SEEING WITH TOOLS, REASONING WITH LANGUAGE: RECASTING VISION AS STRUCTURED SCENE TEXT FOR DRIVING VQA

Anonymous authors
Paper under double-blind review

ABSTRACT

Driving visual question answering (VQA) requires extracting relevant visual information from images or videos and reasoning over it. Current vision-language models (VLMs) couple these tasks, obscuring whether their errors arise from visual information extraction or downstream reasoning. We investigate this distinction with DriveTLM, a training-free framework that delegates visual information extraction to *visual processing tools*, combining off-the-shelf vision models with deterministic algorithms. Their outputs are organized into structured scene text, an explicit language interface that lets the answering model reason without visual tokens. The framework supports images and videos, Rule-based or Agentic tool selection, and both VLM and LLM answer backbones without parameter updates. Across image benchmark DriveBench and video benchmark LingoQA, DriveTLM improves over direct visual input for multiple model families and sizes, with particularly strong gains for smaller backbones and competitive end-to-end latency. Controlled studies examine these gains: most persist without hand-written task rules such as risk inference and answer-option matching; expressing the same information as text is more effective than visual overlays; and selecting relevant information matters more than including every available detail. The model also responds selectively when individual facts are changed. These findings show the value of structured scene text as an interface for delivering tool-derived evidence to frozen answering models. The code will be released upon publication.

1 INTRODUCTION

Understanding traffic scenes requires both accurate visual perception and effective reasoning. A model must first identify relevant objects, spatial geometry, and motion patterns, and then reason about their implications for driving decisions. Driving visual question answering (VQA) provides a targeted way to evaluate these capabilities using real-world images and videos (Qian et al., 2024; Xie et al., 2025; Marcu et al., 2024). By asking questions about spatial relationships, interactions among dynamic agents, and traffic rules, driving VQA can reveal specific strengths and weaknesses that may be obscured by aggregate driving metrics. For autonomous driving systems, it offers an interpretable diagnostic of scene understanding; for general-purpose Vision-Language Models (VLMs) and Vision-Language-Action models (VLAs), it further evaluates whether they can reason about complex, dynamic physical environments (Fu et al., 2024; Tian et al., 2024b; Xu et al., 2024). Driving VQA therefore serves as a broadly applicable test of whether a model truly understands consequential outdoor scenes rather than merely producing plausible descriptions. This raises a fundamental question: when models fail on driving VQA, do failures stem from an inability to extract the necessary visual evidence, or from limitations in reasoning once that evidence is available?

Existing methods largely differ in how visual information is exposed to the answering VLM, yet most still leave visual information extraction on the model's critical path. As shown in Fig. 1, traditional VLMs encode images or videos into visual tokens and rely on the language model to jointly perform perception and reasoning through multimodal attention (Sima et al., 2024). This design faces particular challenges in driving scenes, where benchmarks show that VLMs can rely on language priors, overlook small or distant agents, and struggle with spatial relations, orientation, distance, and motion cues that are essential for downstream decisions (Xie et al., 2025; Guo et al.,

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.

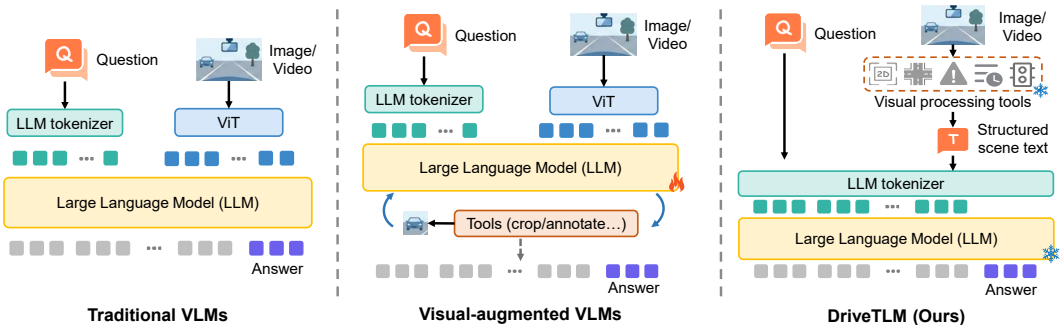

Figure 1: **How visual information reaches the answering model.** *Left:* a traditional VLM extracts visual information and reasons through image or video tokens. *Middle:* agents returning crops or annotations retain visual tokens and add intermediate views to the inference context. *Right:* DriveTLM delegates visual extraction to processing tools and supplies their outputs as structured scene text; the answering model receives text alone.

2024). Fine-tuning on curated driving data can alleviate some of these limitations (Shao et al., 2024; Li et al., 2026a), but requires task-specific annotation and parameter updates, with improvements that do not readily transfer across backbones. Alternatively, visual-augmented methods such as V* and Visual Sketchpad improve access to visual evidence through crops, zooms, annotations, or intermediate views (Wu & Xie, 2024; Hu et al., 2024); however, these methods still require the VLM to extract information from visual tokens. These approaches may make evidence easier to inspect, rather than eliminating the need for visual interpretation, but they retain both the model's perceptual burden and, in iterative settings, the growing token cost of processing multiple visual views.

Therefore, we ask: *can external scene evidence improve an unchanged language model, and how much do deterministic task rules contribute?* We introduce **DriveTLM**, which uses *visual processing tools* as the sole interface to images and videos (right of Fig. 1). This tool library combines off-the-shelf vision models with deterministic image processing and geometric algorithms. It extracts visual information about objects, roads, relations, and motion, which a deterministic textualizer organizes into structured scene text. The text separates tool-derived scene evidence from task interpretations produced by deterministic rules. Because its records are typed and attributable, the interface supports controlled comparisons that hold the answering model fixed while varying evidence content, representation, selection, or interpretation. These comparisons help distinguish failures in extraction and deterministic processing from failures in downstream answering. The answering model only receives structured scene text and the question. All model parameters remain unchanged, and DriveTLM supports both image input (**DriveTLM-Frame**) and video input (**DriveTLM-Video**), with **Rule-based** or **Agentic** tool selection based on the question.

Our results show the value of explicit scene evidence for frozen answering models. Across the image-based DriveBench and video-based LingoQA benchmarks, DriveTLM generally improves over direct visual input across model families and scales, with particularly strong gains for smaller backbones, while requiring no fine-tuning of the answering model and remaining efficient and competitive with driving-specific systems. By representing extracted visual information as structured scene text, DriveTLM also enables lightweight text-only LLMs to answer driving questions effectively, demonstrating compatibility with answer backbones that do not consume visual tokens. More importantly, controlled comparisons show that most of the gain persists without hand-written task rules such as risk inference and answer-option matching, that structured text is more effective than visual overlays for the same evidence, and that predictions respond selectively to task-relevant scene facts. These results motivate improving access to scene evidence. We make four contributions:

- We introduce DriveTLM, a training-free framework that delegates visual information extraction to visual processing tools and supplies their outputs as structured scene text, removing visual tokens from the answering model.
- We extend this interface from images to videos through temporal tools to support different input formats and develop rule-based and agentic tool selection strategies for different answer backbones.
- We demonstrate improvements over direct visual input across model families and sizes on DriveBench and LingoQA, with strong gains for smaller models and competitive overall latency.
- We separate scene evidence from task rules, finding that most of the gain persists without task-level rules, and analyze how textual representation and question-relevant selection affect answers.

2 Related Work

Tool-augmented and agentic VLMs. External tools extend language models with specialized operations (Schick et al., 2023; Qin et al., 2024; Shen et al., 2023; Lu et al., 2023), while visual programming and multimodal agents compose vision modules or tool outputs for reasoning (Gupta & Kembhavi, 2023; Surís et al., 2023; Yang et al., 2023b). Related methods such as SoM, V*, and Visual Sketchpad manipulate visual inputs through region markers, crops, or intermediate visual artifacts (Yang et al., 2023a; Wu & Xie, 2024; Hu et al., 2024); the answering VLM still interprets visual tokens, often with expanded visual context. Driving agents similarly combine perception tools with language reasoning (Qian et al., 2025; Zheng et al., 2026). DriveTLM instead recasts visual-tool outputs into structured scene text, enabling controlled study of how visual information extraction and representation affect downstream answers.

VLMs for autonomous driving. Driving VLMs combine scene understanding, reasoning, and planning (Tian et al., 2024b; Mao et al., 2023; Nie et al., 2024). Many acquire domain competence by fine-tuning on driving data (Marcu et al., 2024; Shao et al., 2024; Li et al., 2026a), sometimes with reinforcement learning (Jiang et al., 2025; Li et al., 2026b) or generated reasoning traces (Chahe & Zhou, 2025); efficiency-oriented variants shrink the model or compress visual tokens (Gopalkrishnan et al., 2024; Cai et al., 2026), and hierarchical QA decomposes questions for a fine-tuned VLM (Mohamud et al., 2025). Other approaches already supply language models with textual descriptions from upstream perception (Mao et al., 2024; Cui et al., 2024; Sha et al., 2023; Mei et al., 2024; Choudhary et al., 2024), or use object-level representations (Chen et al., 2024; Tian et al., 2024a). Building on this line, DriveTLM contributes a question-conditioned tool-to-text interface for images and videos with typed, attributable records, evaluates it across frozen answering backbones, and uses it for controlled studies that separate extraction, representation, selection, and task rules.

Driving VQA. Driving VQA evaluates specific capabilities through questions about traffic scenes. Image benchmarks probe objects, relations, driving decisions, and entity-grounded reasoning (Qian et al., 2024; Sima et al., 2024; Guo et al., 2024; Xie et al., 2025; Corbière et al., 2026). Video benchmarks add questions about motion, interactions, and spatiotemporal relations over time (Zeng et al., 2025; Kim et al., 2025; Yu et al., 2025; Vo et al., 2026; Ishihara et al., 2026). We use both modalities to test whether external visual information extraction improves an unchanged answering model, and controlled studies to determine how structured scene text enables those gains.

3 Methodology

3.1 Preliminaries

Definition of structured scene text. Structured scene text represents extracted visual information as typed, human-readable records organized by semantic headers such as `[Objects]`, `[Road Layout]`, and `[Video Evidence]`. Each current-frame detection receives a frame-local, camera-scoped identifier (e.g., `cam_front_002`) with its category, camera, normalized 2D box, relative position, lane region, and available depth and heading; associated tool outputs, relations, and risks reuse the same identifier. For video, temporal tools associate detections across adjacent frames and assign clip-local track identifiers (e.g., `track_01`), while retaining the current-frame object ID to link trajectories back to `[Objects]`. These identifiers serve only as internal reference keys. Low-confidence detections are discarded; unsupported tool fields remain `unknown`, whereas dedicated task sections derive heuristic interpretations from the retained records. Fig. 3 illustrates the syntax.

Formulation. Let $V = \{I_t^c\}$ denote a driving observation, where t indexes time and c indexes the available cameras, and let q be a question about that observation. We equip the system with a library of visual processing tools $\mathcal{T} = \{\tau_1, \ldots, \tau_K\}$. A router R chooses an active tool set $\mathcal{A}_q \subseteq \mathcal{T}$, whose tools produce observations $Z_q = \{(k, z_k) : z_k = \tau_k(V), \tau_k \in \mathcal{A}_q\}$. A deterministic textualizer converts Z_q and the question into structured scene text with two parts, scene evidence and task rules:

$$\mathcal{A}_q = R(q, \mathcal{T}), \qquad s = \Phi_{\text{ev}}(Z_q, q) \oplus \Phi_{\text{task}}(Z_q, q), \qquad \hat{y} = M(s, q). \tag{1}$$

Here M receives only s and q. Statements in s form five levels: level (i) restates tool outputs, and each higher level is derived from lower ones. Φ_{ev} writes the scene evidence, using q only to select sections: (i) tool estimates (e.g., box, depth), (ii) geometric transforms (e.g., depth order), and (iii)

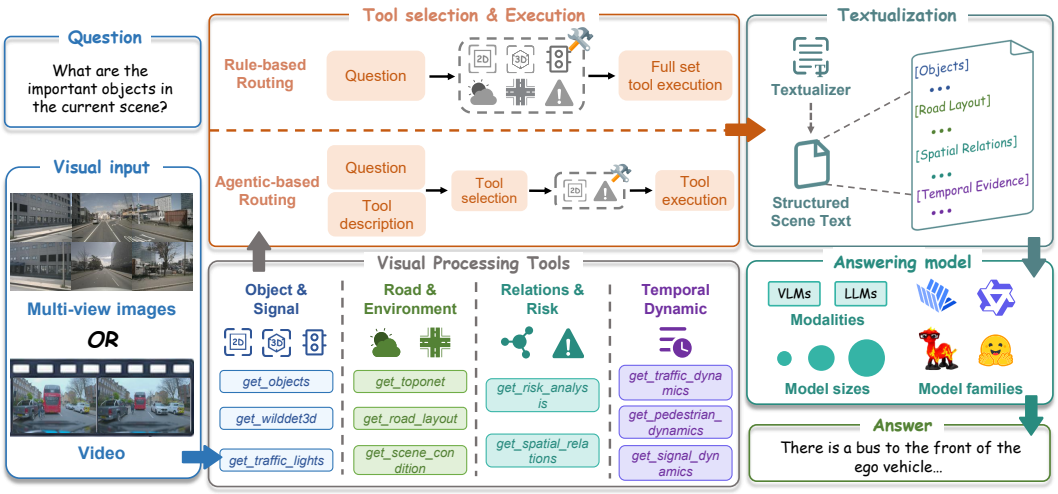

Figure 2: **Overview of DriveTLM.** Visual processing tools extract information from multiview images or videos, covering objects and signals, roads and environment, relations and risks, and temporal dynamics. Rule-based routing executes the full tool pool, whereas Agentic routing selects tools using the question and tool descriptions. A shared textualizer converts tool outputs into structured scene text. The answering model uses only this text and the question, without visual tokens. This interface supports VLMs and LLMs across model sizes and families.

semantic relations (e.g., ahead of the ego vehicle). Φ_{task} adds task rules that interpret this evidence for the question: (iv) ego-state and risk inferences (e.g., likely ego speed, potential conflicts) and (v) answer-oriented statements (e.g., moving-status labels, option matching). Deleting their dedicated sections, such as `[Risk Analysis]`, removes only the task rules (Appendix A.3).

