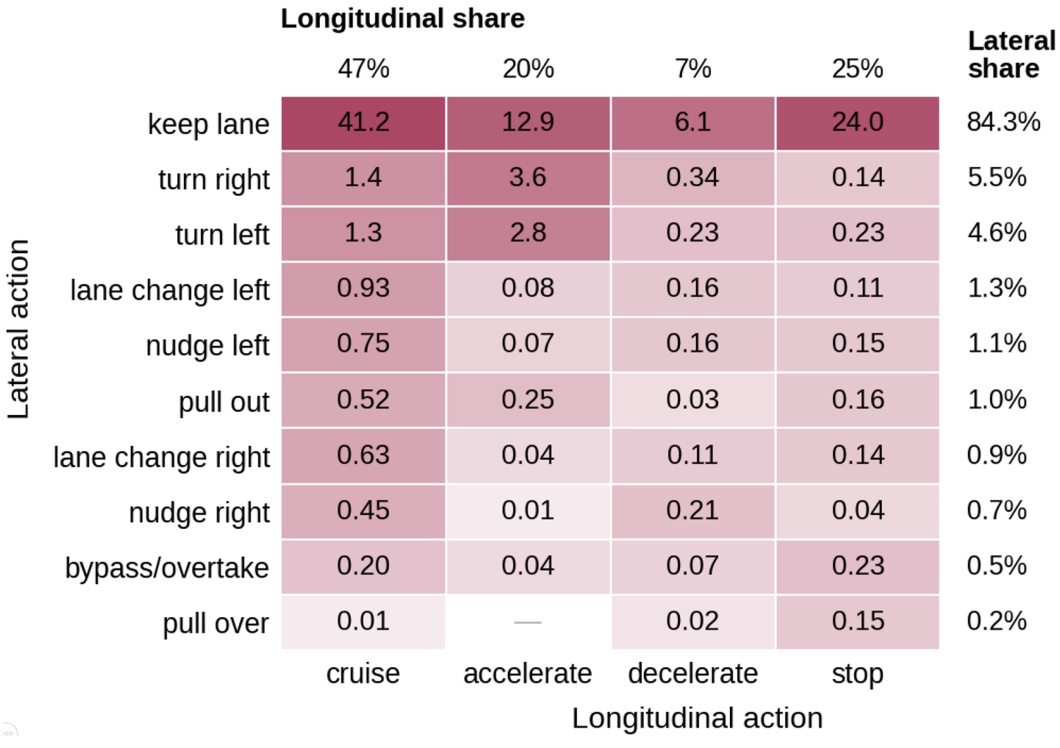


Figure 9: Joint distribution of the normalized lateral and longitudinal actions.

A.4 ANNOTATION EXAMPLES

Figure 10- 12 present the representative examples of the AnchorReasoning annotation. The annotation progressively connects scene context, decision-critical element grounding and attributes, element-level driving implications, frame-level reasoning, and the final action plan.



CONTEXT Sunny · Day · Clear · Urban · Mid-block · Roadside parking

OBJECTS

#1 Sign traffic control 1
content stop here on red light
implication The front-right sign reading "stop here on red light" requires the ego vehicle to hold behind the designated stop position while the signal is red, delaying straight-ahead progress until a permissive indication.

#2 Traffic light traffic control 2
content red
implication The front traffic light showing red closes the ego vehicle's straight-through right-of-way, so near-term motion is limited to decelerating to a stop and waiting for green to proceed.

#3 Stop line traffic control 3
implication The front stop line defines the exact boundary for stopping, so the ego vehicle's available forward motion is constrained to braking to a halt before this line while waiting for the red to clear.

REASONING The signal ahead is red and the signage requires stopping at the designated stop position, so the ego must brake to a complete halt before the stop line and hold until a permissive indication.

PLANNING **decelerate to a full stop at the stop line and wait**

Figure 10: Examples of Complete Visual and Textual Annotations.

B AUTOMATIC ANNOTATION TOOLS

This section describes the automatic tools used to assist dataset annotation and evaluates their reliability before large-scale deployment. To assess annotation quality, we use a manually annotated subset of the dataset as reference and compare the outputs of each automatic tool against the corresponding human labels. The automatic tools are applied independently without access to these reference labels. Because human annotations may themselves contain ambiguity, the reported results measure human–automatic agreement rather than absolute annotation accuracy.



CONTEXT Sunny Day Clear Urban Residential Mid-block

OBJECTS

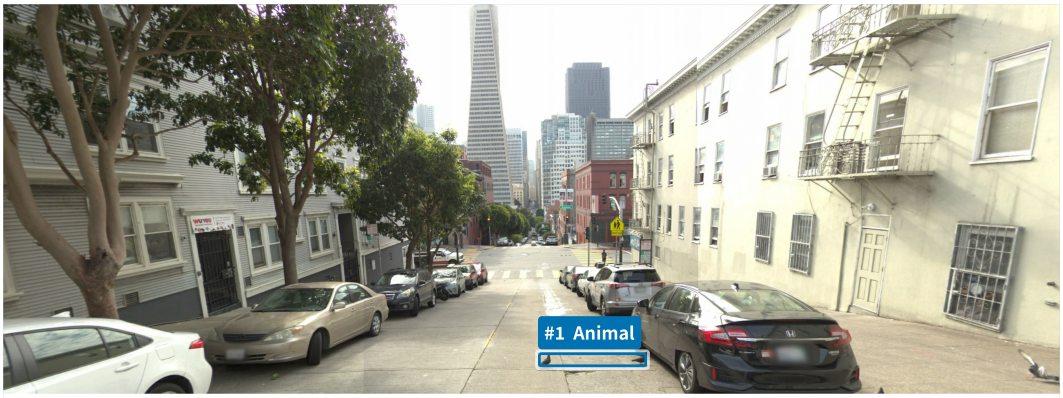
#1 Object obstacle 1
description flatbed location front right in the same lane
implication The flatbed obstacle front-right in the same lane occupies the right side of the ego vehicle's straight-ahead corridor, narrowing usable space and potentially requiring a slight left shift to pass if the remaining lane width stays open.

