# AgentThink: A Unified Framework for Tool-Augmented Chain-of-Thought Reasoning in Vision-Language Models for Autonomous Driving

**Kangan Qian [1,3,*,‡], Sicong Jiang[2,*], Yang Zhong[3,*], Ziang Luo[1,3,*], Zilin Huang[4], Tianze Zhu[1], Kun Jiang[1,†], Mengmeng Yang[1,†], Zheng Fu[1], Jinyu Miao[1], Yining Shi[1], He Zhe Lim[1], Li Liu[3], Tianbao Zhou[3], Hongyi Wang[3], Huang Yu[3], Yifei Hu[3], Guang Li[3], Guang Chen[3], Hao Ye[3,†,‡], Lijun Sun[2], Diange Yang[1,†]**

[1]School of Vehicle and Mobility, Tsinghua University, [2]McGill University,
[3]Automotive and Robotics, Xiaomi Corporation, [4]University of Wisconsin-Madison

**Correspondence:** jiangkun@tsinghua.edu.cn, ydg@tsinghua.edu.cn

## Abstract

Vision-Language Models (VLMs) show promise for autonomous driving, yet their struggle with hallucinations, inefficient reasoning, and limited real-world validation hinders accurate perception and robust step-by-step reasoning. To overcome this, we introduce **AgentThink**, a pioneering unified framework that, for the first time, integrates Chain-of-Thought (CoT) reasoning with dynamic, agent-style tool invocation for autonomous driving tasks. AgentThink's core innovations include: **(i) Structured Data Generation**, by establishing an autonomous driving tool library to automatically construct structured, self-verified reasoning data explicitly incorporating tool usage for diverse driving scenarios; **(ii) A Two-stage Training Pipeline**, employing Supervised Fine-Tuning (SFT) with Group Relative Policy Optimization (GRPO) to equip VLMs with the capability for autonomous tool invocation; and **(iii) Agent-style Tool-Usage Evaluation**, introducing a novel multi-tool assessment protocol to rigorously evaluate the model's tool invocation and utilization. Experiments on the DriveLMM-o1 benchmark demonstrate AgentThink significantly boosts overall reasoning scores by **53.91%** and enhances answer accuracy by **33.54%**, while markedly improving reasoning quality and consistency. Furthermore, ablation studies and robust zero-shot/few-shot generalization experiments across various benchmarks underscore its powerful capabilities. These findings highlight a promising trajectory for developing trustworthy and tool-aware autonomous driving models. Code is available at https://github.com/curryqka/AgentThink.

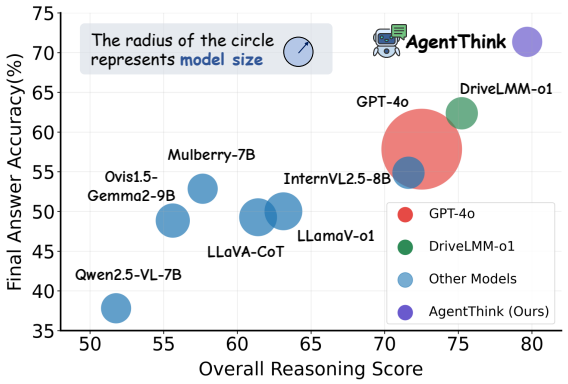


Figure 1: The performance of proposed AgentThink framework on the DriveLMM-o1 benchmark.

## 1 Introduction

Recent advances in foundation models have opened new opportunities for autonomous driving (Jiang et al., 2025b), where pretrained Large Language Models (LLMs) (Guo et al., 2025; Budzianowski and Vulić, 2019) and Vision-Language Models (VLMs) (Tian et al., 2024; Zhuang et al., 2025; Qian et al., 2025a) are increasingly employed to enable high-level scene understanding, commonsense reasoning, and decision making. These models aim to transcend traditional perception pipelines, which rely on hand-crafted components such as object detection (Qian et al., 2025c; Shi et al., 2024), motion prediction (Qian et al., 2024a; Wang et al., 2022), and rule-based planning (Cheng et al., 2024) by providing richer semantic representations and broader generalization grounded in web-scale knowledge.