3.2 Overview of DriveTLM

As illustrated in Fig. 2, DriveTLM consists of four stages: tool selection, tool execution, textualization, and language reasoning. Frame tools process the available camera views, while temporal tools align historical observations and summarize motion and state changes. Current presence, count, and signal color are derived from the latest frame. The resulting structured scene text and question are then passed to the answering model in a single generation call.

3.3 Visual Processing Tool Design

Category definition. Visual processing tools combine off-the-shelf vision models with deterministic algorithms for image processing, geometry, and temporal association. Each tool is an independently callable semantic interface; several tools may reuse the same model output. The underlying models are used as released; we assess each tool by its contribution to answers and the pipeline's stability when tool outputs change under threshold shifts or visual corruption (Appendices D.4 and C.5). The library has eight frame and three temporal tools, grouped by the information they provide. The frame categories describe objects and signals, roads and environment, and relations and risks; temporal tools add changes over time. Fig. 3 shows representative contracts; Appendix A gives their implementations.

Object and signal perception tools. `get_objects` uses Faster R-CNN (Ren et al., 2015) to detect traffic participants and control devices in the available camera views. Each detection records its category, confidence, camera identity, and normalized 2D box, from which coarse position and lane region are derived. `get_wilddet3d` associates projected WildDet3D predictions (Huang et al., 2026) with these detections to add finer categories, metric depth, and 3D orientation. `get_traffic_lights` reuses detected light and sign regions, estimates signal state from crop colors, and records its location and relevance to the ego vehicle. Simple geometric transforms of each 2D box also supply coarse depth order, complementing the available metric depth estimates. These interfaces share object records while exposing complementary identity, geometry, and control information, so the answering model can relate attributes to the same observed object.

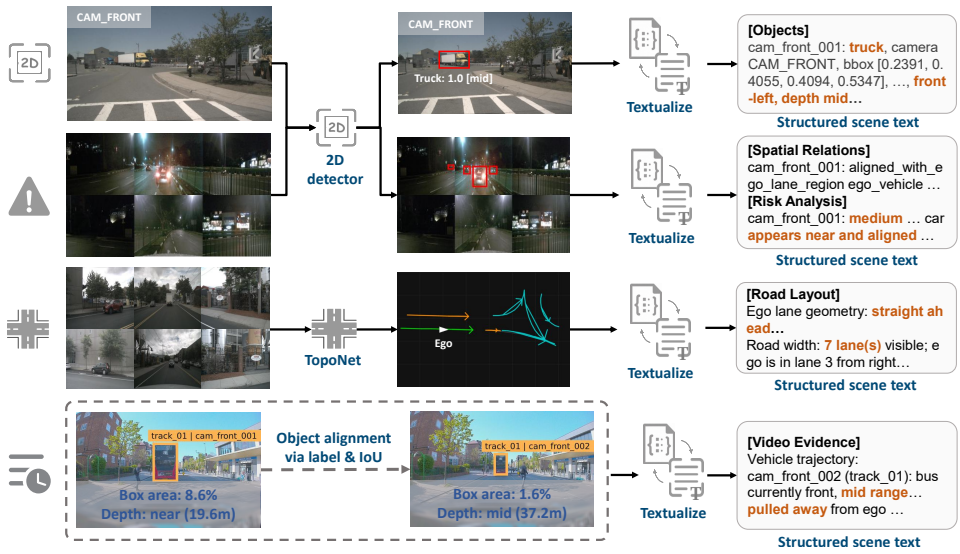

Figure 3: **Representative tool contracts and structured scene text syntax.** Each row traces a visual input through a tool and the textualizer. The examples cover all categories: object perception, road understanding, derived risk analysis, and temporal dynamics. Orange text marks the semantically important facts inserted into each fixed section template.

Road and environment understanding tools. `get_scene_condition` summarizes weather, illumination, visibility, road surface, and time of day from the front view, combining image statistics with optional CLIP scene classification (Radford et al., 2021). `get_road_layout` extracts visible lane boundaries and cues for crosswalks, intersections, and stop lines through image processing, abstaining when markings are weak. `get_toponet` uses TopoNet (Li et al., 2023) on calibrated camera views to predict lane geometry and connectivity. The textualizer converts this output into lane count and relative order, ego lane geometry, curvature, and a coarse ego direction. The local and multiview interfaces remain separate because they provide different levels of road information: a lightweight estimate of visible markings and a richer description of lane topology.

Relational and risk analysis tools. `get_spatial_relations` and `get_risk_analysis` apply deterministic rules to the shared object records without additional visual encoders. The former combines camera identity, box location, lane assignment, and depth to express relations such as whether an object lies ahead of the ego vehicle or aligns with its lane. The latter identifies potential conflicts, including nearby vulnerable road users and vehicles that may block the ego path, and reports the responsible object, risk type, reason, and confidence. Relations form level (iii) and risk statements level (iv) of Eq. 1; both are rule-derived from estimated states, and source identifiers preserve attribution to detections.

Temporal dynamics tools. The temporal tools associate consecutive detections by category, box overlap, and center displacement to form persistent tracks. `get_traffic_dynamics` combines changes in box scale, image trajectory, lane region, and available depth to describe whether a vehicle's relative gap opens, closes, or remains stable, together with reliable lateral motion and pose direction. `get_pedestrian_dynamics` uses lateral displacement and lane transitions to distinguish active crossing from static presence. `get_signal_dynamics` reports a signal transition only when reliable observations near the end of the clip disagree. These tools link historical changes to current objects when applicable, allowing temporal questions to be answered from explicit motion information. Relative trends are never interpreted as absolute speed, and current presence, count, and signal color remain anchored to the latest frame.

3.4 Tool Selection Methods

Rule-based tool selection. The Rule-based variant executes the full frame tool pool independently of the question, allowing its outputs to be reused across questions about the same observation. A deterministic renderer retains the sections appropriate to the question type; for example, moving-status questions keep object and relation records but omit the ego and risk sections. When a question

Table 1: **Evaluations on DriveBench.** GPT-judge score, MCQ accuracy, and ROUGE-L on the clean set. **Bold**: best input setting within each backbone block, ties included; shaded rows : ours. **Avg.**: unweighted mean of the eight task metrics (latency excluded), a summary of the task columns. †: reproduced from a released checkpoint; other baseline entries are published results.

Model	Size	Type	Input	Perception		Prediction		Planning		Behavior		Lat. (s) ↓	Avg.
				GPT	MCQ	GPT	ROUGE-L	GPT	ROUGE-L	GPT	MCQ		
Human	-	-	Image	47.7	93.3	-	-	-	-	69.5	66.7	-	-
GPT-4o	-	Zero-shot	Image	35.4	59.0	51.3	19.7	75.8	6.5	45.4	25.5	-	39.8
DriveLM	7 B	Fine-tuned	Image	16.9	40.0	44.3	40.0	68.7	53.1	42.8	44.0	-	43.7
Dolphins	7 B		Image	9.6	7.0	32.7	25.2	52.9	12.9	8.8	0.5	-	18.7
ReCogDrive	8 B		Image	65.0	55.0†	49.3	24.6†	70.2	9.0†	42.4	36.5†	-	44.0
DriveAgent-R1	3 B		Image	34.1	-	32.9	-	61.9	-	43.7	-	-	-
Qwen2.5-VL	3 B	Zero-shot	Image	33.3	57.5	19.2	15.1	28.5	8.5	27.7	38.5	2.0	28.5
			Caption	34.7	49.5	27.8	25.3	55.1	**10.8**	39.0	38.0	**1.6**	35.0
			Rule-based (Ours)	47.8	62.5	46.8	36.7	**77.1**	8.0	**60.3**	52.0	3.8	48.9
			Agentic (Ours)	**51.2**	**65.0**	**48.3**	**38.6**	76.8	10.0	58.9	**53.0**	1.7	**50.2**
LLaVA-1.6	7 B	Zero-shot	Image	**48.7**	52.0	35.2	18.3	61.9	5.5	50.6	41.5	6.2	39.2
			Caption	39.0	49.5	33.5	20.6	57.8	7.6	36.4	36.0	9.0	35.0
			Rule-based (Ours)	46.0	**66.0**	**35.5**	**37.7**	**68.4**	**8.6**	**52.5**	**50.5**	**5.8**	**45.7**
InternVL3	38 B	Zero-shot	Image	58.7	59.0	48.5	20.5	82.8	6.5	58.7	41.0	8.0	47.0
			Caption	**60.3**	54.0	42.2	19.6	78.3	5.5	56.9	40.5	13.8	44.7
			Rule-based (Ours)	59.2	59.0	**54.8**	24.2	83.8	6.7	**67.2**	**52.5**	7.7	**50.9**
			Agentic (Ours)	58.7	**64.0**	52.8	**29.2**	**85.7**	**7.2**	60.8	49.0	**5.3**	**50.9**
Qwen3-LLM	0.6 B	Zero-shot	**Rule-based (Ours)**	**41.7**	63.0	34.2	26.7	57.7	**12.8**	45.5	47.0	3.1	41.1
			Agentic (Ours)	38.8	**64.5**	**36.5**	**35.2**	**65.9**	12.0	**49.0**	**49.5**	**1.5**	**43.9**

references an object by camera and image coordinates, the renderer binds it to the nearest detected box in that camera. For video, a fixed temporal pool processes aligned frame records before rendering. This strategy prioritizes information coverage without requiring the answering model to select tools.

Agentic tool selection. The Agentic variant first asks the answering model to select tools from their descriptions using the question and a schema-constrained list of tool names. A resolver adds required dependencies, such as `get_objects` for the WildDet3D, relation, and risk tools, and executes only the resulting set in a fixed order. Their outputs use the same textualizer as Rule-based selection, isolating tool selection from representation, and skipping unused tools reduces both latency and irrelevant input. Both strategies share question-conditioned retention and interpretation rules; only Agentic selection also uses the question to control tool execution. This design makes the trade-off explicit: Rule-based selection maximizes reusable evidence coverage, whereas Agentic selection adds one tool-name-only model call to avoid unnecessary tool execution.

4 Experiments

Implementation details. We evaluate on DriveBench (Xie et al., 2025) and the LingoQA validation split (Marcu et al., 2024), reporting their official GPT-judge and Lingo-Judge respectively. Since judge scores need not track exact answer correctness, we provide MCQ accuracy for option selection and ROUGE-L, BLEU-4 for reference overlap in open-ended answers. Answering backbones are frozen Qwen2.5-VL (Bai et al., 2025), LLaVA-1.6 (Liu et al., 2024), InternVL3 (Zhu et al., 2025), and the text-only Qwen3 (Yang et al., 2025); baselines are GPT-4o/4V (Achiam et al., 2023) and fine-tuned driving models DriveLM (Sima et al., 2024), Dolphins (Ma et al., 2023), ReCogDrive (Li et al., 2026a), DriveAgent-R1 (Zheng et al., 2026), and LingoQA (Marcu et al., 2024). Within each backbone, Image feeds the source views directly, and Caption replaces them with a caption generated by the same backbone. Runs use H100 GPUs, batch size one, greedy decoding, and vLLM (Kwon et al., 2023); input methods share the same backend and source resolution per backbone. Latency includes input construction and answer inference, averaged over the full split, excluding initialization and offline scoring. Appendix B details the protocol, complementary metrics, and MCQ extractor.