#2 School bus vehicle 1
location front left in the oncoming lane state stopping intention stopping
implication The stopping school bus front-left in the oncoming lane reduces available lateral buffer for any leftward avoidance around the flatbed, so the ego vehicle's ability to borrow oncoming space is constrained until the bus remains stopped or clears.

REASONING The flatbed narrows the ego's forward corridor while the stopped oncoming bus limits safe leftward clearance, so slowing down maintains lane keeping and preserves time/space to pass without needing to borrow the oncoming lane.

PLANNING **keep lane and decelerate**

Figure 11: Examples of Complete Visual and Textual Annotations.



CONTEXT Cloudy | Day | Clear | Urban Residential | Mid-block | Roadside parking

OBJECTS #1 **Animal** obstacle 1
description birds location front in the same lane
implication The animal obstacle 1 located directly ahead in the ego vehicle's lane indicates a potential obstruction that may require the ego vehicle to slow down or wait for the animal to clear the path before continuing straight.

REASONING The animal obstacle directly ahead blocks the ego vehicle's path, requiring the ego to slow down and wait for the obstacle to clear before proceeding.

PLANNING **decelerate and keep lane**

Figure 12: Examples of Complete Visual and Textual Annotations.

B.1 SCENE CONTEXT

Method. We use Qwen3-VL-8B-Instruct to classify five scene-level attributes from the image: weather, time of day, visibility, scenario type, and road structure. The model receives only the image and a predefined label taxonomy, and returns a structured JSON output containing the predicted value, confidence, and brief visual evidence. To reduce frame-level fluctuations, predictions are further stabilized within each video. Weather and visibility are kept temporally consistent, missing scenario labels are propagated from adjacent frames, and isolated road-structure changes are detected and corrected. The complete prompt used for scene-context annotation is shown in Fig. 15.

Reliability Evaluation. We evaluate the context classifier on 7,450 manually annotated frames. For each context field, human–automatic agreement is computed as

$$A = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1} \left[\hat{y}_i = y_i^{\text{human}} \right], \tag{3}$$

where y_i^{human} is the human annotation, $\hat{y}_i$ is the automatic prediction, and $\mathbf{1}[\cdot]$ denotes the indicator function.

Table 6: Human–automatic agreement for scene-context annotations.

Field	# Frames	Agreement
Time of day	7,450	100.00%
Visibility	7,450	88.68%
Weather	7,450	83.57%
Road structure	7,450	82.32%
Scenario	7,450	76.51%

B.2 TRAFFIC-LIGHT STATE

Method. Traffic-light states are determined using a rule-based HSV classifier. For each annotated traffic light, pixels are extracted from its instance segmentation mask rather than the full bounding box to reduce background contamination. RGB pixels are converted to HSV space, and bright pixels with $V \geq 150$ are selected for analysis. If fewer than 10 bright pixels are available, all pixels within the instance mask are used instead. The adopted HSV thresholds are shown in Table 7. A color is considered detected only when at least 10 pixels satisfy the corresponding thresholds and account for at least 0.5% of the valid pixels.

Table 7: HSV thresholds used for traffic-light state recognition.

State	Hue Range	$\mathbf{S}_{\min}$	$\mathbf{V}_{\min}$
Red	$[0, 25] \cup [160, 180]$	50	80
Yellow	$[20, 50]$	25	120
Green	$[50, 95]$	40	40

Reliability Evaluation. Evaluation is performed on three manually annotated partitions containing 11,318 matched traffic-light instances. The classifier achieves 97.81% overall agreement. Red and green recognition is highly reliable, while yellow remains the primary source of error.

The red–green confusion rate ranges from 0.40% to 1.34%. In contrast, yellow recall is 0.42 on p19 and 0.59 on p20, despite relatively high precision. This motivates the conservative labeling strategy used in dataset construction: automatic predictions are used to lbael red and green states. Human annotate the yellow labels.

We additionally evaluate three threshold configurations to examine the trade-off between yellow recall and false yellow detections. The results on p19 are summarized in Table 9.

Table 8: Traffic-light recognition results on the evaluation partitions.

Partition	# Lights	Agreement	Red Recall	Green Recall
p19	5,300	98.28%	0.9935	0.9938
p20	4,869	97.00%	0.9949	0.9733
p21	1,149	99.04%	0.9832	1.0000
All	11,318	97.81%	–	–

Table 9: Ablation of traffic-light color thresholds.

Setting	Overall Agreement	Yellow Recall	Yellow Precision
Adopted	98.28%	0.42	0.95
Relaxed yellow	73.47%	0.87	0.06
Intermediate	97.02%	0.69	0.41

Relaxing the yellow threshold substantially improves yellow recall but introduces a large number of false yellow predictions from red lights. The adopted configuration therefore provides a more reliable overall trade-off.