Many recent approaches recast autonomous driving tasks as visual question answering (VQA) problems, applying supervised fine-tuning (SFT) to foundation VLMs with task-specific prompts for object identification, risk prediction, or motion planning (Sima et al., 2024; Marcu et al., 2024; Xu et al., 2024b; Ding et al., 2024; Wang et al., 2024a). This paradigm are widely used in offline pipelines to mine rare driving scenarios, such as identify-

Kangan Qian and Ziang Luo pursued this work during his master's degree at the School of Vehicle and Mobility, Tsinghua University, and finalized it during an internship at Xiaomi.

Zilin Huang did not receive any funding for this work.

* The authors contribute equally to this work.
†Corresponding author. ‡Project leader.



*Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10663–10682
November 4-9, 2025 

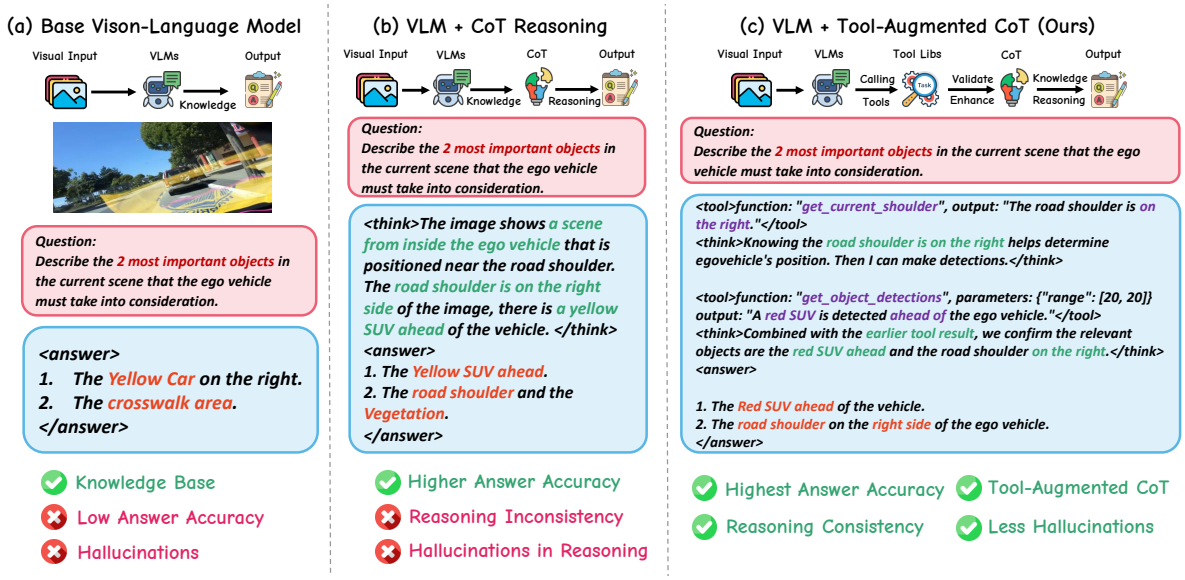


Figure 2: Illustration of the motivation and key highlights of our proposed framework. (a) Base VLMs use static input-output mapping with no reasoning, leading to low accuracy and frequent hallucinations. (b) VLM + CoT introduces structured reasoning, improving interpretability, but still suffers from inconsistencies and lack of verification. (c) AgentThink (Ours) augments CoT with dynamic tool use, enhancing accuracy, reducing hallucinations, and improving reasoning consistency through external verification.

ing intersections with a typical traffic signals or dense pedestrian activities. However, as shown in Fig.2 (a), these models typically treat reasoning as a static input-to-output mapping, neglecting the uncertainty, and verifiability essential to real-world decision making. Consequently, they often suffer from poor generalization, hallucinated outputs, and limited interpretability (Xie et al., 2025).

To improve robustness and transparency, recent works have explored incorporating chain-of-thought (CoT) reasoning into VLMs, as shown in Fig. 2 (b). Some approaches adopt rigid CoT templates (Tian et al., 2024; Hwang et al., 2024), promoting structured logic at the expense of flexibility. Others use open-ended reasoning formats (Nie et al., 2024; Ishaq et al., 2025), but may overfit to token patterns and exhibit shallow or redundant reasoning. Moreover, most existing methods rely purely on imitation learning from curated trajectories, lacking the ability to detect knowledge uncertainty or invoke tools for intermediate verification (Zhang et al., 2025; Qian et al., 2025a,b).