Configuration development. We develop the DriveTLM configuration, including tool settings, thresholds, textual templates, driving-level rule parameters, and the agentic routing prompt, on a development set of 200 questions from 100 DriveLM-nuScenes v1.1 training keyframes (Sima et al., 2024) in 100 scenes, with no scene overlap with DriveBench (Appendix B.1). The configuration is fixed before benchmark evaluation and uses no DriveBench or LingoQA labels. Both benchmarks use the same tool library and parameters, shared by rule-based and agentic selection; LingoQA adds only the temporal tools, set heuristically without LingoQA data. Zero-shot in the tables denotes answering backbones without driving-specific fine-tuning.

Table 2: **Evaluations on LingoQA.** Frames is the number of source frames. Within each backbone block, **bold** denotes the best evaluated setting for each metric, with lower latency preferred. Shaded rows are ours.

Model	Size	Type	Frames	Input	Lingo-J ↑	BLEU-4 ↑	METEOR ↑	CIDEr ↑	Latency (s) ↓
Human	-	-	5	Raw frames	96.6	81.0	52.9	361.8	-
LingoQA	-	Fine-tuned	5	Raw frames	60.8	15.0	18.6	65.6	-
GPT-4V	-	Zero-shot	5	Raw frames	59.6	6.3	12.3	42.8	-
InternVL3	2 B	Zero-shot	5	Raw frames	42.6	5.0	**15.0**	30.8	**0.5**
			5	Caption	44.8	3.2	13.9	17.5	1.0
			1	**Rule-based (Ours)**	43.0	5.7	13.4	36.4	1.0
			1	**Agentic (Ours)**	43.0	**6.1**	13.1	36.2	0.7
			5	**Rule-based (Ours)**	**52.8**	5.4	14.2	35.3	1.0
			5	**Agentic (Ours)**	51.2	5.9	14.3	**40.1**	1.0
Qwen2.5-VL	3 B	Zero-shot	5	Raw frames	45.8	5.0	14.6	25.2	**0.5**
			5	Caption	45.0	3.8	14.2	32.7	1.3
			1	**Rule-based (Ours)**	49.0	8.0	**14.8**	45.7	0.9
			1	**Agentic (Ours)**	48.6	7.4	14.2	41.9	**0.5**
			5	**Rule-based (Ours)**	**55.2**	7.7	14.3	44.7	1.0
			5	**Agentic (Ours)**	54.2	**8.1**	14.4	**46.0**	1.0
LLaVA-1.6	7 B	Zero-shot	5	Raw frames	48.6	1.7	12.5	8.3	5.4
			5	Caption	41.2	2.2	13.4	11.2	11.8
			1	**Rule-based (Ours)**	45.2	4.8	14.6	**38.5**	**1.7**
			5	**Rule-based (Ours)**	**52.8**	**5.0**	**14.8**	38.1	2.0

Table 3: **Evidence construction stages.** In (a), S1–S3 jointly change rendering, selection, and identifiers. S3 is DriveTLM with the Φ_{task} sections deleted, keeping levels (i)–(iii) of Eq. 1; the last row of (a) thus isolates task rules. bAcc: balanced accuracy. Appendix D.1 splits this step. Shaded rows are the complete DriveTLM.

(a) DriveBench: frame evidence

Stage	Evidence	Perc. bAcc	Pred. GPT	Plan. GPT	Beh. bAcc
–	Question only	32.5	0.0	45.7	28.4
–	Raw image	34.0	19.2	28.5	40.0
S1	Numeric tool outputs	48.1	42.8	61.5	35.1
S2	S1 + language and pruning	47.1	42.5	72.3	51.2
S3	S2 + identifiers/heading	53.4	42.9	69.6	49.6
DriveTLM	S3 + task rules (iv)–(v)	**60.1**	**46.8**	**77.1**	**52.4**

(b) LingoQA: video evidence

Stage	Evidence	Lingo-J
–	Question only	27.2
–	Raw video	45.8
S1	Frame scene text	47.8
DriveTLM-Frame	S1 + task rules (iv)–(v)	49.0
S2	S1 + temporal scene text	51.2
DriveTLM-Video	S2 + task rules (iv)–(v)	**55.2**

4.1 Can Structured Scene Text Replace the Image?

Results on DriveBench. Tab. 1 tests whether structured scene text improves an unchanged answering model. At 3B, Agentic improves all eight task metrics over Image and raises the table average from 28.5 to 50.2, whereas Caption reaches only 35.0 and lowers both MCQ accuracies. This contrast shows the benefit of exposing task-relevant scene evidence rather than replacing pixels with generic text. Across the three backbones, Behavior MCQ gains are significant (+14.5, +9.0, and +11.5), while the paired per-question gains are largest at 3B (+21.6, versus +6.1 and +3.7 at 7B and 38B; Appendix C.1). Structured evidence offers its largest measured benefit to the smaller backbone and remains effective at larger scales. Rule-based and Agentic reach similar accuracy, but Agentic lowers latency at both evaluated DriveBench scales through selective execution (Appendix C.2). Sec. 4.2 separates scene-evidence and task-rule contributions.

Results on LingoQA. Tab. 2 separates frame evidence from temporal evidence. Relative to raw five-frame video, the one-frame Rule-based variant changes Lingo-J by only +0.4, +3.2, and −3.4 points at 2B, 3B, and 7B. Temporal tools add a further +9.8, +6.2, and +7.6 points to the same frame pipeline, yielding net gains of +10.2, +9.4, and +4.2 over raw video. These gaps show that temporal records, not final-frame reformulation alone, drive the video gains. At 3B, the temporal gain is significant (+6.2, 95% CI [2.4, 10.0]). Raw video shares the pipeline's response schema, so the gain over it is not an answer-format effect and peaks on gap/action and pedestrian questions. Temporal evidence alone scores 41.4, versus 51.2 when combined with frame evidence, showing that motion cues work best when anchored to the current scene. Appendix C.6 reports the full per-type analysis with sample counts and paired confidence intervals.

Answering with an LLM. Agentic DriveTLM lets text-only Qwen3-0.6B exceed raw-image Qwen2.5-VL-3B on all eight metrics, but not its structured scene text variant (Tab. 1); Appendix C.3 reports broader transfer across 11 VLMs and four text-only LLMs.

Efficiency comparison. End-to-end, including tools, DriveTLM-Video takes 1.0 s at 3B (raw/caption: 0.5/1.3 s) and 2.0 s on LLaVA-1.6-7B (5.4/11.8 s), where it is fastest. Removing visual tokens can therefore offset tool cost for larger backbones, but does not always reduce latency (Tab. 2).

Takeaway 1: Tool-derived scene text improves frozen VLMs across families, scales, and image or video inputs, most for small backbones, and lets text-only LLMs answer driving questions.

4.2 Where Does the Gain Come From?

Full-system gains combine extraction, representation, and task rules. We examine components with Qwen2.5-VL-3B and rule-based selection on clean DriveBench unless noted. Perception and Behavior report balanced accuracy (Appendix B.4), while Prediction and Planning use GPT score.

Scene evidence versus task rules. Tab. 3 measures how much gain remains without task rules. S3 keeps the scene description (i)–(iii) without risk inference, option matching, or answer directives, yet exceeds raw images by 19.4 Perception bAcc, 23.7 Prediction GPT, 41.1 Planning GPT, and 9.6 Behavior bAcc. Task rules add a further 6.7, 3.9, 7.5, and 2.8, a minority of the gain on every task. On Perception, the remainder interprets existing records rather than adding facts: presenting the queried object as a moving-status label with option references instead of raw 3D measurements raises bAcc by 9.2 (S3c→S3d, Appendix D.1). For video, temporal scene evidence adds 3.4 Lingo-J points without task-specific interpretations, which contribute a further 4.0 points. The question-only Planning score reflects generic advice rewarded without evidence.

Free-form caption vs. structured scene text. Text input alone does not explain the gains. At 3B, captions raise the average from 28.5 to 35.0 but reduce both MCQ accuracies; rule-based and agentic structured scene text reach 48.9 and 50.2 while improving both (Tab. 1).

Table 4: **Evidence channel with fixed tool outputs.** Annotated images draw the tool outputs as region marks (*e.g.*, boxes, categories, lanes); tool facts express the same outputs as text (S2 in Tab. 3). Perception and Behavior report balanced accuracy; Prediction and Planning report GPT Score. *Avg.* is the unweighted mean.

Interface	Perc. bAcc	Pred. GPT	Plan. GPT	Beh. bAcc	Avg.
Raw images	33.97	19.18	28.46	39.96	30.39
Tool-annotated images	25.90	4.51	34.89	39.58	26.22
Selected tool facts as text	47.10	42.52	72.29	51.24	53.29
Annotated images + tool text	33.05	42.05	48.73	48.95	43.20
Raw images + tool text	23.66	34.13	52.51	49.28	39.90
DriveTLM	**60.14**	**46.79**	**77.14**	**52.42**	**59.12**

Visual overlay vs. structured scene text. Tab. 4 holds tool outputs fixed and varies how they reach the model. Following the region-marking practice of SoM and Visual Sketchpad (Yang et al., 2023a; Hu et al., 2024), annotated images draw these outputs on the frames; expressing them as text instead averages 53.29, exceeding raw images by 22.90 points and annotated images by 27.07. Appending either image type to the text lowers every task score. For the same tool outputs, text is a more usable interface than region marks for these backbones. The textual serialization also matters: language rendering and question pruning (S2) raise Planning and Behavior over the same outputs serialized as JSON records (S1) by 10.8 and 16.1 points.

Takeaway 2: Most of the gain comes from scene evidence rather than task rules. Both extraction and representation matter: visual processing tools supply useful information, and structured scene text makes it more usable than the same outputs drawn as visual marks.

4.3 What Must the Text Contain?

Information importance w.r.t. questions. Tab. 5 tests concept contributions. Spatial relations give the largest standalone gains on Perception and Prediction, road context on Behavior, and ego context and risk on Planning. Ego context and risk, the only concept from task rules (level (iv)), has a Planning gain (+30.83) close to those of relations and road context (+27.22, +26.30), and its removal costs only 3.48. Removing object identity causes the largest Perception loss; removing road context causes the largest Prediction loss and is the only removal that lowers every metric. Negative necessity values show that information useful alone can be counterproductive alongside other records.

Evidence stability under tool perturbations. Under joint $\pm 20\%$ perturbations of five tool thresholds, raw records change for 54.8–72.0% of samples, yet rendered prompts change for only 9.0–19.4%; content score stays within 0.22 points of the default (Appendix D.4).

Takeaway 3: The value of extracted visual information depends on the question. Structured scene text should preserve relevant concepts while limiting details that distract downstream reasoning.

Table 5: **Information contribution.** *Sufficiency* is the score with only one concept minus the question-only score; *Necessity* is the all-concept score minus the score after removing that concept and its dependents. Positive values indicate useful evidence; **bold**: largest sufficiency per task. The question-only Prediction score is 0.0 (judge refusals), inflating every Prediction sufficiency.

Evidence concept	Perception bAcc		Prediction GPT		Planning GPT		Behavior bAcc	
	Sufficiency	Necessity	Sufficiency	Necessity	Sufficiency	Necessity	Sufficiency	Necessity
Object identity and category	+16.40	+4.17	+46.23	+0.97	+8.87	+0.59	+1.38	-5.50
3D geometry and orientation	+4.61	-8.66	+38.70	-0.20	+20.00	+2.19	+6.63	-4.65
Road and control context	+0.00	+0.85	+37.30	+8.60	+26.30	+4.11	**+17.49**	+1.20
Object–ego spatial relations	**+17.09**	-4.37	**+48.72**	-7.07	+27.22	+6.20	+8.21	-7.21
Ego context and risk (iv)	+0.00	+0.00	+42.36	-2.21	**+30.83**	+3.48	+5.25	-4.69

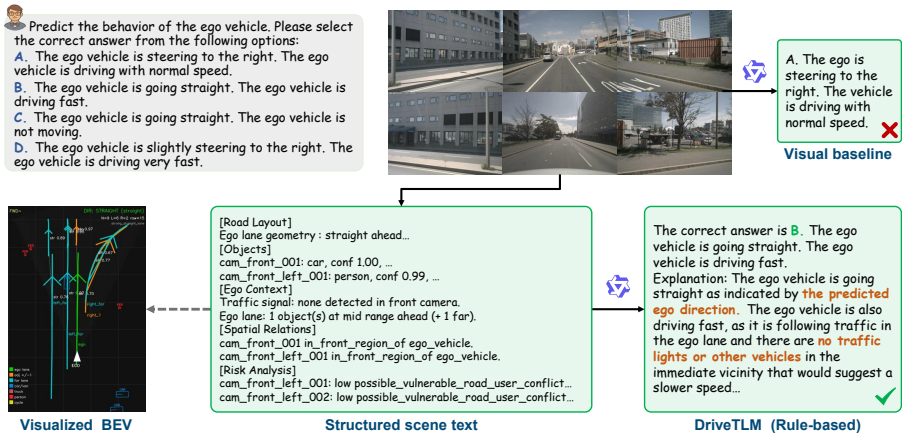

Figure 4: **A qualitative example on DriveBench.** The visual baseline selects the wrong behavior, whereas DriveTLM Rule-based answers correctly using structured scene text. The BEV rendered from this text illustrates the extracted geometry and spatial relations it preserves.