B.3 Object Location

Method. Object bearing is labeled using the fixed geometry of the iamge. Five spatial regions are manually defined on a reference image and reused for all frames.

For each vehicle, vulnerable road user, or obstacle, we compute the overlap ratio between its bounding box B and each spatial region R_k:

$$s_k = \frac{\mathrm{Area}(B \cap R_k)}{\mathrm{Area}(B)}. \tag{4}$$

The object is assigned to the region with the largest overlap. When the difference between the two highest overlap scores is no greater than 0.05, the bottom-center point of the bounding box is used as an additional point-in-polygon cue because it more closely approximates the object's contact position on the road.

Only the directional prefix of the location label is automatically labeled, while the road-topological relation annotation is labeled by human. For example, for the label *front left in the oncoming lane*, only *front left* is automatically labeled.

Reliability Evaluation. We compare the automatically determined directional prefix with the human annotation on 4,439 aligned objects. The two annotations agree on 4,369 instances, corresponding to an agreement rate of 98.42%. Most disagreements occur between neighboring spatial regions, such as *front* and *front left*, rather than between opposite directions. This indicates that the remaining errors are primarily associated with region boundaries.

B.4 Vehicle Type

Method. Vehicle types are classified using Qwen3-VL-8B-Instruct. Each vehicle is cropped from the image using its bounding box with a 20-pixel margin and classified into one of seven categories: *Car*, *Truck*, *Emergency vehicle*, *Bus*, *Construction vehicle*, *School bus*, and *Other*. The prompt is shown in Fig. 16

Reliability Evaluation. The classifier is evaluated on 3,720 manually labeled vehicles and agrees with the human annotations on 3,681 instances, yielding 98.95% agreement. The remaining disagreements are concentrated around visually similar vehicle categories, particularly the boundary between cars, pickup trucks, and large van-like vehicles.

B.5 Reasoning Chain Generation

Method. The reasoning chain is generated in two stages. First, an element-level driving implication is generated for each decision-critical element. The inputs include the grounded iamge, scene context, ego state, navigation intent, and the structured attributes of the element. The implication describes how the element affects the ego vehicle's intended path, available motion, right-of-way, or local maneuver options. Each element is interpreted independently so that its causal effect is established before multiple factors are integrated. Second, the element-level implications are combined with their impact ranks, ego state, and navigation intent to generate a frame-level rationale and final action plan. The rationale integrates the dominant constraints and enabling factors, while the final action plan specifies the resulting short-horizon longitudinal and lateral action. The plan is constrained to the action vocabulary used in AnchorReasoning.

To balance annotation quality and computational cost, scenes are routed according to annotation complexity. Simple scenes, defined as those containing at most one decision-critical element or exactly two traffic-control elements, are processed using a locally deployed Qwen3.5-122B model, while more complex scenes are processed using GPT-5.2. All generated implications, rationales, and action plans are subsequently inspected and corrected through a dedicated annotation interface. Consecutive frames with identical annotations may reuse the previous output to improve temporal consistency and reduce unnecessary inference. The complete prompt used for driving implication annotation and rationale is shown in Fig. 17, 18.

B.6 Action–Trajectory Consistency Check.

After annotation, we apply a rule-based consistency check to verify whether each annotated action plan agrees with the corresponding future trajectory. The action plan is mapped to one of four longitudinal categories, {*accelerate*, *decelerate*, *cruise*, *stop*}, and, when explicitly specified, to one of three lateral categories, {*left turn*, *right turn*, *straight*}.

The future trajectory is converted to the same motion representation using kinematic and geometric rules. Let v_0 denote the current ego speed, v_{end} the mean speed over the final 1 s of the future trajectory, and $v_{\max}$ the maximum future speed. The longitudinal trajectory category is defined as

$$\psi_{\text{lon}}(\tau) = \begin{cases} \text{stop}, & v_{\text{end}} < 0.5, \\ \text{accelerate}, & (v_0 < 0.5 \wedge v_{\max} \geq 1.0) \vee (v_{\text{end}} - v_0 > \max(1.0, 0.2v_0)), \\ \text{decelerate}, & v_0 - v_{\text{end}} > \max(1.0, 0.2v_0) \wedge v_{\text{end}} \geq 0.5, \\ \text{cruise}, & \text{otherwise}. \end{cases} \tag{5}$$

For lateral motion, the trajectory is classified as a left or right turn when the absolute heading change exceeds 25°, with the sign determining the turning direction; otherwise, it is classified as straight.

Consistency is checked using the compatibility rules rather than strict category equality. For example, an annotated *decelerate* action is compatible with a trajectory that decelerates or stops, whereas an annotated *stop* action is inconsistent with a trajectory that immediately continues at normal speed. A sample is considered consistent only when its longitudinal action and any explicitly specified lateral action are non-contradictory. Inconsistent samples are flagged for manual review and correction.

C Training and Evaluation Details

C.1 Hardware and Implementation Details

All training and evaluation experiments are conducted on NVIDIA B200 GPUs, with two GPUs used for training. Training uses distributed data parallelism with DeepSpeed ZeRO Stage 2 and bfloat16 precision. FlashAttention-2 is enabled for efficient attention computation, and gradient checkpointing is applied to reduce memory consumption. We use full-parameter fine-tuning for all trainable components without LoRA or other parameter-efficient adaptation methods.