These challenges lead to a pivotal question: *How can a VLM truly function as a decision-making agent—cognizant of its knowledge boundaries, proficient in verification, and capable of learning from tool-guided feedback?* Inspiration comes from experienced human drivers who will consult aids like mirrors or GPS to refine their judgment when uncertain. Similarly, a capable autonomous agent must not only reason explicitly but also recognize its limitations and dynamically employ tools, such as object detectors or motion predictors, to steer its reasoning and decision-making.

Therefore, we present **AgentThink**, a unified framework for VLMs in autonomous driving that models reasoning not as a static output, but as an *agent-style process*—in which the model learns to utilize tools to generate Tool-Augmented reasoning chains, verify intermediate steps, and refine its conclusions. As illustrated in Fig.2 (c), rather than blindly mapping inputs to outputs, AgentThink dynamically decides when and how to use tools during inference to support or revise reasoning paths. To enable this behavior, we create a data-training-evaluation pipeline. First, we construct a structured dataset of Tool-Augmented reasoning traces. Then, we introduce a two-stage training pipeline: (i) SFT to warm up reasoning capabilities, and (ii) GRPO (Shao et al., 2024), a reinforcement learning (RL)-based strategy that refines reasoning depth and tool-use behavior through structured rewards. Finally, we propose a comprehensive evaluation protocol beyond answer correctness to assess tool selection, integration quality, and reasoning-tool alignment.

As shown in Fig. 1, experiments on the advanced DriveLMM-o1 benchmark (Ishaq et al., 2025) demonstrate that **AgentThink** achieves new state-of-the-art performance in both answer accuracy and reasoning score, surpassing existing mod-

els. The effectiveness of our approach in cultivating dynamic, tool-aware reasoning is further substantiated by comprehensive ablation studies and robust generalization capabilities across multiple benchmarks. These collective results strongly suggest that empowering vision-language agents with learned, dynamically invoked tool use is pivotal for creating more robust, interpretable, and generalizable systems for autonomous driving.

Generally, our contributions are as follows:

- Propose **AgentThink**, the first framework to integrate dynamic, **agent-style tool invocation** into vision-language reasoning for autonomous driving tasks. It significantly reduces hallucination in VQA and improves overall reasoning consistency by grounding each reasoning step in reliable tool outputs.
- Develop a scalable data generation pipeline that produces **structured, self-verified** data with **integrated tool usage** and reasoning chains.
- Introduce a two-stage training pipeline that combines **SFT** with **GRPO**, enabling models to learn when and how to invoke tools to enhance reasoning performance.
- Design new evaluation metrics tailored to **autonomous driving tool invocation**, capturing tool selection, integration quality, and reasoning-tool alignment.

## 2 Related Works

### 2.1 Language Models in Autonomous Driving

Recent advancements in language modeling have opened up new opportunities for autonomous driving, particularly in enabling interpretable reasoning, commonsense understanding, and decision-making (Cui et al., 2024). Early efforts integrated LLMs such as GPT series (OpenAI, 2023) by recasting driving tasks, e.g., scene description (Xu et al., 2024b; Mao et al., 2023a), decision-making (Fu et al., 2024; Wen et al., 2023), and risk prediction (Ma et al., 2024) into textual prompts, allowing for zero-shot or few-shot inference. While these approaches showcased the reasoning potential of LLMs, they often lacked step-by-step interpretability and struggled with generalization in out-of-distribution scenarios (Wang et al., 2023).

Recent works have augmented LLMs with prompting strategies, memory-based context construction, or vision inputs (Huang et al., 2024). For instance, DriveVLM (Tian et al., 2024) introduces a CoT approach with modules for scene description, analysis, and hierarchical planning, while DriveLM (Sima et al., 2024) focuses on graph-structured visual question answering. EMMA (Hwang et al.) demonstrates how multimodal models can directly map raw camera inputs to driving outputs, including trajectories and perception objects. Despite these advancements, both LLM-centric and VLM-based methods often treat reasoning as a static input-output mapping, with limited ability to detect uncertainty, perform intermediate verification, or incorporate physical constraints (Ishaq et al., 2025). Challenges such as hallucinations, over-reliance on rigid templates, and a lack of domain-specific reward feedback persist. To address these limitations, our work introduces a Tool-Augmented, RL-based reasoning framework that enables dynamic and verifiable decision-making for autonomous driving.