4.4 Does the Model Actually Use the Text?

Matched single-fact interventions. Because the answering model sees only text, its answers should follow the relevant scene facts and ignore irrelevant ones. We test this by appending one controlled fact to otherwise identical structured scene text and varying it between paired conditions (Tab. 6). Relevant paired edits change answers in 57.0–93.4% of cases, whereas irrelevant weather edits leave 77.0–97.0% unchanged; positive margins on all four templates indicate greater responsiveness to the relevant edits than to the weather control (Appendices D.3 and D.5).

Table 6: **Matched single-fact evidence interventions.** Sensitivity is the percentage of answers that change when the queried fact is reversed; invariance is the percentage that remain unchanged after an irrelevant weather edit. Margin = sensitivity + invariance −100, so positive values indicate a selective response to the relevant fact.

Question template	n	Sensitivity	Invariance	Margin
Notice/action	61	**93.4**	77.0	**70.4**
Behavior	200	74.0	92.5	66.5
Sign or barrier	200	68.5	**97.0**	65.5
Moving status	200	57.0	88.0	45.0

Qualitative results. In Fig. 4, the visual baseline predicts a right turn, whereas structured scene text explicitly describes the straight ego trajectory and supports the correct answer. Appendix E provides further frame and video examples, including tool selection and temporal information.

5 Conclusion

DriveTLM recasts vision as structured scene text through visual processing tools, allowing the answering model to reason without visual tokens or parameter updates. Across DriveBench and LingoQA, it improves over direct visual input across model families and sizes, especially for smaller backbones. Controlled studies show that most gains persist without task rules and depend on both scene evidence and its textual representation. DriveTLM therefore provides a modular, inspectable interface that helps reveal how visual extraction, task rules, and downstream language reasoning jointly shape both successful and failed answers in driving VQA.

Reproducibility Statement

Sec. 3 defines the tool library, Rule-based and Agentic routing, the deterministic textualizer, and frame and video evidence construction. Sec. 4 specifies the benchmarks, answering backbones, input conditions, decoding setup, hardware, latency protocol, and scene-disjoint configuration development. The appendix provides model configurations, tool thresholds, object and temporal association, and the renderer scope (Appendix A); development-set construction, benchmark protocols, answer extraction, and controlled-study metrics (Appendix B); paired uncertainty, cross-backbone transfer, robustness, and controlled video comparisons (Appendix C); and ablation, intervention, threshold-sensitivity, and failure-analysis protocols (Appendix D). All benchmark data, answering backbones, and tool-model checkpoints used in local runs are publicly released; API-based judges follow the benchmark protocols. The code will be released upon publication.

AI Use Statement

We used LLMs to improve the clarity and readability of the manuscript and to assist in identifying relevant literature. All AI-assisted content and retrieved references were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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. *arXiv preprint arXiv:2502.13923*, 2025.

Yuliang Cai, Dongqiangzi Ye, Zitian Chen, and Chongruo Wu. Efficient visual question answering pipeline for autonomous driving via scene region compression. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 2026.

Amirhosein Chahe and Lifeng Zhou. Reasondrive: Efficient visual question answering for autonomous vehicles with reasoning-enhanced small vision-language models. *arXiv preprint arXiv:2504.10757*, 2025.

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. In *IEEE International Conference on Robotics and Automation*, pp. 14093–14100, 2024.

Tushar Choudhary, Vikrant Dewangan, Shivam Chandhok, Shubham Priyadarshan, Anushka Jain, Arun K. Singh, Siddharth Srivastava, Krishna Murthy Jatavallabhula, and K. Madhava Krishna. Talk2bev: Language-enhanced bird's-eye view maps for autonomous driving. In *IEEE International Conference on Robotics and Automation*, 2024.

Charles Corbière, Simon Roburin, Syrielle Montariol, Antoine Bosselut, and Alexandre Alahi. DrivingVQA: A dataset for interleaved visual chain-of-thought in real-world driving scenarios. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026.

Yaodong Cui, Shucheng Huang, Jiaming Zhong, Zhenan Liu, Yutong Wang, Chen Sun, Bai Li, Xiao Wang, and Amir Khajepour. Drivellm: Charting the path toward full autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*, 9(1):1450–1464, 2024.

Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, pp. 910–919. IEEE, 2024.

Akshay Gopalkrishnan, Ross Greer, and Mohan Trivedi. Multi-frame, lightweight & efficient vision-language models for question answering in autonomous driving. *arXiv preprint arXiv:2403.19838*, 2024.

Xianda Guo, Ruijun Zhang, Yiqun Duan, Yuhang He, Chenming Zhang, Shuai Liu, and Long Chen. Drivemllm: A benchmark for spatial understanding with multimodal large language models in autonomous driving. *arXiv preprint arXiv:2411.13112*, 2024.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, 2024.

Weikai Huang, Jieyu Zhang, Sijun Li, Taoyang Jia, Jiafei Duan, Yunqian Cheng, Jaemin Cho, Matthew Wallingford, Rustin Soraki, Chris Dongjoo Kim, Shuo Liu, Donovan Clay, Taira Anderson, Winson Han, Ali Farhadi, Bharath Hariharan, Zhongzheng Ren, and Ranjay Krishna. Wilddet3d: Scaling promptable 3d detection in the wild. *arXiv preprint arXiv:2604.08626*, 2026.

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*, 2026.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

Younggun Kim, Ahmed S Abdelrahman, and Mohamed Abdel-Aty. Vru-accident: A vision-language benchmark for video question answering and dense captioning for accident scene understanding. In *2025 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 772–782. IEEE, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, Junchi Yan, Ping Luo, and Hongyang Li. Graph-based topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *ICLR*, 2026a.

Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-r1: Bridging reasoning and planning in vlms for autonomous driving with reinforcement learning. In *AAAI*, 2026b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next`.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. Chameleon: Plug-and-play compositional reasoning with large language models. In *Advances in Neural Information Processing Systems*, 2023.

Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. Dolphins: Multimodal language model for driving. *arXiv preprint arXiv:2312.00438*, 2023.

Jiageng Mao, Yuxi Qian, Junjie Ye, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023.

Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. In *Conference on Language Modeling*, 2024.

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, 2024.

Jianbiao Mei, Yukai Ma, Xuemeng Yang, Licheng Wen, Xinyu Cai, Xin Li, Daocheng Fu, Bo Zhang, Pinlong Cai, Min Dou, Botian Shi, Liang He, Yong Liu, and Yu Qiao. Continuously learning, adapting, and improving: A dual-process approach to autonomous driving. In *Advances in Neural Information Processing Systems*, 2024.

Safaa Abdullahi Moallim Mohamud et al. Hierarchical question-answering for driving scene understanding using vision-language models. *arXiv preprint arXiv:2506.02615*, 2025.

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 *ECCV*, 2024.

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)*, 2025.

Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *AAAI Conference on Artificial Intelligence*, 2024.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11993–12003. IEEE, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. 2023.

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.

Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15120–15130, 2024.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*, 2023.

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 *European Conference on Computer Vision*, pp. 256–274, 2024.

Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Ran Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In *Conference on Robot Learning*, 2024a.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, 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*, 2024b.

Hao Vo, Khoa Vo, Phu Loc Nguyen, Sieu Tran, Duc Minh Nguyen, Ngo Xuan Cuong, Gladys Gawugah, Sreevenkata Anjani Tishita Godavarthi, Chase Rainwater, Nghi DQ Bui, et al. Drivespatial: A benchmark for spatiotemporal intelligence in vlms for autonomous driving. *arXiv preprint arXiv:2605.23176*, 2026.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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 *ICCV*. IEEE, 2025.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*, 9(10):8186–8193, 2024.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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*, 2023a.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023b.

Seungjun Yu, Seonho Lee, Namho Kim, Jaeyo Shin, Junsung Park, Wonjeong Ryu, Raehyuk Jung, and Hyunjung Shim. Waymoqa: A multi-view visual question answering dataset for safety-critical reasoning in autonomous driving. *arXiv preprint arXiv:2511.20022*, 2025.

Tong Zeng, Longfeng Wu, Liang Shi, Dawei Zhou, and Feng Guo. Are vision llms road-ready? a comprehensive benchmark for safety-critical driving video understanding. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5972–5983, 2025.

Weicheng Zheng, Xiaofei Mao, Nanfei Ye, Pengxiang Li, Kun Zhan, Xianpeng Lang, and Hang Zhao. Driveagent-r1: Advancing vlm-based autonomous driving with active perception and hybrid thinking. In *ICLR*, 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.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Implementation Details

This appendix specifies how the tool interfaces in the main text are implemented and connected. We focus on model configurations, object association, and the construction of question-specific evidence; the main text defines the representation and the role of each tool category.

A.1 Frame Tools and Object Association

Model configurations. The object detector uses Faster R-CNN with a ResNet-50 feature pyramid backbone and the torchvision V2 weights (Ren et al., 2015). WildDet3D (Huang et al., 2026) processes each camera view separately, using the categories car, truck, bus, pedestrian, bicycle, and motorcycle. Scene classification uses CLIP ViT-L/14 at resolution 336 for weather and time of day (Radford et al., 2021), alongside image statistics for illumination and visibility. CLIP classification of individual object attributes is disabled in the evaluated configuration. TopoNet (Li et al., 2023) supplies lane topology within a lateral range of 10.5 m and a forward range of 50 m. All models use their publicly released weights without fine-tuning.

Associating 2D and 3D detections. WildDet3D attributes are attached to 2D detections within the same camera view. A candidate pair is retained if its box intersection over union (IoU) is at least 0.05 or its normalized center distance is at most 0.25. Candidates are ranked by IoU plus a 0.08 category-compatibility bonus minus 0.25 times the center distance. Each 3D detection is used at most once. The matched record preserves the 2D box and adds available orientation, depth, and fine category, together with association metadata and the source. This association provides the provenance needed to relate geometric attributes to a particular detected object.

Tool dependencies and queried objects. The agentic router returns a list of valid tool names, after which dependency resolution adds `get_objects` whenever WildDet3D, spatial relations, or risk analysis requires it. The resolver removes duplicate calls and restores a fixed execution order. When a question identifies an object through a camera and image coordinates, the renderer selects the nearest detected box center in that camera. This uses the object reference supplied in the question, rather than an answer annotation, and makes the resulting state statement attributable to a specific detection.

Operating point. The frame configuration uses confidence thresholds of 0.35 for the 2D detector and 0.20 for WildDet3D, a maximum association center distance of 0.25, a minimum association IoU of 0.35 for the queried-object orientation hint, and a TopoNet score threshold of 0.65. The association gate and the stricter hint threshold serve different purposes: the former attaches candidate attributes, while the latter controls their use in question-specific text. Appendix D.4 reports joint perturbations of these five parameters.

A.2 Temporal Evidence

Implementation interface. The vehicle interface described as `get_traffic_dynamics` in the main text is implemented as `get_traffic_gap_dynamics`. It operates on associated frame detections alongside `get_pedestrian_dynamics` and `get_signal_dynamics`. The video builder stores snapshot and temporal text separately, then combines the selected sections before answer generation. This separation supports the snapshot-only and temporal-only controls in Appendix C.6 without rerunning visual extraction.

Interpreting a tracked trajectory. Fig. 5 illustrates the distinction between relative motion and absolute speed. Across F0–F4, the bus box area decreases from 13.9% to 1.6% of the image, while estimated depth increases from 16.8 to 37.2 m. Both observations support an opening gap relative to the ego vehicle. The rendered trajectory links this change to the current bus identifier; it does not infer the bus's absolute speed or use a historical detection as evidence of current presence.