C.2 Training Configuration

For both domain adaptation and our Supervised Fine-Tuning(SFT), each sample uses the same input representation: a single panorama stitched from three forward-facing cameras, the ego-vehicle motion history, and the high-level navigation intent. The motion history contains 16 waypoints over the preceding 4 s and is provided as textual input. No temporal image sequence, LiDAR, HD map, or auxiliary prediction head is used.

Domain Adaptation. For the domain-adapted baselines, training is performed on the original WOD-E2E data without using the additional AnchorReasoning annotations. We preserve each model's original training objective, output format, and training configuration as much as possible. The input also follows each model's original formulation, with the only modification being the visual input, which is replaced by a single-frame panorama stitched from three forward-facing cameras. The target output follows each model's original formulation, such as its native reasoning, action, or trajectory representation, rather than the visually grounded chain-of-thought (VG-CoT) annotation format.

Supervised Fine-Tuning. Our SFT uses the same input representation but supervises the model with the structured VG-CoT output. Stage I focuses on scene-level semantics and decision-critical element understanding, including traffic events, element presence, element identification, visual grounding, impact rank, and element attributes. Stage II is initialized from the best Stage-I checkpoint and extends the output to the complete VG-CoT chain, including ego-state understanding, element-level implications, driving-action rationale, action planning, motion, and future trajectory.

Table 10 summarizes the optimization settings used for SFT across all backbones. Stage I starts from the original pretrained or released backbone, whereas Stage II is initialized from the best Stage-I checkpoint. The same optimization settings are used across backbones unless otherwise specified.

Table 10: **Training hyperparameters for Stage I and Stage II.**

Hyperparameter	Stage I	Stage II	Hyperparameter	Stage I	Stage II
Epochs	2	2	Optimizer	AdamW	AdamW
Per-device batch size	1	1	Adam β_1, β_2	0.9, 0.999	0.9, 0.999
Effective batch size	64	64	Adam ϵ	10^{-8}	10^{-8}
Learning rate	1×10^{-5}	5×10^{-6}	Weight decay	0	0
LR scheduler	Cosine	Cosine	Max. gradient norm	1.0	1.0
Warmup ratio	0.03	0.03	Precision	bf16	bf16
Visual encoder	Trainable	Frozen	DeepSpeed	ZeRO-2	ZeRO-2
Visual LR multiplier	0.1	–	Random seed	42	42

C.3 Evaluation Metrics

Scene and Element-Level Semantics. We evaluate scene understanding using Context Acc. and Traffic Event F1. Context Acc. is the macro-average accuracy over five scene-context dimensions: weather, daytime, visibility, scenario type, and road structure. Traffic Event F1 measures set-level agreement between predicted and annotated traffic events. Presence Acc. is the frame-level binary accuracy of whether at least one decision-critical element is present. Element State Recall and Element Intention Recall are macro-averaged recall scores over normalized behavior categories for matched vehicles and vulnerable road users. Sign-Content Recall evaluates the semantic content of matched traffic signs using macro-averaged recall over the closed-set control labels. R/G Conf. measures the fraction of red and green traffic lights that are confused with each other; lower values are better.

Decision-Critical Element Grounding. The model emits one grounding point for each predicted decision-critical element. Because no bounding box is predicted, conventional IoU-based matching is not applicable. We therefore use a point-to-region distance defined with respect to the ground-truth

segmentation mask. For a predicted point x and ground-truth element g,

$$d(x, g) = \begin{cases} 0, & x \in M_g, \\ \min\limits_{p \in M_g} \|x - p\|_2, & \text{otherwise}, \end{cases} \tag{6}$$

where M_g denotes the segmentation mask of g. If a mask is unavailable, we fall back to the distance to the ground-truth bounding-box boundary rather than its center.

Predicted and ground-truth elements are matched one-to-one using a scale-adaptive assignment threshold

$$\tau_{\text{assign}}(g) = \text{clip}\left(1.5\, d_g,\, 60,\, 300\right), \tag{7}$$

where d_g is the diagonal length of the ground-truth bounding box in pixels. Ground-truth elements are processed in increasing numerical impact rank, with rank 1 matched first, and each element claims the nearest unmatched prediction within τ_{assign}. Matching is type-agnostic so that localization and semantic classification are evaluated independently. Unmatched ground-truth elements are counted as misses, while unmatched predictions are treated as false positives.

Element Recall is defined as

$$R_{\text{det}} = \frac{|\mathcal{M}|}{|\mathcal{G}|}, \tag{8}$$

where $\mathcal{G}$ is the set of ground-truth decision-critical elements and $\mathcal{M}$ is the subset successfully matched to predictions. Rank-1 Recall applies the same criterion only to elements with impact rank 1. Element Type Acc. measures semantic classification accuracy over the matched elements.

Localization precision is evaluated using Point-in-Mask Rate,

$$R_{\text{in}} = \frac{1}{|\mathcal{M}|} \sum_{g \in \mathcal{M}} \mathbf{1}\left[x_{\pi(g)} \in M_g\right], \tag{9}$$

where $\pi(g)$ denotes the prediction matched to g and $x_{\pi(g)}$ is its predicted image point. Unlike bounding-box containment, this metric requires the predicted point to fall on the actual segmented object rather than on background pixels inside its bounding box.