### 2.2 Visual Question Answering in Autonomous Driving

Visual Question Answering (VQA) for autonomous driving has emerged as a benchmark paradigm to evaluate perception, prediction, and planning capabilities. Benchmarks such as BDD-X (Kim et al., 2018), DriveBench (Xie et al., 2025), DriveMLLM (Guo et al., 2024), Nuscenes-QA (Qian et al., 2024b), and DriveLMM-o1 (Ishaq et al., 2025) provide structured QA tasks covering complex reasoning scenarios in urban and highway environments. For VQA tasks, recent approaches such as Reason2Drive (Nie et al., 2024), Alphadrive (Jiang et al., 2025a), OmniDrive (Wang et al., 2025), and DriveCoT (Wang et al., 2024b) introduce COT reasoning to enhance model interpretability.

However, many adopt rigid reasoning templates or rely solely on imitation learning, making them prone to overfitting and hallucination. These methods often overlook dynamic reasoning processes and fail to verify intermediate steps using external tools. In contrast, our framework combines structured data generation with step-level rewards and tool-verification during inference. By employing RL via GRPO, we optimize the model's reasoning trajectory to align with correctness, efficiency, and real-world applicability, setting a new direction for VQA in autonomous driving.

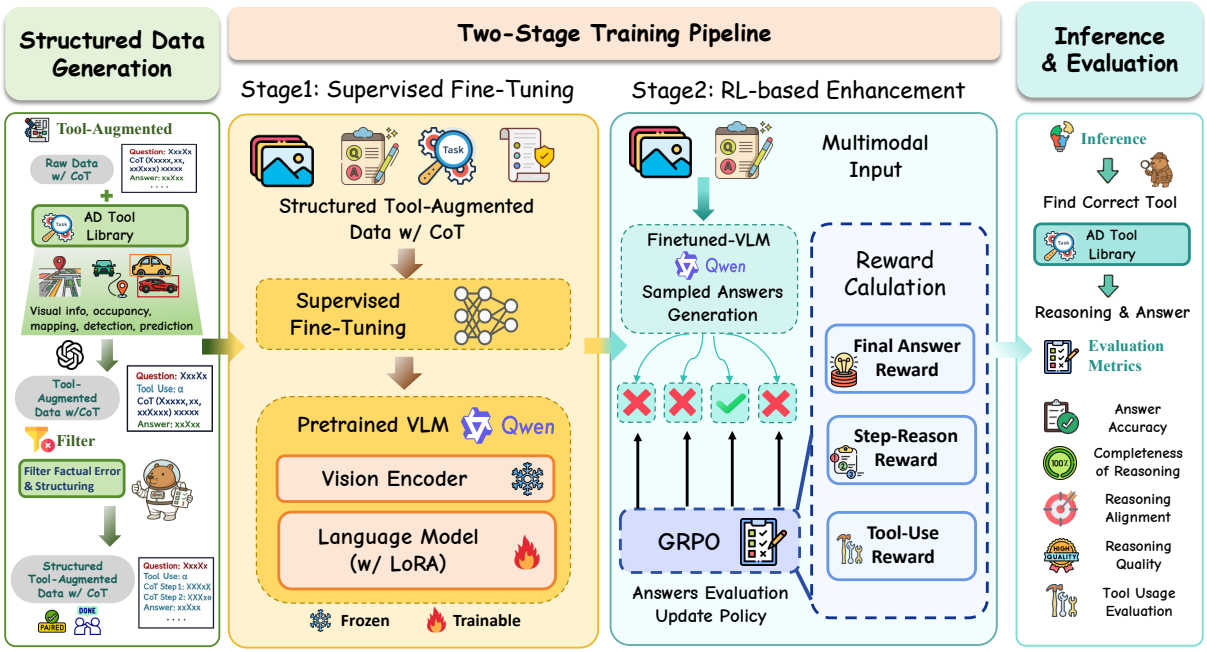


Figure 3: AgentThink Framework Architecture. (i) Structured and scalable data generation pipeline that constructs tool-augmented reasoning; (ii) Two-stage training pipeline that first performs SFT and then applies GRPO to improve reasoning and tool-use behavior; and (iii) Unified inference and evaluation protocol that dynamically invokes tools and assesses final answers based on reasoning completeness, consistency, and tool-use effectiveness.