A.3 Scope of the Deterministic Renderer

The renderer implements the two parts of Eq. 1, and we assign each statement to a level by how it is constructed rather than by which callable tool produces it. Φ_{ev} writes (i) *tool estimates*, such as

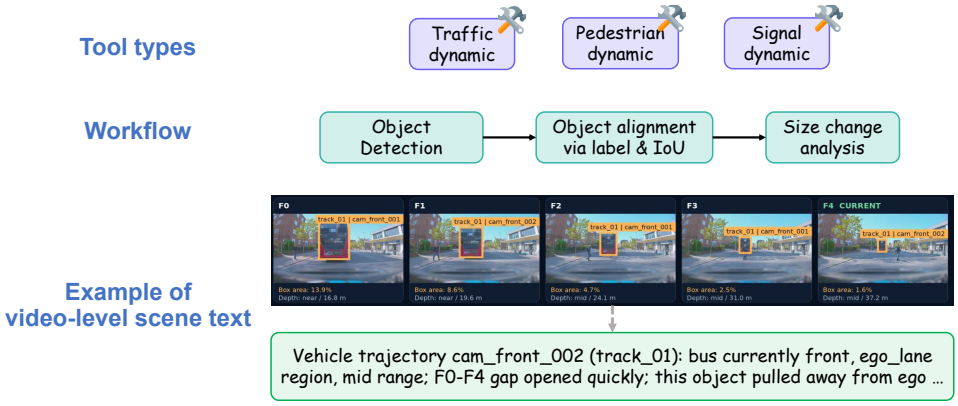

Figure 5: **From frame observations to temporal text.** The temporal interfaces associate detections and summarize changes over F0–F4. The bus example links an opening relative gap to a current object identifier. The record describes observed change and does not estimate absolute vehicle speed.

categories, boxes, depth, yaw, and lane predictions; (ii) *geometric transforms*, such as box-derived position and depth order; and (iii) *semantic relations*, such as being ahead of the ego vehicle or in its lane. Φ_{task} writes (iv) *ego-state and risk inferences* to `[Ego Context]` and `[Risk Analysis]`, and (v) *answer-oriented statements*, namely moving-status labels, option matching, action recommendations, and answer directives, to `[Question-Relevant Evidence]`, `[Queried Object Orientation]`, and, for video, `[Planning Candidate]`. Level (i) contains predictions rather than ground truth; later levels inherit upstream errors and add deterministic assumptions.

In Φ_{ev}, the question only selects sections; Φ_{task} also uses it to resolve the object referenced by camera and coordinates and to construct task interpretations. For example, a camera-dependent rule maps estimated yaw to moving-status wording, and another compares estimated ego direction with Behavior options. A yaw-based moving-status label is a heuristic task prediction, not a direct measurement of motion. These rules supply task interpretations of the estimated states, which the controls below separate from the scene description.

Three controls remove these levels. S3 in Tab. 3 deletes the Φ_{task} sections from the frozen DriveTLM prompts and leaves every other token unchanged, isolating levels (iv)–(v); S3a restores level (iv) alone (Appendix D.1). The last group of Tab. 5 contains only level (iv), so dropping it retains level (v). S2 also lacks levels (iv)–(v) but is constructed separately, so its comparison with DriveTLM additionally changes object references and prompt construction. S3c and S3d further hold the queried-object records fixed while changing their form, from raw 3D measurements to answer-oriented derived values (Appendix D.1).

Training-free here means that the pipeline updates no model parameters. The configuration used in all reported benchmark evaluations was developed on the scene-disjoint development set in Appendix B.1 and fixed before those evaluations; evaluation labels and benchmark scores were not used to revise it. The local threshold study measures sensitivity around this operating point.

B Evaluation and Experimental Protocols

We distinguish benchmark comparison from diagnostic evaluation throughout the appendix. Benchmark tables retain the released task metrics, subject to the Behavior extraction correction below. Controlled studies additionally measure recovered answer content and sensitivity to changes in the supplied evidence.

B.1 Development Set

Source and role. We use the DriveLM-nuScenes v1.1 training split (Sima et al., 2024) as a development source for the pipeline configuration, including tool settings, thresholds, textual templates, driving-level rule parameters, and the agentic routing prompt. The designation "training split" fol-

Table 7: **Development-set diagnostics.** Rule-based DriveTLM versus raw-image input with Qwen2.5-VL-3B-Instruct on 100 DriveLM-nuScenes keyframes from 100 scenes disjoint from DriveBench, with 100 questions per task. Acc. and bAcc are percentages; *Unans.* counts responses without an extracted option. Accuracy differences are in percentage points, with paired bootstrap 95% intervals in brackets.

Input	Moving status			Behavior		
	Acc.	bAcc	Unans.	Acc.	bAcc	Unans.
Raw images	33.0	28.9	9	37.0	38.0	8
DriveTLM (rule-based)	**58.0**	**57.0**	**7**	**48.0**	**48.9**	**0**
Δ	+25.0 [+13.0, +37.0]	+28.1	−2	+11.0 [0.0, +22.0]	+10.9	−8

lows DriveLM: we use its annotations for pipeline development without training or fine-tuning the underlying VLM. Benchmark evaluation is conducted separately on DriveBench and LingoQA.

Scene-level separation. The DriveLM-nuScenes v1.1 training split contains 696 scenes. The 200 DriveBench frames, associated with 1,461 questions, originate from 174 of these scenes. We exclude all 174 scenes before constructing the development set, leaving 522 scenes containing 3,003 keyframes. Thus, development and DriveBench evaluation share neither scenes nor keyframes, although they share the underlying data source. We refer to this subset as a *development set* because its annotations inform configuration choices.

Eligibility criteria. A keyframe is eligible when it contains both a suitable moving-status annotation and a suitable behavior annotation. The moving-status question must reference an object by camera and pixel coordinates, with an answer belonging to the three-option vocabulary: *Going ahead*, *Turn left*, or *Turn right*. The behavior answer must contain exactly one steering sentence and one speed sentence from the DriveBench vocabulary, which comprises 5 steering phrases and 5 speed phrases. These criteria retain 2,674 eligible keyframes from 514 scenes.

Sampling. Using a fixed random seed, we shuffle the eligible scenes and sample one eligible keyframe from each of 100 scenes.

Question construction. Each selected keyframe yields two multiple-choice questions, giving 200 questions in total:

- **Moving status.** We sample one question with the target moving-status label and normalize the referenced object's pixel coordinates by the 1600×900 image dimensions. We shuffle the three answer options and use the referenced camera image as input.
- **Behavior.** We use the DriveLM behavior annotation as the correct answer. Of the 25 possible steering–speed combinations, the remaining 24 form the distractor pool. We sample three distractors without replacement, shuffle the resulting four options, and use all six camera views as input.

This conversion allows development and benchmark evaluation to use the same multiple-choice inference and answer-extraction pipeline.

Configuration development. We use this set to develop the pipeline configuration. Development feedback informs tool settings, thresholds, rule parameters, textual templates, and the agentic routing prompt. These are pipeline-level choices; no VLM parameters are updated. Development experiments use Qwen2.5-VL-3B-Instruct with greedy decoding in vLLM. The configuration is fixed before benchmark evaluation, and the reported benchmark results use it without further adjustment based on evaluation labels or scores.

Scoring protocol. Moving status is scored with the released DriveBench answer extractor, whereas behavior uses the corrected extractor (Appendix B.3). Balanced accuracy (bAcc) is the mean recall over answer letters (Appendix B.4), rather than over semantic moving-status classes. A response is counted as unanswered when the extractor fails to identify an option. The raw-image baseline uses the original 1600×900 source images. No GPT-based judge is used for this development evaluation.

Development-set observations. Table 7 reports moving-status accuracy of 58.0% with rule-based DriveTLM versus 33.0% with raw images, and behavior accuracy of 48.0% versus 37.0%. Balanced accuracy increases from 28.9% to 57.0% for moving status and from 38.0% to 48.9% for behavior.

Scope. The development set covers moving status and behavior, with questions in the DriveBench multiple-choice format. LingoQA, with a different annotation source and video input, is evaluated with the same configuration.

B.2 Benchmark Protocol for DriveBench

Question set and judging. Our clean-set runs use the same $1{,}461$ questions. Under the released task distribution, $1{,}261$ receive a GPT score; the 200 Sign/barrier questions do not. GPT scoring uses the released task-specific rubrics with `gpt-3.5-turbo` at temperature 0. The Perception GPT cell pools 200 MCQs and 200 open-ended questions. Behavior MCQ accuracy uses the corrected extractor in Appendix B.3; Perception MCQ accuracy retains the released extractor.

Aggregate score. The **Avg.** column in Tab. 1 gives equal weight to the eight reported task metrics: a GPT score and either MCQ accuracy or ROUGE-L for each of the four tasks. This summary retains both judged answer quality and the task's direct metric. We recompute the same mean for published baselines when all eight cells are available; incomplete rows have no aggregate. This eight-metric mean differs from the four-task diagnostic summaries used in the controlled studies.

Importance of complementary metrics. Model-based judgments and exact answer correctness capture different aspects of a response. An explanation may convey relevant scene semantics or partially agree with the reference while still committing to the wrong option. Consistent with this distinction, rule-based Qwen2.5-VL-3B obtains a higher Behavior GPT score than its agentic counterpart (60.3 versus 58.9), but lower MCQ accuracy (52.0% versus 53.0%). A related disagreement appears on LingoQA: with Qwen3-VL-32B, captioning achieves a higher Lingo-J score than rule-based video DriveTLM (62.4 versus 58.2), yet much lower CIDEr (0.2 versus 22.0). These comparisons show that judge agreement, option correctness, and reference overlap need not rank systems identically; they do not by themselves establish that richer wording inflates judge scores. We therefore retain judge-based evaluation alongside MCQ accuracy and reference-based language metrics. MCQ accuracy directly tests the selected answer, whereas ROUGE-L measures overlap with the reference wording. Since overlap is itself sensitive to phrasing, we additionally use the content metrics in Appendix B.4 to distinguish recovered driving facts from surface similarity.

Visual input and response format. The benchmark comparisons use 1600×900 source images, with the same source resolution and inference backend across input conditions within each backbone. Model-specific visual preprocessing remains part of each backend. The raw-image rows of Tabs. 4 and 11 reuse the benchmark image run; the remaining image controls of Tab. 4 use the same source resolution with the pipeline prompt. Benchmark answers follow the released instruction to give the selected choice together with an explanation.

Published and reproduced baselines. Published results retain their reported values and evaluation coverage. The human study covers Perception and Behavior, and DriveAgent-R1 provides only its published clean-setting metrics. For ReCogDrive, we supplement its published GPT scores with four metrics reproduced from the released checkpoint and inference protocol. That reproduction preserves camera tags, the benchmark prompt in both system and user turns, dynamic tiling, and decoding with temperature 0.01, top-p 0.001, top-k 1, and 50–4096 new tokens. The reproduced cells are marked with † in Tab. 1; they are distinct from the common-interface experiments in Appendix C.4.

B.3 Answer Extraction for Multiple-Choice Questions

Artifact in the released extractor. The released extractor can confuse an explanation with the selected option. It lowercases the response and searches for whole-word matches to A, B, C, and D in that order. Consequently, the article "a" in "a slow, cautious turn" is sufficient to return A even when the response explicitly selects D. Mentioning A among rejected options can produce the same error. The resulting bias is directional: an always-A predictor scores 21.5% on Behavior, where A occurs 43 times, but 59.0% on Perception, where A occurs 118 times.

Correction and score coverage. The corrected extractor selects the earliest explicit answer marker: "the answer is X," "Answer: X," "Option X," or an uppercase option letter at the beginning of the response or a line, with the required punctuation or standalone form. The option letter must remain uppercase, preventing an ordinary article from matching. Responses without a recognized choice are marked unanswered and remain in the denominator. If every response omits a scoreable option letter, we report the cell as unavailable rather than interpreting an extractor artifact as model accuracy.

Effect on Behavior accuracy. Applying the correction to stored responses changes Qwen3-VL-8B from 21.5 to 52.5 and InternVL3-38B with rule-based DriveTLM from 27.5 to 52.5. The two Qwen2.5-VL-3B DriveTLM variants change from $37.0/29.5$ to $52.0/53.0$. Image and caption baselines also improve, so the correction narrows some gains. It is applied consistently to our reproduced Behavior results, while numbers quoted from prior work remain as published when their raw responses are unavailable.