Reasoning. Element-level implications and frame-level rationales are open-ended textual outputs for which lexical-overlap metrics are poorly aligned with semantic correctness. We therefore evaluate them using GPT-5.5 as a judge.

For Element Implication, the judge assesses whether the prediction correctly characterizes the corresponding decision-critical element, including its driving-relevant state or content, and whether it correctly describes the element's influence on ego driving. For Driving Action Rationale, the judge assesses whether the prediction appropriately integrates the relevant scene factors, identifies the dominant factor shaping the driving decision, and correctly explains how that factor influences ego action.

For each prediction, GPT-5.5 returns two binary judgments, $s_i^{\text{cause}}, s_i^{\text{effect}} \in \{0, 1\}$. We average the two judgments for each sample and then average across all valid samples. The Element Implication score is

$$S_{\text{EI}} = \frac{1}{N_{\text{EI}}} \sum_{i=1}^{N_{\text{EI}}} \frac{s_{i,\text{EI}}^{\text{cause}} + s_{i,\text{EI}}^{\text{effect}}}{2}, \tag{10}$$

and the Driving Action Rationale score is

$$S_{\text{DAR}} = \frac{1}{N_{\text{DAR}}} \sum_{i=1}^{N_{\text{DAR}}} \frac{s_{i,\text{DAR}}^{\text{cause}} + s_{i,\text{DAR}}^{\text{effect}}}{2}, \tag{11}$$

where N_{EI} and N_{DAR} denote the numbers of valid judged samples for the two metrics, respectively. The complete GPT-5.5 judge prompts are shown in Fig. 19 and Fig. 20.

Trajectory Prediction. All predicted trajectories are converted to a common 5 s, 4 Hz evaluation grid before scoring. ADE@1s, ADE@3s, and ADE@5s measure the mean Euclidean displacement error between the predicted and ground-truth trajectories over the corresponding prediction horizons. FDE@5s measures the Euclidean error at the final 5 s waypoint. We further report the official Waymo Rater Feedback Score (RFS) using the released WOD-E2E scoring function and its default parameters. RFS Frame averages the per-frame RFS over all evaluation frames, whereas RFS Cluster first averages RFS within each official scenario cluster and then averages across clusters, thereby assigning equal weight to different scenario types. RFS ranges from 0 to 10, with higher values indicating better agreement with human-rated driving trajectories.

D Qualitative Examples

D.1 Failure Cases

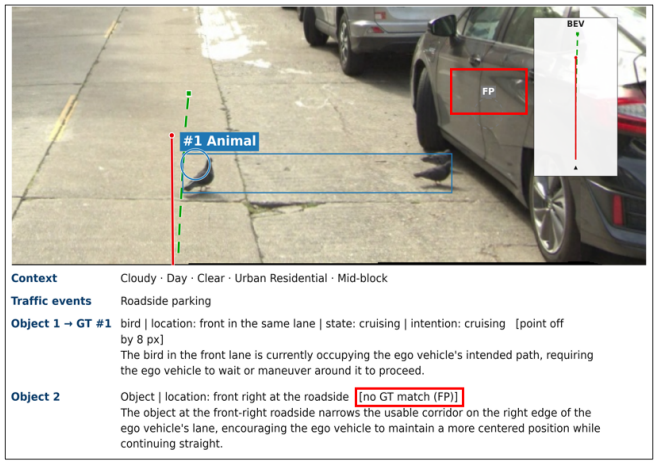
Fig. 13 presents representative failure cases. In Fig. 13(a), the model incorrectly predicts the state and intention of the vehicle at the front-right. This reflects a limitation of single-frame input: motion-related attributes such as state and intention are difficult to infer reliably without explicit temporal visual cues. Fig. 13(b) shows a case where the model identifies a decision-critical element in text but fails to ground it correctly in the image. This suggests that semantic recognition and visual localization are not always fully aligned: the model may recognize the relevance of an element from scene context while failing to establish an accurate spatial correspondence, especially for small, partially occluded, or visually ambiguous elements. Fig. 13(c) shows a more challenging failure in which two human-annotated decision-critical elements are omitted from both the textual prediction and visual grounding. Such disagreement may arise from insufficient geometric supervision. The current annotations do not explicitly provide cues such as object distance, distance change rate, or ego-vehicle dimensions, which are important for determining whether an element constrains the ego vehicle's future driving space. Incorporating these spatial cues may improve decision-critical element selection in future work.

D.2 Successful Cases

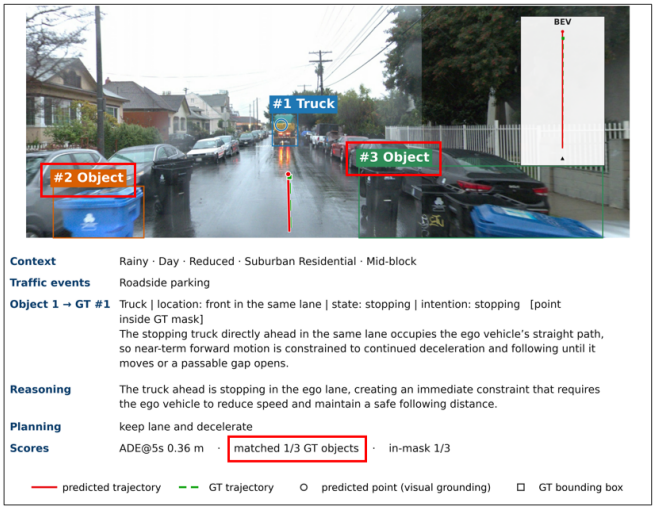
Fig. 14 presents representative successful predictions. In Fig. 14(a), the model correctly identifies and grounds both the parked bus and the oncoming vehicle. It further captures their different effects on ego driving: the bus constrains the available lateral space, while the oncoming vehicle limits the feasibility of moving further left. These element-level implications are consolidated into a coherent driving rationale and action plan. Fig. 14(b) illustrates a more complex multi-object interaction. The model correctly identifies and grounds all three decision-critical elements, including the backing vehicle, construction vehicle, and pedestrian, and interprets how each constrains the ego vehicle's future motion. Their implications are then integrated into the final decision to keep lane and decelerate, producing a trajectory closely aligned with the ground truth. This example shows that VG-CoT can connect multiple visually grounded elements with reasoning and trajectory planning in complex scenes.