## 3 Methodology

Our framework is designed to address three key challenges: (1) *How to generate reasoning data that incorporates tool usage?* (2) *How to equip the VLM with the capability to effectively utilize tools?* and (3) *How to enable reasoning-driven tool invocation during inference, and how to evaluate the model's ability to leverage tools for solving driving-related vision-question-answering tasks?*

Fig. 3 illustrates AgentThink's three key components: (i) a scalable pipeline for generating structured, tool-augmented reasoning data; (ii) a two-stage training pipeline with SFT and GRPO to improve reasoning and tool-use abilities; and (iii) a new evaluation methodology focused on assessing the model's effective tool utilization and its impact on reasoning.

### 3.1 Data Generation Pipeline

While prior studies (Wang et al., 2024a; Nie et al., 2024) have explored reasoning in VLMs, persistent hallucinations remain a challenge. We contend that reliable autonomous driving reasoning, akin to human decision-making, necessitates not only internal knowledge but also the ability to invoke external tools when needed. Addressing this, we introduce a Tool-Augmented data generation pipeline. Unlike existing datasets (Wang et al., 2023; Ishaq et al., 2025) focused solely on reasoning steps and final answers, our pipeline uniquely integrates explicit tool usage into the reasoning process.

**Tool library.** We developed a specialized tool library inspired by Agent-Driver (Mao et al., 2023b), featuring core modules for five driving-centric modules—*visual info*, *detection*, *prediction*, *occupancy*, and *mapping*, plus single-view vision utilities (open-vocabulary detection, depth, cropping, resizing). This is augmented by fundamental single-view vision tools like open-vocabulary object detectors and depth estimators. Together, these enable comprehensive environmental information extraction for diverse perception and prediction tasks. Details can be found in the Appendix A.

**Prompt Design.** Initial tool-integrated reasoning steps and answers are automatically generated using GPT-4o, guided by a prompt template (as shown in Fig. 3) designed to elicit a tool-augmented reasoning chain for instruction $\mathcal{L}$ rather than a direct answer.

Specifically, for a pretrained VLM $\pi_\theta$, an input image $\mathcal{V}$, and a task instruction $\mathcal{L}$, the reasoning step at time $t$ is generated as:

$$R_t = \pi_\theta(\mathcal{V}, \mathcal{L}, [R_1, \ldots, R_{t-1}]) \quad (1)$$

where $R_t$ denotes the $t$-th reasoning step, and $[R_1, \ldots, R_{t-1}]$ represents the previously generated

steps in the trajectory. The complete reasoning trajectory is denoted as $\mathcal{T}_R = (R_1, \ldots, R_M)$, where $M$ is the maximum number of reasoning steps.

Each reasoning step $R_t$ includes five key elements: the chosen tool ($Tool_i$), a generated sub-question ($Sub_i$), an uncertainty flag ($UF_i$), a guessed answer ($A_i$), and the next action choice ($AC_i$) such as *continue reasoning* or *conclude*. If internal knowledge suffices for $Sub_i$, $A_i$ is outputted and $UF_i = \texttt{False}$; otherwise, $UF_i = \texttt{True}$ and $A_i$ is left blank. This process is repeated to sample $N$ structured reasoning trajectories per QA pair.

**Data Assessment.** A separate LLM audits each data for factual accuracy and logical consistency, pruning samples with mismatched steps or unsupported conclusions. The result is a high-quality corpus that couples explicit tool use with coherent, verifiable reasoning.

### 3.2 Two-stage Training Pipeline

After constructing the structured dataset, we design a two-stage training pipeline to progressively enhance the model's reasoning capabilities and proficiency in tool usage.