Why Perception remains separate. Perception responses often state a valid letter in the middle of a sentence, such as "The moving status of object ⟨c1,...⟩ is C. Going ahead." The corrected extractor can miss this format. An audit of 180 stored runs found that adding a mid-sentence fallback changed 23 Perception scores and no Behavior score. We therefore retain the official extractor for benchmark Perception accuracy and explicitly distinguish stricter extraction diagnostics below. Neither extractor fully resolves responses that provide only option text.

B.4 Metrics for the Controlled Studies

Balanced MCQ accuracy. Balanced accuracy measures mean recall over answer options, preventing a dominant label from driving the mechanism conclusions. The moving-status split has A/B/C counts of $118/63/19$: the question-only condition obtains 57.5% raw accuracy but 32.5% balanced accuracy, close to the three-class chance level of 33.3%. Behavior counts are $43/55/51/51$ for A/B/C/D. The main mechanism tables use balanced accuracy for both MCQ tasks, whereas the benchmark table reports raw accuracy. Unanswered predictions count as misses in both cases.

Recovered answer content. GPT judging every controlled condition is costly, and ROUGE-L rewards reference phrasing rather than recovered facts (Appendix D.2). The controlled studies therefore use content metrics, which score the facts expressed in an answer after canonicalizing its wording. Important objects uses set-F1 over category and relative-position pairs; color is omitted because the evaluated tools do not recover it. Collision action maps text to 12 canonical classes, checking negation before action phrases. Safe actions uses set-F1 over 14 atomic actions. Possible action averages action-set F1, reason accuracy, and probability accuracy; Notice averages category, camera, and state set-F1. Sign or barrier uses balanced yes/no accuracy because the references contain 194 No and 6 Yes answers. The content pipeline uses explicit-choice extraction for both MCQ families, which must be distinguished from benchmark Perception extraction.

Parser coverage and limitations. The reference parsers cover 99.0% of Objects, 100% of Motion, 98.4% of Notice, 100% of Sign, 100% of Behavior, 97.0% of Possible, 100% of Collision, and 99.5% of Safe. This check establishes that the reference vocabulary is largely recoverable; it does not establish equivalent coverage of arbitrary model paraphrases. Every condition uses the same parsers. Low action scores can therefore reflect both an incorrect decision and a mismatch with the closed vocabulary, a distinction revisited in Appendix D.5.

Task grouping and aggregation. Perception combines Objects and Motion, Prediction combines Notice and Sign, Planning combines Possible, Collision, and Safe, and Behavior contains its MCQ family. Within-task means follow question counts: the equal-sized templates have equal weight, while Prediction weights Notice and Sign by 61 and 200. In Tab. 11, the first four columns use these content scores, including raw explicit-choice accuracy for MCQs; the final columns separately give Perception recalls and balanced accuracy under benchmark extraction. The threshold study instead reports an equal-weight mean over the five scene-state families, Objects, Motion, Notice, Sign, and Behavior, and names this aggregate in its caption.

B.5 Protocol for LingoQA

Questions and metrics. Our comparisons use all 500 validation questions. Lingo-J is the percentage judged correct by the released Lingo-Judge classifier, not a mean logit or probability. BLEU-4, METEOR, and CIDEr use `pycocoevalcap`, with values multiplied by 100 for presentation; this scaling does not bound CIDEr at 100. Latency includes input construction and answer inference and excludes initialization and offline evaluation, as in the main experiments.

Input conditions. Raw frames supplies the source images directly, while Caption supplies a caption generated by the same answer backbone. A one-frame DriveTLM condition constructs evidence from the final frame; a five-frame condition also uses the clip's temporal records. At each granularity, rule-based and agentic selection share the renderer and answer stage. Missing conditions indicate unavailable or unexecuted combinations, rather than zero performance.

Matched response-schema control. The evidence comparison in Appendix C.6 gives every condition the same question-specific output rule, copied from the complete video pipeline. It removes derived ego/action recommendations from the snapshot and temporal evidence controls while retaining observed facts. Question-only, current-frame, and raw-video controls receive the same output rule with no tool text. This setup distinguishes access to evidence from access to the requested answer format; the standard benchmark baseline already follows this rule.

C Additional Experimental Results

The following experiments extend the main comparisons across answer backbones, driving-specific checkpoints, and corrupted observations. A matched video control then identifies where temporal evidence helps and whether raw video adds to tool text.

C.1 Paired Uncertainty for DriveBench

We compare each backbone's best structured scene text setting in Tab. 1 with its raw-image input, recomputing MCQ accuracy and per-question GPT scores from the stored outputs and resampling questions with replacement (10,000 paired bootstrap draws). Behavior MCQ gains are significant at all three scales: $+14.5$ [6.5, 23.0] at 3B, $+9.0$ [1.0, 17.0] at 7B, and $+11.5$ [2.5, 20.5] at 38B. The average gains are also significant, $+21.6$ [19.7, 23.6], $+6.1$ [4.1, 8.1], and $+3.7$ [2.1, 5.2]; these intervals use per-question ROUGE-L, which differs slightly from the corpus-level value in Tab. 1. At 3B, every GPT score gain excludes zero with a lower bound of at least 11.1 points.

C.2 Rule-based and Agentic Tool Selection

Agentic selection retains performance close to full-pool Rule-based execution while reducing the number of executed tools. On DriveBench, it improves six of eight metrics at 3B and raises the table average from 48.9 to 50.2, while reducing latency from 3.8 s to 1.7 s. At 38B, the two strategies obtain the same average score of 50.9, with Agentic reducing latency from 7.7 s to 5.3 s. Across the evaluated LingoQA backbones, their Lingo-J scores remain within 1.6 points. These results characterize Agentic selection as a sparse execution strategy that preserves most of the full-pool performance while avoiding unnecessary tool calls.

C.3 Transfer Across Answer Backbones

Frame inputs. Fig. 6 extends the DriveBench comparison to 11 VLMs and four text-only Qwen3 models. The gains in Perception and Behavior extend across multiple model families and sizes, while Planning shows a less consistent response to the input representation. The LLM results span 0.6B to 8B, establishing that the interface can be consumed without a visual encoder across several language-model scales. Qwen2.5-VL-32B remains a notable Behavior failure, examined in Appendix D.5; model size alone does not ensure reliable use of the supplied records.

Video inputs. Fig. 7 compares nine backbones on LingoQA. For every plotted backbone, at least one video DriveTLM variant exceeds raw video in Lingo-J and exceeds both raw video and captions in

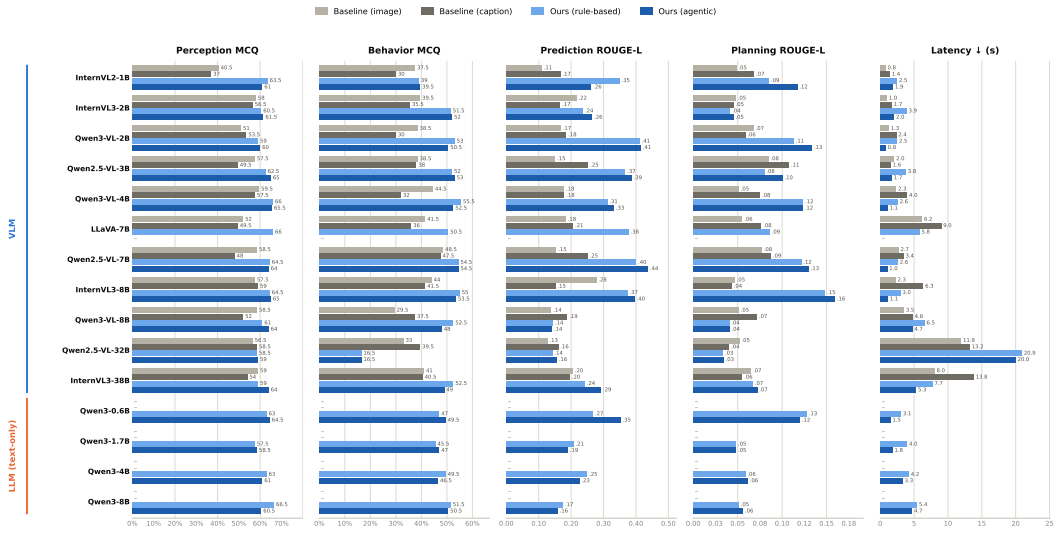

Figure 6: **Additional answer backbones on DriveBench.** 11 VLMs and four text-only Qwen3 models, ordered by size within each group. Image and Caption denote direct visual input and a caption from the same backbone; the two DriveTLM conditions use rule-based or agentic selection. MCQ values are percentages, plotted ROUGE-L values are fractions, and latency is in seconds. Missing bars indicate unavailable or unexecuted conditions.

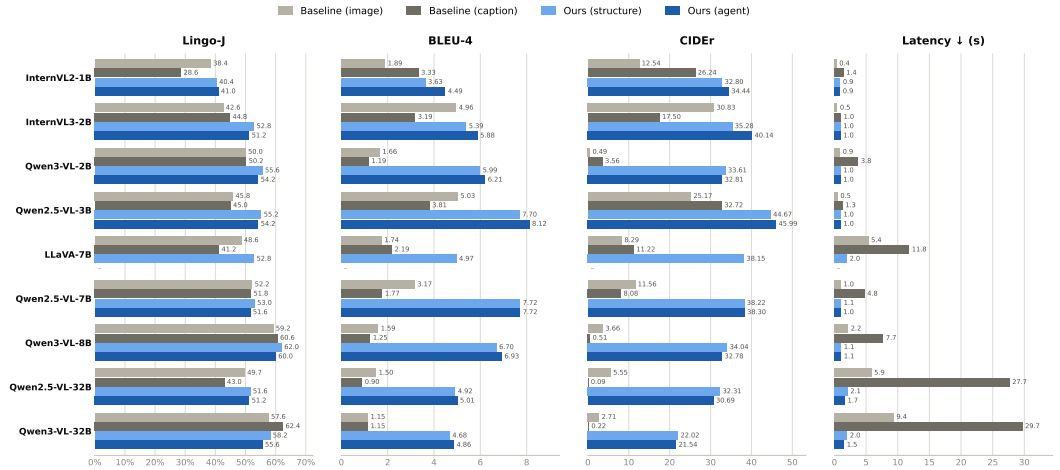

Figure 7: **Additional answer backbones on LingoQA.** 9 VLMs receive raw video, a caption from the same backbone, or five-frame DriveTLM evidence. In the plot legend, "image" denotes the raw-frame video input and "structure" denotes rule-based selection. Lingo-J, BLEU-4, and CIDEr use the paper's percentage scaling; latency is in seconds. Missing bars indicate unavailable or unexecuted conditions.

BLEU-4 and CIDEr. Its latency is comparable to or below captioning throughout the plot. These comparisons establish broad compatibility, but the preferred representation can still depend on the metric: agreement with Lingo-Judge and overlap with reference wording need not rank answers identically.

A larger-backbone exception. Qwen3-VL-32B illustrates this metric dependence. Temporal evidence adds 1.4 Lingo-J points to the rule-based frame pipeline, but the resulting video system exceeds raw video by only 0.6 points. Captioning attains a higher Lingo-J of 62.4, yet its CIDEr is 0.2 and its latency is 29.7 s, compared with 22.0 and 2.0 s for rule-based video DriveTLM. Thus, the caption advantage under the judge does not extend to reference overlap or runtime.

Table 8: **Input representation with driving-specific checkpoints on DriveBench.** All conditions use the clean split; accuracy and ROUGE-L are percentages on their corresponding task subsets. Latency is in seconds. Bold marks the best value within each checkpoint, with lower latency preferred. Here † denotes training on the source QA data used to construct DriveBench. A Behavior dash indicates no scoreable option letter in any response. Only completed conditions are shown.

Model	Input	Perc. MCQ	Beh. MCQ	Pred. R-L	Plan. R-L	Latency ↓
ReCogDrive† (Li et al., 2026a)	Raw image	58.0	38.5	25.9	11.3	**1.0**
	Caption	57.5	40.5	23.8	11.5	2.7
	Rule-based (Ours)	58.0	**51.5**	**32.4**	11.7	3.2
	Agentic (Ours)	**59.5**	**51.5**	32.2	**12.0**	3.5
AutoVLA†	Raw image	**59.0**	–	**8.0**	7.1	1.1
	Caption	**59.0**	–	7.3	**8.8**	**0.2**
	Rule-based (Ours)	**59.0**	–	6.8	7.6	2.6
	Agentic (Ours)	**59.0**	–	6.8	7.6	0.7
SimLingo	Raw image	**59.0**	–	2.5	7.2	**0.4**
	Caption	1.0	–	**2.7**	**8.0**	0.8
	Rule-based (Ours)	1.0	–	2.6	7.2	2.6

Table 9: **Robustness to five visual corruptions.** Qwen2.5-VL-3B GPT-judge scores averaged over motion blur, fog, low light, camera crash, and H.265 compression on DriveBench. The shaded row is agentic DriveTLM; bold marks the best score per task.