(a)

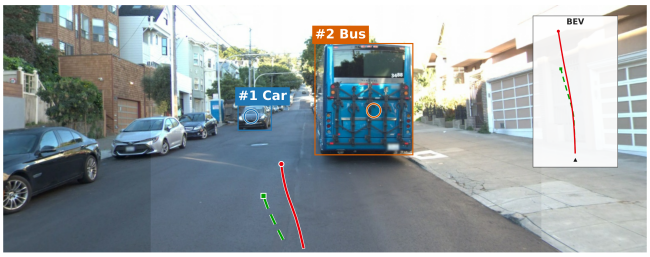


(b)



(c)

Figure 13: Representative failure cases.



Context Sunny · Day · Clear · Urban Residential · Mid-block

Traffic events Roadside parking

Object 1 → GT #2 Bus | location: front at the roadside | state: parking | intention: parking [point inside GT mask]
The parked bus at the front roadside narrows the available lateral space on the right, but since the ego vehicle is already positioned slightly left of center and the path ahead remains open, the ego vehicle should continue straight with a small left offset to maintain clearance.

Object 2 → GT #1 Car | location: front left in the oncoming lane | state: cruising | intention: cruising [point inside GT mask]
The oncoming cruising car at the front-left occupies the opposing lane and creates an oncoming conflict that discourages borrowing left space, so the ego vehicle's near-term motion is to continue straight within its lane while it passes.

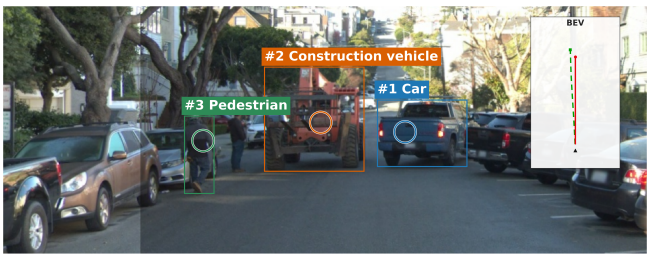
Reasoning The parked bus narrows the right side of the lane, so a small left offset maintains clearance while continuing straight; the oncoming car makes any leftward move unnecessary and less safe.

Planning left nudge and cruise

Scores ADE@5s 2.23 m · matched 2/2 GT objects · in-mask 2/2

predicted trajectory · GT trajectory · predicted point (visual grounding) · GT bounding box

(a)



Context Sunny · Day · Clear · Urban Residential · Mid-block

Traffic events Roadside parking

Object 1 → GT #1 Car | location: front in the same lane | state: backing | intention: parking [point inside GT mask]
The vehicle 1 is backing into a parking spot directly ahead in the ego vehicle's lane, which blocks the forward path and requires the ego vehicle to wait or maneuver around it to proceed.

Object 2 → GT #2 Construction vehicle | location: front left in the same lane | state: stopping | intention: stopping [point inside GT mask]
The vehicle 2 is stopping in the front-left lane and is blocking the ego vehicle's intended straight path, creating a wait-then-go condition until the obstruction is cleared or a safe passing gap opens.

Object 3 → GT #3 Pedestrian | location: front left at the roadside | state: walking along the road | intention: walking along the road [point inside GT mask]
The pedestrian walking along the front-left roadside narrows the usable corridor on the left edge of the ego vehicle's intended path, encouraging the ego vehicle to keep its lane position slightly right while continuing forward.

Reasoning The vehicle directly ahead is backing into the ego's lane, creating an immediate blockage that requires the ego to slow down and wait for the path to clear. The pedestrian on the left further narrows the corridor, reinforcing the need for the ego to reduce speed and be prepared to adjust its path if necessary.

Planning keep lane and decelerate

Scores ADE@5s 0.29 m · matched 3/3 GT objects · in-mask 3/3

predicted trajectory · GT trajectory · predicted point (visual grounding) · GT bounding box

(b)

Figure 14: Representative successful cases.

E PROMPTS

This section provides the full prompts used in our annotation and evaluation pipeline, including scene-context annotation (Fig. 15), vehicle-type classification (Fig. 16), element-level implication generation (Fig. 17), frame-level rationale and action planning (Fig. 18), and the judge prompts for evaluating implications and rationales (Figs. 19 and 20).