#### 3.2.1 SFT-based Reasoning Warm-up

In the first phase, we perform SFT on the Tool-Augmented CoT dataset to warm up the model's ability to generate structured reasoning chains and appropriate tool calls. Each training sample is represented as $\tau = (\mathcal{V}, \mathcal{L}, \mathcal{T}_R, \mathcal{A})$, where $\mathcal{V}$ is the visual input, $\mathcal{L}$ is the language instruction, $\mathcal{T}_R$ is the step-by-step reasoning trace including both reasoning steps and explicit tool invocation (e.g., `Tool(name, params)`), and $\mathcal{A}$ is the final answer. The objective is to maximize the likelihood of generating correct reasoning and action sequences:

$$\mathcal{L}_{\text{SFT}}^{(1)} = -\mathbf{E}_{\tau\sim\mathcal{D}} \sum_{t=1}^{T} \log \pi_\theta(R_t \mid \mathcal{V}, \mathcal{L}, R_{<t}), \tag{2}$$

where only the generation of reasoning steps and tool calls (action type and parameters) is optimized, while outputs from environment responses (e.g., tool returns) are masked out during loss computation. This phase teaches the model *what tools to use* and *how to configure them* based on the driving context.

In the second phase, we refine the model with full context exposure, including actual tool outputs in the reasoning chain. Training samples now include observed environment feedback $\mathcal{O}$ (e.g., API results, detected objects, or retrieved text), forming an enhanced tuple $(\mathcal{V}, \mathcal{L}, \mathcal{T}_R \cup \mathcal{O}, \mathcal{A})$. The loss remains maximum likelihood but now covers both action generation and observation grounding:

$$\mathcal{L}_{\text{SFT}}^{(2)} = -\mathbf{E}_{\tau\sim\mathcal{D}} \sum_{t=1}^{T'} \log \pi_\theta(z_t \mid \mathcal{V}, \mathcal{L}, z_{<t}), \tag{3}$$

where $z_t$ denotes tokens in the extended sequence containing reasoning, actions, and observed outcomes. This joint modeling of actions and observations enables the model to learn an implicit understanding of expected tool outputs, thereby grounding subsequent reasoning in real-world feedback. This two-phase warm-up prepares the model for robust reasoning and effective tool integration prior to RL fine-tuning.

#### 3.2.2 RLFT-based Reasoning Enhancement

To further optimize the model beyond imitation learning, we adopt Reinforcement Learning Fine-Tuning (RLFT) using GRPO, which effectively leverages structured rewards without relying on a learned critic.

**GRPO Overview.** GRPO avoids the need for a value function by computing the relative advantage of each sample within a group. Given a question $q$ and $G$ responses $\{o_i\}_{i=1}^{G}$ sampled from the old policy $\pi_{\theta_{\text{old}}}$, the GRPO objective is (Shao et al., 2024):

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}\sim\pi_{\text{old}}} \left[ \frac{1}{G} \sum_{i=1}^{G} \mathcal{L}_i - \beta\, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right] \tag{4}$$

where the group-wise clipped loss is defined as:

$$\mathcal{L}_i = \min\left(w_i A_i,\ \text{clip}(w_i,\ 1-\epsilon,\ 1+\epsilon) A_i\right) \tag{5}$$

and the importance weight $w_i$ and normalized advantage $A_i$ are given by:

$$w_i = \frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)} \tag{6}$$

$$A_i = \frac{r_i - \text{mean}(r)}{\text{std}(r)} \tag{7}$$

where $r_i$ denotes the reward assigned to output $o_i$, and $\beta$ and $\epsilon$ are tunable hyperparameters.

**Reward Design.** To guide the model toward accurate, interpretable, and tool-aware reasoning, we design a structured reward function with three major components:

| Reward Type | Description |
|---|---|
| Final Answer Reward | Verifies final answer against ground truth; promotes task-level correctness. |
| Step Reasoning Reward | Evaluates intermediate step logic & structure. Sub-rewards for:<br>(i) Step Matching: Align with reference steps and penalize incorrect ordering.<br>(ii) Coherence: Smooth, logical transitions between steps. |
| Tool Use Reward | Promotes appropriate & meaningful tool usage. Sub-rewards for:<br>(i) Format Compliance: Adherence to expected output structure (e.g., "Tool", "Step Reasoning").<br>(ii) Integration Quality: Effective coherent incorporation of tool outputs into reasoning. |

Table 1: GRPO reward for tool-augmented reasoning

This structured reward design provides more targeted and interpretable supervision than generic similarity metrics. It enables GRPO to optimize both the quality of the reasoning process and the model's ability to invoke tools when needed.