Model	**Perception**	**Prediction**	**Planning**	**Behavior**
Image	34.5	14.5	32.6	34.1
Caption	28.3	27.9	53.2	39.0
Agentic (Ours)	**47.2**	**45.0**	**76.4**	**50.3**

C.4 Compatibility with Driving-Specific Checkpoints

The common-interface comparison in Tab. 8 tests three driving-specific checkpoints under the same input alternatives. ReCogDrive benefits from structured scene text: rule-based routing improves Prediction ROUGE-L from 25.9 to 32.4 and Behavior accuracy from 38.5% to 51.5%, while agentic routing reaches 59.5% Perception accuracy. These runs test representation compatibility and use a different inference setup from the released-protocol reproduction in Tab. 1; their raw-image rows are consequently different baselines.

AutoVLA (Zhou et al., 2026) and SimLingo (Renz et al., 2025) expose a limit of this transfer. Their outputs frequently contain action-oriented text, waypoints, or failure messages instead of the requested option and explanation, and none of their Behavior responses contains a scoreable option letter. Their 59.0% Perception entries also require caution: this is exactly the score of an always-A response under the released extractor. These results establish difficulty with the tested prompt and response interface, but do not determine whether the checkpoints lack the relevant visual knowledge.

C.5 Robustness to Visual Corruption

Tab. 9 evaluates motion blur, fog, low light, camera crash, and H.265 compression with Qwen2.5-VL-3B. Averaging the four task scores gives 54.7 for agentic DriveTLM, 37.1 for Caption, and 28.9 for Image. Agentic DriveTLM therefore retains approximately 93% of its clean four-task GPT average and leads the two baselines by 17.6 and 25.8 points.

C.6 Controlled Video Evidence on LingoQA

Protocol and question partitions. Tab. 10 compares eight evidence conditions with Qwen2.5-VL-3B under the matched output rule in Appendix B.5. A fixed hierarchy assigns questions to signal change, pedestrian interaction, direction, gap/action, or static description using question text alone. The partitions contain 0, 30, 9, 161, and 300 questions, respectively.

Overall performance and temporal subsets. The raw-video row is the standard 3B baseline in Tab. 2. The complete video pipeline leads it on every subset except direction, where they tie, most on

Table 10: **Video evidence under a matched response schema.** Lingo-J (%) with Qwen2.5-VL-3B on 500 LingoQA questions. Parentheses give partition sizes. The final row is the complete rule-based video pipeline; other rows change the available evidence under its output rule. Bold marks the highest score per column, including ties. The validation split contains no signal-change question.

Condition	Overall	Static (300)	Gap/action (161)	Direction (9)	Pedestrian (30)	Signal change (0)
Question only	27.2	29.3	19.9	33.3	43.3	–
Current frame	44.2	46.0	38.5	**66.7**	50.0	–
Raw video	45.8	49.3	38.5	**66.7**	43.3	–
Snapshot tool text	47.8	51.0	42.2	22.2	53.3	–
Temporal tool text	41.4	42.3	38.5	55.6	43.3	–
Snapshot + temporal text	51.2	51.7	49.7	55.6	53.3	–
Raw video + tool text	50.4	50.0	50.9	55.6	50.0	–
DriveTLM-Video (complete pipeline)	**55.2**	**53.3**	**57.1**	**66.7**	**60.0**	–

gap/action (57.1 versus 38.5) and pedestrian (60.0 versus 43.3). On both, raw video is no better than the current frame alone, so the earlier frames give the model little usable motion information.

Paired uncertainty. We resample the 500 questions with replacement (10,000 paired bootstrap draws) for the Qwen2.5-VL-3B comparisons in Tab. 2. Adding temporal information to the frame pipeline raises Lingo-J by 6.2 points (95% CI [2.4, 10.0]), including +4.7 [0.3, 9.0] on static, +8.7 [0.6, 16.8] on gap/action, and +44.4 [11.1, 77.8] on direction questions. Relative to raw video, DriveTLM-Video gains 9.4 points [4.0, 14.8] overall and 18.6 [8.7, 28.6] on gap/action.

Complementary snapshot and temporal information. Combining snapshot and temporal tool text raises Lingo-J from 47.8 to 51.2, whereas temporal text alone scores 41.4. Temporal records therefore complement the current scene description rather than replace it. Adding raw video to the combined text gives 50.4, slightly below text alone, so raw frames bring no gain beyond the tool text.

D Additional Ablations and Analysis

These analyses expand the main representation study into individual construction steps, test sensitivity to relevant facts, and examine tool and rendering failures. Unless specified otherwise, they use Qwen2.5-VL-3B on the 1,461 clean DriveBench questions.

D.1 A Detailed Representation Ladder

Relation to the main stages. Tab. 11 expands the construction stages in Tab. 3a. S1, S2, and S3 denote the same stages as in the main table. S1a separates language rendering from pruning between S1 and S2, and S3a–S3d examine components of the transition from S3 to DriveTLM. S3c and S3d are alternative extensions of S3b, not successive additions of the same fields. The first four columns assess recovered content, while the final columns isolate the class balance of Perception predictions. Only the bAcc column is shared with the main table, which reports GPT scores for Prediction and Planning and balanced accuracy for Behavior.

Intermediate construction steps. The construction steps have mixed effects. S3 exceeds S2 by 6.3 Perception bAcc points but lowers Prediction content by 6.6. This comparison changes identifiers and the orientation heading, so the difference cannot be attributed to identifiers alone. Derived ego and risk statements at S3a recover 2.8 Prediction points, and queried-object facts at S3b add a further 2.7 while reducing Perception bAcc by 7.6. S3 and S3a have identical moving-status prompts because question-based pruning has already removed the ego and risk sections. S3 and S3a–S3d delete sections or lines from the frozen full prompts; S3c additionally inserts raw queried-object measurements from the frozen scene records.

Raw measurements versus driving states. S3c supplies raw queried-object measurements, whereas S3d retains answer-oriented derived values, including moving-status wording and direction-consistent option references, but removes answer directives and action recommendations. Switching from S3c to S3d raises Perception content by 12.3 points and balanced accuracy by 9.2. Raw measurement insertion at S3c had lowered content by 5.8 points despite raising balanced accuracy by 5.2. Under

Table 11: **Detailed evidence stages and extraction diagnostics.** S1, S2, and S3 match the main table; ⋆ marks a corresponding main-table stage. S3c and S3d branch from S3b; D1 and D2 branch from S2, with D2 fixing the image input resolution at 1600×900. The first four columns use the content protocol of Appendix B.4, including raw explicit-choice MCQ accuracy. The last two columns use benchmark Perception extraction to report option recalls and balanced accuracy. All scores are percentages.

Stage	Evidence	Perc.	Pred.	Plan.	Beh.	Perc. recall A/B/C	bAcc
–	Question only	26.00	38.44	9.67	27.50	97 / 0 / 0	32.5
–	Raw image	26.03	42.38	4.45	38.50	92 / 10 / 0	34.0
S1⋆	Numeric tool outputs	18.22	**51.36**	12.13	35.00	37 / 49 / 58	48.1
S1a	S1 + language rendering	38.86	45.07	10.35	48.50	49 / 48 / 42	46.3
S2⋆	S1a + question-based pruning	44.09	46.17	9.97	51.00	50 / 49 / 42	47.1
S3⋆	S2 + identifiers/heading change	45.49	39.57	9.45	49.50	47 / 56 / 58	53.4
S3a	S3 + ego and risk statements	44.30	42.40	10.13	48.00	47 / 56 / 58	53.4
S3b	S3a + queried-object facts	46.72	45.09	11.77	48.50	57 / 33 / 47	45.8
S3c	S3b + raw 3D measurement	40.97	42.72	12.43	48.50	49 / 46 / 58	51.0
S3d	S3b + answer-oriented derived values	53.31	46.36	15.63	51.50	**64 / 63 / 53**	**60.2**
DriveTLM ⋆	S3d + action recommendations and directives	**53.83**	50.87	**19.99**	**52.00**	63 / 65 / 53	60.1
Diagnostics: branches off S2, not rungs of the ladder							
D1	S2 + empty answer fields	45.64	44.86	5.76	39.50	100 / 0 / 0	33.3
D2	S2 + the raw image	31.50	42.51	3.79	21.50	65 / 27 / 21	37.8

explicit-choice extraction, the answering model achieves 36.0% at S3c and 62.0% at S3d, with 51 and 17 responses lacking a recognized option. These are model-output accuracies; the comparison measures the effect of supplying an interpretation of the same records.

Schema and image controls. D1 and D2 branch from S2 rather than extend the ladder. D1 names the required answer fields but supplies no object, camera, state, or option value. Its Perception recalls of $100/0/0$ yield 59.0% raw accuracy and 33.3% balanced accuracy, indicating reliance on the dominant option. D2 adds raw images at a fixed input resolution of 1600×900 and lowers Perception content from 44.09 to 31.50. Neither an empty answer schema nor image addition recovers the missing Perception content.

How much is already expressed in the prompt? Directly reading the renderer's derived moving-status value yields 59.0% accuracy on the 200 Perception MCQs, compared with 62.5% for the complete system. This is distinct from the 62.0% model-output accuracy of S3d above, which also supplies other scene evidence. The 3.5-point gap is a net accuracy difference, not an isolated estimate of reasoning gain. Moreover, an always-A prediction also scores 59.0% on this imbalanced split, so the direct-readout accuracy alone does not establish an advantage over the majority baseline or imply identical predictions. For Behavior, the direction-matching statement isolates one option on 76 questions and is correct on 67.1% of those; this conditional accuracy is not directly comparable with the complete system's 52.0% over all 200 questions. These audits show that task predictions are supplied to the answering model, but do not quantify how often it corrects or overrides them.

D.2 Content Recovery and Answer Phrasing

Why overlap is insufficient. Tab. 12 demonstrates that reference overlap can improve without better object recovery. Raw images obtain 16.70 ROUGE-L but only 2.01 object content F1. Adding required output fields to selected tool facts raises ROUGE-L from 24.97 to 57.26, while content F1 falls from 41.67 to 38.79. A response can therefore resemble the reference template while naming the wrong objects or positions.

What answer composition changes. The full composition condition reaches 43.94 Objects ROUGE-L and 45.16 content F1, improving content even though it has less surface overlap than the schema condition. Notice also requires more than identifying the object category: selected facts score 64.54 on category but zero on camera and state fields, whereas full composition scores 69.24, 50.93, and 50.27. These measurements distinguish recovering a relevant object from expressing the complete information requested by the question.

Limits of the phrasing diagnostic. Tab. 13 reports whitespace-token ROUGE-L for the five open-ended templates, computed by the supplementary analysis script. These values are phrasing

Table 12: **Answer content versus reference overlap.** Objects uses category-position set-F1; Notice columns use set-F1 for the indicated fields. ROUGE-L uses the supplementary whitespace-token diagnostic. All scores are percentages; bold marks the highest value per column.

	Objects		Notice fields		
Condition	ROUGE-L	content F1	category	camera	state
Raw images	16.70	2.01	19.13	5.74	20.77
Selected tool facts	24.97	41.67	64.54	0.00	0.00
+ required output fields	**57.26**	38.79	63.31	37.71	3.28
Full answer composition	43.94	**45.16**	**69.24**	**50.93**	**50.27**

Table 13: **Phrasing across evidence conditions.** Whitespace-token ROUGE-L (%) on the five open-ended templates. Mean gives equal weight to these templates and is not an official benchmark aggregate. Bold marks the largest value per column. These scores complement, rather than replace, content evaluation.

Evidence given to the answer model	Objects	Notice	Possible	Collision	Safe	Mean
Question only	12.02	5.25	7.35	1.15	3.41	5.84
Raw images	16.70	14.49	**18.84**	**4.42**	**13.19**	13.53
Numeric tool outputs	16.32	21.26	10.36	1.51	4.81	10.85
Tool facts in language	22.62	18.36	8.29	1.76	3.65	10.94
Question-relevant tool facts	24.97	17.52	8.21	1.41	3.66	11.15
+ derived high-level facts	24.10	17.26	11.34	1.84	3.94	11.70
Tool values composed for the answer	**43.94**	**33.45**	12.69	2.15	4.70	**19.39**
Raw images + question-relevant facts	18.69	22.96	18.56	2.90	8.55	14.33

diagnostics and need not equal the official evaluator's tokenization-dependent ROUGE-L. The composed Notice sentence itself reaches 59.4 ROUGE-L, compared with 33.5 for the model's answer, so the answer stage does not simply copy all supplied phrasing. We use the content results to assess recovered facts and retain these overlap scores to make the contribution of wording visible.