System Prompt

You are an expert driving scene analyst specializing in autonomous vehicle perception systems.
Your task is to accurately classify scene attributes from driving images following industry standards
(SAE J3016, ISO 34501, Euro NCAP protocols). Analyze the image carefully and provide structured
classifications based on visual evidence only. Return ONLY valid JSON (no Markdown, no extra text),
matching the requested schema exactly.

User Prompt

Analyze this driving scene image and classify the following attributes. Follow the detailed rules below carefully.

=== WEATHER CLASSIFICATION ===
Options: ["Sunny", "Cloudy", "Rainy", "Snowy", "Fog", null]
- Sunny: ONLY FOR DAYTIME. Clear blue sky, bright sunlight casting shadows, <30% cloud coverage.
- Cloudy: ONLY FOR DAYTIME. Overcast (>70% coverage), diffuse lighting, no precipitation.
- Rainy: Active rainfall, OR wet road with reflection, OR puddles, OR wipers in use. Day or night.
- Snowy: Active snowfall, OR snow accumulation, OR icy patches. Day or night.
- Fog: Visible atmospheric obscuration; takes precedence if visibility is significantly reduced.
- null: Use null for NIGHTTIME when the sky is clear. "Sunny" is NOT valid at night.

=== DAYTIME CLASSIFICATION ===
Options: ["Day", "Night", "Dawn/Dusk"]

=== VISIBILITY CLASSIFICATION ===
Options: ["Clear", "Reduced", "Limited", "Poor"]
- Clear: no significant impairment (cloudy alone does NOT reduce visibility).
- Reduced: night with ABUNDANT artificial lighting.
- Limited: night with SPARSE lighting, OR night with adverse weather.
- Poor: night with MINIMAL lighting, OR heavy adverse weather.
Key principle: Nighttime NEVER has "Clear" visibility.

=== SCENARIO CLASSIFICATION ===
(Urban / Suburban / Rural / Highway / Parking Lot / Residential /
Commercial, plus combinations such as "Urban Residential", "Suburban Highway", ...)

=== ROAD STRUCTURE CLASSIFICATION ===
Options: ["Mid-block", "Intersection", "Roundabout", "Merge", "Diverge"]
MERGE/DIVERGE RULE: these should primarily co-occur with Highway scenarios.

=== OUTPUT FORMAT ===
Return ONLY valid JSON with this exact schema:
{
"weather": {"value": "<option>", "confidence": <0.0-1.0>, "evidence": "<brief visual evidence>"},
"daytime": {"value": "<option>", "confidence": <0.0-1.0>, "evidence": "<brief visual evidence>"},
"visibility": {"value": "<option>", "confidence": <0.0-1.0>, "evidence": "<brief visual evidence>"},
"scenario": {"value": "<option>", "confidence": <0.0-1.0>, "evidence": "<brief visual evidence>"},
"road": {"value": "<option>", "confidence": <0.0-1.0>, "evidence": "<brief visual evidence>"}
}

Figure 15: Prompt used for automatic scene-context annotation.

System Prompt

You are an expert vehicle classifier for autonomous driving systems.
Your task is to identify the type of vehicle shown in the image.
Analyze the vehicle's shape, size, markings, and any distinctive features.
Return ONLY valid JSON (no Markdown, no extra text).

User Prompt

Classify the vehicle type in this image.

Choose EXACTLY ONE from these options:
- "Car": sedans, SUVs, hatchbacks, coupes, minivans, AND pickup trucks
- "Truck": large commercial trucks, semi-trucks, delivery trucks, box trucks (NOT pickup trucks)
- "Emergency vehicle": police cars, ambulances, fire trucks
- "Bus": public transit / coach / shuttle buses (NOT school buses)
- "Construction vehicle": excavators, bulldozers, dump trucks, cement mixers, cranes, road rollers
- "School bus": yellow school buses with "SCHOOL BUS" markings
- "Other": motorcycles, bicycles, ATVs, trailers, or unidentifiable vehicles

IMPORTANT: Pickup trucks (personal trucks with open cargo bed) should be classified as "Car", not "Truck".

Return JSON:
{"vehicle_type": {"value": "<chosen type>", "confidence": <0.0-1.0>, "evidence": "<why>"}}

Figure 16: Prompt used for automatic vehicle-type classification.

Element-Level Driving Implication Prompt

System Prompt

You are an expert in autonomous driving scene reasoning.
Your task is to generate object-level driving implications for the ego vehicle.
For each annotated object, explain what this object means for:
- the ego vehicle's intended path
- the ego vehicle's near-term available motion
- and, if useful, the ego vehicle's object-local maneuver tendency in the current scene

[six aspects] 1 Path occupancy · 2 Drivable space and corridor shape · 3 Right-of-way and timing to proceed · 4 Interaction geometry · 5 Traffic control meaning · 6 Persistent vs temporary effect

Important rules:
- Write exactly ONE concise sentence per object.
- Use the annotated relative position as the primary geometric anchor.
- Do NOT weaken "front" into "side", or "front-right" into "right", unless the image clearly contradicts the annotation.
- Condition the implication on the ego vehicle's current intent and current motion state.
- You may mention a brief object-local motion hint ("can continue cruising", "may need to wait", "may pass after it clears", "requires pulling out", "requires borrowing space to bypass", ...).
- Do NOT output a full frame-level driving plan. Do NOT summarize multiple objects together.
- Do NOT use vague phrases such as "be careful", "is important", "may be risky", "maintain clearance".

State interpretation rules: parking = persistent roadside occupancy (not an active maneuver); stopping = temporary pause; decelerating = still behaviorally active; cruising = continuing motion.