### 3.3 Inference and Evaluation

During inference, as illustrated in Fig. 4, the VLM dynamically accesses tools from a predefined library to acquire relevant information, enabling step-by-step reasoning. This dynamic tool-calling mechanism not only improves answer accuracy but also mirrors the structure of our tool-augmented training data. However, existing benchmarks (Guo et al., 2024; Ishaq et al., 2025) do not evaluate tool-usage capabilities. To bridge this gap, we introduce three metrics (Table 2) that assess the model's tool utilization during reasoning, leveraging the LLM-as-Judge paradigm (Jiang et al., 2025b).

| Metric | Description |
|---|---|
| Tool Usage Appropriateness | Assesses whether tools are logically selected and meaningfully used to support individual reasoning steps. |
| Tool Chain Coherence | Evaluates whether the sequence of tool invocations is clear, logically ordered, and efficiently contributes to reasoning. |
| Perception-Guided Tool Alignment | Measures how well tool usage aligns with multimodal inputs, including visual observations and scene context. |

Table 2: Evaluation metrics for tool-use in reasoning

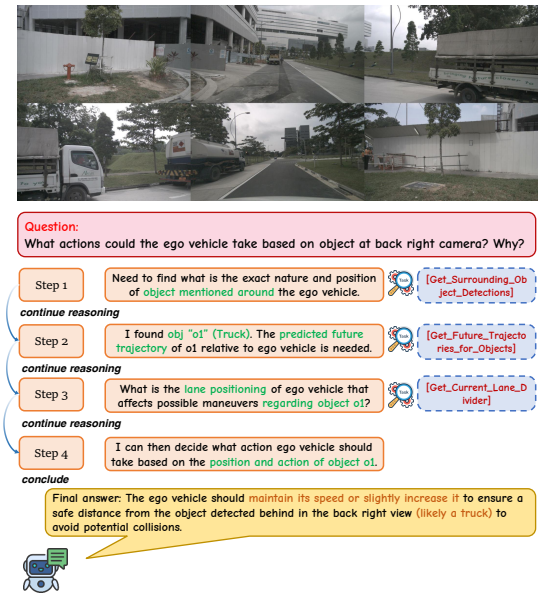


Figure 4: The model generates structured reasoning chains, dynamically invokes external tools to resolve uncertainties (e.g., object detection, trajectory prediction, lane width), and concludes with an interpretable action recommendation.

## 4 Experiments

In this section, we conduct extensive experiments to validate the effectiveness of AgentThink. Our experiments are designed to answer the following core questions:

**Q1.** Can dynamic Tool-Augmented reasoning improve both final answer accuracy and reasoning consistency over existing VLM baselines?

**Q2.** Does our structured reward design (Final Answer, Step-wise Reason, Tool-use) contribute meaningfully to reasoning behavior?

**Q3.** How well does AgentThink generalize to unseen datasets under zero-shot settings?

**Evaluation Metrics.** We leverage DriveLMM-o1's evaluation metrics, specifically utilizing the overall reasoning score to gauge the reasoning of VLMs, and employing multiple choice quality (MCQ) to assess the accuracy of the final answers, with further details provided in the Appendix C. Additionally, we introduce new metrics to evaluate tool-use capability as described in Table 2.

**Model and Implementation.** We employ Qwen2.5-VL-7B as our base model and keep its vision encoder frozen. Supervised fine-tuning (SFT) is applied via LoRA for 20 epochs with a learning rate of $1 \times 10^{-4}$, followed by GRPO fine-tuning