D.3 Grounding Interventions

Paired edits. The interventions in Tab. 6 append a `[Controlled Fact]` statement to the same selected scene facts, then vary only that statement between paired conditions. The pairs contrast road barrier/passenger car for Sign, moving forward/parked for Motion, steering left/right for Behavior, and stationary/going ahead for the nearest object in Notice. The Motion pair probes sensitivity to a controlled state change rather than a swap between two benchmark answer options. A separate weather edit is applied to the same samples as an irrelevant control. Matching the population and preserving the surrounding text reduce the context changes introduced by whole-scene swaps. Because the answering model receives no pixels, responsiveness to the edited fact is the property the text interface requires; visual grounding rests with the tools, whose contributions Appendix D.4 evaluates.

Comparison unit. Sensitivity compares the extracted answer values under the two relevant edits; invariance compares the original answer with the answer after the weather edit. The comparison uses an explicit option letter for MCQs, yes/no for Sign, and canonicalized category, camera, and state sets for Notice. It reads no ground-truth answer. Possible and Safe are omitted because their open action vocabulary has no paired answer value.

Conflicting controls. The Objects and Collision edits are retained as failure diagnostics because their coarse position statements conflict with more specific object facts already present. Collision has a selective margin of only 1.5 points, and Objects has a negative margin of -10.0 (Tab. 14). Such conflicting additions destabilize answers, which the four matched templates in the main table avoid.

Table 14: **Conflicting interventions as failure diagnostics.** Sensitivity, invariance, and margin are percentages; margin equals sensitivity plus invariance minus 100. The edited coarse position conflicts with existing object facts, so these rows are not clean tests of grounding.

Template	n	Sensitivity	Invariance	Margin
Collision action	200	27.5	74.0	1.5
Important objects	200	28.5	61.5	-10.0

Table 15: **Total effects of removing tools and their dependent text.** Full-system content score minus the score after removal, in percentage points, using the task grouping in Appendix B.4. Positive values indicate that retaining the removed tool and its dependent text improves the score; negative values indicate that their removal improves the score. Bold marks the largest value in each task. All Behavior values are negative, so the bold entry is the least negative measured effect.

Removed tool	Perception	Prediction	Planning	Behavior
2D detector	**+29.75**	**+9.34**	-7.81	-7.50
WildDet3D	-2.54	-0.57	**+2.46**	-3.00
TopoNet	-4.15	+6.63	+2.45	**-1.00**
Scene classifier	-4.61	+6.37	+2.16	-5.00

Table 16: **Local stability under joint threshold changes.** Raw-record and prompt change rates and mean token Jaccard are percentages. Score is the equal-weight content mean over Objects, Motion, Notice, Sign, and Behavior, rather than the benchmark average. Δ is relative to the frozen default. The profile coordinates are specified in the text.

Profile	Raw change	Prompt change	Token Jacc.	Score	Δ
Default	0.0	0.0	100.0	53.16	+0.00
Permissive 20%	70.4	15.9	98.5	53.27	+0.11
Conservative 20%	72.0	19.4	98.0	53.37	+0.21
Mixed A	55.0	11.0	98.8	52.94	-0.22
Mixed B	60.2	9.0	99.3	53.23	+0.07
Mixed C	54.8	12.1	98.9	52.97	-0.19
Mixed D	61.5	15.7	98.2	53.25	+0.09

D.4 Tool Contributions and Threshold Sensitivity

Contributions of individual tools. Tab. 15 reports score differences after removing each tool together with the statements that depend on it. The detector has the largest positive difference on Perception and Prediction, while TopoNet and scene classification have positive differences on Prediction and Planning. WildDet3D is positive on Planning and negative on the other tasks at this operating point. Negative values mean that the ablated condition scores higher; they are consistent with errors or distraction in the removed tool-dependent text, but this intervention does not identify the cause.

Perturbation design. The threshold study jointly varies the five parameters listed in Appendix A. In that order, each profile applies $\theta'_j = \theta_j(1 + 0.2d_j)$. The permissive and conservative profiles use $d = (-1, -1, 1, -1, -1)$ and its negative. Mixed A–D use $(-1, 0.5, 0, 1, -0.5)$, $(-0.5, -1, 1, 0, 0.5)$, $(0.5, 1, -0.5, -1, 0)$, and $(1, -0.5, -1, 0.5, 1)$, respectively. All changes remain within 20% of the fixed operating point, and the mixed profiles avoid testing only jointly permissive or jointly conservative settings.

Observed stability. Raw tool records change for 54.8–72.0% of samples, but only 9.0–19.4% of rendered prompts change (Tab. 16). Mean prompt-token Jaccard similarity remains at least 98.0%, and the mean content score over the five scene-state families stays within 0.22 points of the default. The renderer therefore attenuates these local parameter changes.

D.5 Failure Analysis

Failures can arise at each stage of the modular pipeline: visual tools may miss or misestimate scene elements, deterministic task rules may overinterpret incomplete evidence, and the answering model may misuse otherwise accurate records. The cases below separate these failure sources where the available traces permit.

Matching multiple facts to an option. Qwen2.5-VL-32B's Behavior accuracy falls from 33.0% with raw images to 16.5% with structured scene text, below the 25% four-class chance level. Its responses can quote the supplied ego direction and speed but select an option that matches only part of that information. In the audited run, 117 of 200 responses choose A, often following familiar direction or speed phrases without jointly checking them. This case demonstrates a downstream matching failure even when the response has accessed the relevant text.

Missing detections and overconfident inference. Question-conditioned routing can omit a tool needed to interpret an otherwise useful record. Fig. 8 illustrates this risk: the agentic context lacks the front-object evidence present in the rule-based context and infers a clear path. Absence from a selected tool's output does not establish absence in the scene. The example motivates distinguishing unobserved information from a verified negative observation when interpreting derived ego-state statements.

Closed action vocabulary. Collision scoring conflates action quality with expression in the benchmark's unprovided canonical vocabulary. The evaluated raw-image, caption, question-only, and structured conditions remain below 10% canonical accuracy. These low scores do not identify whether the model chose an unsafe action, paraphrased a valid one beyond the parser's coverage, or misunderstood the required answer. We therefore treat this result as an unresolved evaluation boundary, without assuming that vocabulary mismatch explains every failure.

E Additional Qualitative Results

Coverage under different routes. Fig. 8 compares the records selected for the same ego-behavior question. Both routes retain a straight-road description, but the rule-based route also includes front and side objects and their spatial and risk relations. The agentic route is shorter and infers a clear path from its reduced context. This trace makes the coverage tradeoff observable.

How temporal context changes an answer. Fig. 9 compares direct visual input, frame scene text, and video scene text for a question about not stopping at an intersection. The frame condition emphasizes current objects and possible conflicts and recommends slowing down. The video condition adds that the bus is pulling away and produces a different speed recommendation. The trace illustrates how temporal evidence changes the reasoning context, while the generated recommendation still requires scrutiny: the gap trend alone does not establish the correct explanation for the intersection decision.

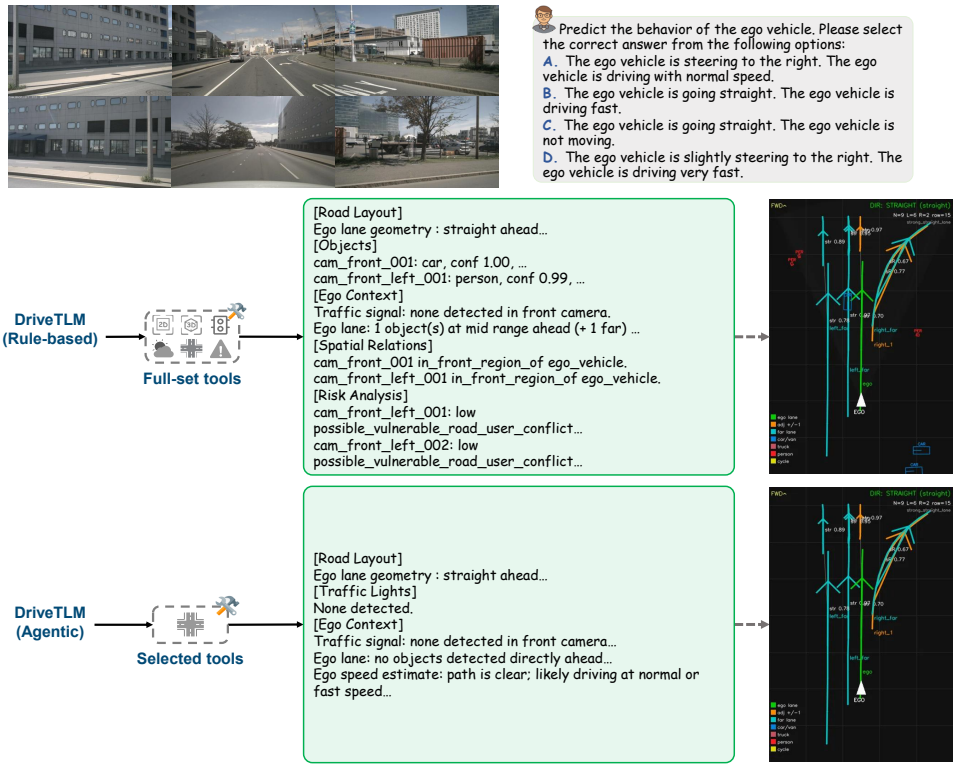

Figure 8: **Tool selection changes evidence coverage.** For the same Behavior question, the rule-based context contains object, relation, and risk records that are absent from the shorter agentic context. Both retain road geometry, but the agentic clear-path inference illustrates the risk of treating missing detections as confirmed absence.

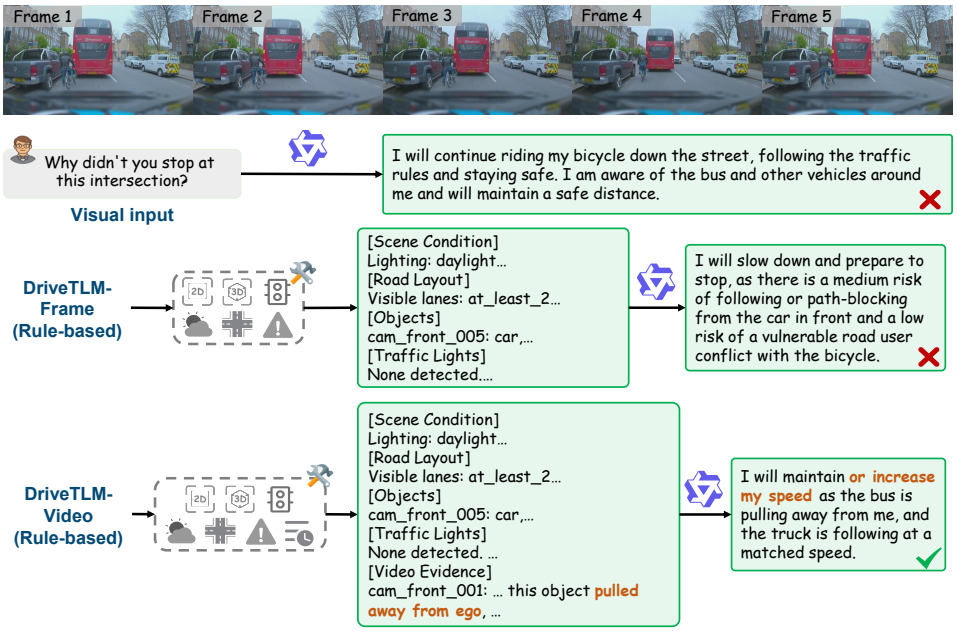

Figure 9: **A video trace with and without temporal evidence.** The frame condition emphasizes current objects and possible conflicts; the video condition additionally describes a bus pulling away and changes the generated speed recommendation. The example illustrates a change in the information used, rather than certifying the recommendation's correctness.

F LIMITATIONS AND FUTURE DIRECTIONS

The quality of structured scene text is naturally influenced by the accuracy of the underlying visual processing tools, and stronger or better-calibrated tools may further improve the evidence supplied to the answering model. The current Rule-based and Agentic tool selection strategies are designed for zero-shot use; a promising extension is to construct supervision for tool-use decisions and train a specialized router or agent for more accurate and efficient tool selection. Because this work focuses on the zero-shot setting and on understanding the capabilities enabled by structured scene text, we leave the construction of such training data and the training of learned tool-selection models to future work.