Interaction interpretation rules: e.g. in a narrow shared passage, an oncoming vehicle that is decelerating or stopping may be YIELDING space rather than merely pausing; a front roadside parked object often implies blocked forward progress / pull-out / bypass requirement.

Output JSON: {"object_implications":[{"object_id":"vehicle 1","driving_implication":"..."}]}
Return JSON only. One entry per annotated object. Use the object's class name as object_id.

User Prompt

Scene Context
The scene is {scenario} on a {road} during {daytime} in {weather} weather. Traffic events: {events}.

Ego Vehicle Status
The ego vehicle is currently {LONGITUDINAL} (longitudinal) and {LATERAL} (lateral) at approximately {v} km/h.
The intended behavior is: {INTENT}.

Annotated Objects (see bounding boxes in image)
- [vehicle 1] is a {type}, located at {location}, in state: {state}, with intention: {intention}.
- [vehicle 2] ...

Task
For each annotated object above, generate a driving_implication explaining what it means for the ego vehicle.
Output JSON with object_implications array. Use the object class name as object_id.

Figure 17: Prompt used for generating element-level driving implications.

Reasoning and Action Planning Prompt

System Prompt

You are an expert in autonomous driving scene reasoning.
Your task is to generate a frame-level final_plan and a concise reason for the ego vehicle.
You will be given: an image with annotated important objects, ego state, ego intent,
object-level driving implications, and the impact rank for each object.

final_plan: short-horizon motion plan, one short phrase; may contain two sequential sub-actions when
the near-term motion naturally has two stages ("wait, then continue through the guided corridor").
Good: "keep lane and cruise" / "creep forward and prepare to stop" / "pull out left, then follow the lead vehicle".
Bad: a long paragraph, repeating implications, listing alternatives, or "keep lane" without a longitudinal term.

reason: explains WHY that plan is organized this way — the dominant constraint/opportunity and, if
useful, one enabling factor. Must NOT restate every implication, must NOT mention impact rank, must
NOT describe objects that do not change the plan, must NOT use "because it is safer".

Lateral vocabulary: left/right nudge · keep lane · keep left/right · borrow left/right ·
lane change left/right · pull out left/right (choose the SMALLEST motion term that fits)
Longitudinal vocabulary: cruise · accelerate · decelerate · creep forward · stop/wait · follow
IMPORTANT: "keep lane" MUST be combined with a longitudinal term.

Use impact rank only internally as an attention cue; never mention rank numbers.

Return JSON only: {"final_plan": "...", "reason": "..."}

User Prompt

Ego vehicle:
- Speed: {v} km/h
- Current behavior: longitudinal={LON}, lateral={LAT}
- Intent: {INTENT}

Each annotated object in the image has the following driving_implication and impact_rank (1 = highest):
- [vehicle 1] (impact_rank: 1): {implication}
- [traffic control 1] (impact_rank: 2): {implication}

Synthesize these into reason and final_plan. Return JSON with "reason" and "final_plan" keys.

Figure 18: Prompt used for frame-level rationale and action planning.

System Prompt

You evaluate an autonomous-driving model's explanation of ONE key object. Every REFERENCE explanation has two parts: (A) the CAUSE = what the object is and its driving-relevant state/position; (B) the EFFECT = how it constrains or releases the ego vehicle's motion. Compare the MODEL explanation to the REFERENCE on BOTH axes. Be LENIENT about wording: accept paraphrase, a coarser or a more detailed description, a different sentence order, and omitted non-essential modifiers. Score 0 ONLY when the prediction refers to a DIFFERENT object/condition than the reference, or asserts the OPPOSITE direction of constraint (e.g. 'releases the path' vs 'blocks the path'). Judge the two axes independently -- a wrong CAUSE does not force EFFECT to 0, and vice versa. Reply with EXACTLY two digits separated by a comma: cause,effect (e.g. '1,0'). No other text.

User Prompt

OBJECT: {GT type + GT location}

REFERENCE: {GT implication}

MODEL: {predicted implication}

Score cause,effect as two digits.

Figure 19: Prompt used for implication judge.

System Prompt

You evaluate an autonomous-driving model's one-sentence REASON for its driving plan. The REFERENCE reason names (A) the DOMINANT factor that shapes the ego vehicle's motion -- the object or condition that blocks, narrows, opens, guides or yields its path -- and (B) the EFFECT that factor has on the ego's motion. Score the MODEL reason on both axes: cause = it identifies the same dominant factor (pointing at the same object or the same kind of condition is enough; it need not match the level of detail); effect = the direction of the influence agrees. Be LENIENT about wording: accept paraphrase, a coarser or a more detailed description, a different sentence order, and omitted non-essential modifiers. Score 0 ONLY when the prediction refers to a DIFFERENT object/condition than the reference, or asserts the OPPOSITE direction of constraint (e.g. 'releases the path' vs 'blocks the path'). Judge the two axes independently -- a wrong CAUSE does not force EFFECT to 0, and vice versa. Reply with EXACTLY two digits separated by a comma: cause,effect (e.g. '1,0'). No other text.

User Prompt

OBJECTS IN THE SCENE (context only): {GT object list in the frame, used only as context and not scored}

REFERENCE: {GT reason}

MODEL: {predicted reason}

Score cause,effect as two digits.

Figure 20: Prompt used for rationale judge.