| Model | Driving Metrics (%) ↑ | | | Scene Detail (%) ↑ | | Overall(%) ↑ | |
|---|---|---|---|---|---|---|---|
| | Risk Assess. | Rule Adh. | Scene Aware. | Relevance | Missing | Reason. | MCQ |
| GPT-4o (Islam and Moushi, 2024) | 71.32 | 80.72 | 72.96 | 76.65 | 71.43 | 72.52 | 57.84 |
| Ovis1.5-Gemma2-9B (Lu et al., 2024) | 51.34 | 66.36 | 54.74 | 55.72 | 55.74 | 55.62 | 48.85 |
| Mulberry-7B (Yao et al., 2024) | 51.89 | 63.66 | 56.68 | 57.27 | 57.45 | 57.65 | 52.86 |
| LLaVA-CoT (Xu et al., 2024a) | 57.62 | 69.01 | 60.84 | 62.72 | 60.67 | 61.41 | 49.27 |
| LlamaV-o1 (Thawakar et al., 2025) | 60.20 | 73.52 | 62.67 | 64.66 | 63.41 | 63.13 | 50.02 |
| InternVL2.5-8B (Chen et al., 2024) | 69.02 | 78.43 | 71.52 | 75.80 | 70.54 | 71.62 | 54.87 |
| Qwen2.5-VL-7B (Bai et al., 2025) | 46.44 | 60.45 | 51.02 | 50.15 | 52.19 | 51.77 | 37.81 |
| DriveLMM-o1 (Ishaq et al., 2025) | 73.01 | 81.56 | 75.39 | 79.42 | 74.49 | 75.24 | 62.36 |
| **AgentThink (Ours)** | **80.51** | **84.98** | **82.11** | **84.99** | **79.56** | **79.68** | **71.35** |

Table 3: Evaluation results on the DriveLMM-o1 benchmark. AgentThink significantly improves reasoning and answer accuracy across all categories by leveraging dynamic Tool-Augmented reasoning.

| Model Variant | SFT | GRPO Reward Setting | | | Driving Metrics (%) ↑ | | | Scene Detail (%) ↑ | | Overall (%) ↑ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Setting | Answer | Step R. | Tool Use | Risk Assess. | Rule Adh. | Obj Und. | Relevance | Missing | Reason. | MCQ |
| Base Model | ✗ | ✗ | ✗ | ✗ | 46.44 | 60.45 | 51.02 | 50.15 | 52.19 | 51.77 | 37.81 |
| + SFT | ✓ | ✗ | ✗ | ✗ | 70.25 | 79.83 | 75.41 | 81.45 | 71.68 | 72.54 | 62.95 |
| + GRPO | ✗ | ✓ | ✗ | ✗ | 69.25 | 75.41 | 71.58 | 75.86 | 68.05 | 69.41 | 61.41 |
| + GRPO† | ✗ | ✗ | ✓ | ✗ | 69.29 | 75.43 | 72.66 | 76.77 | 69.03 | 69.43 | 57.19 |
| + SFT + GRPO | ✓ | ✓ | ✓ | ✗ | 71.00 | 77.35 | 73.23 | 78.13 | 69.08 | 70.83 | 64.58 |
| **AgentThink (Ours)** | ✓ | ✓ | ✓ | ✓ | **80.51** | **84.98** | **82.11** | **84.99** | **79.56** | **79.68** | **71.35** |

Table 4: Ablation study of AgentThink on reward design and training strategy. The full model (bottom row) benefits from the combination of SFT, GRPO, and structured tool-use rewards. Ablation models here trained with 8 epochs.

| **Model** | Tool Usage Appro. | Tool Chain Coh. | Percep-Guided Tool Align. | Overall Tool Score |
|---|---|---|---|---|
| Base + DirectTool | 59.61 | 73.29 | 69.71 | 67.54 |
| Base + SFT | 62.38 | 78.19 | 75.78 | 72.12 |
| Base + GRPO | 68.44 | 80.73 | 80.82 | 76.66 |
| **AgentThink (Ours)** | **70.92** | **82.16** | **84.25** | **79.11** |

Table 5: Tool Evaluation Results of Different Qwen2.5-VL-7B Variants.

for 8 epochs with a learning rate of $1 \times 10^{-5}$. The training batch size is set to 1 per device. We use the `bfloat16` data type to improve computational efficiency. All experiments are conducted on 16× NVIDIA A800 GPUs. During the GRPO phase, we perform 2 rollouts per question. Additional implementation details are provided in Appendix B.

### 4.1 Main Experiment Results

**Comparison with Open-Source VLMs.** Table 3 presents the main results on the DriveLMM-o1 benchmark, comparing AgentThink with a range of strong open-source VLM models, including DriveLMM-o1(Ishaq et al., 2025), InternVL2.5 (Chen et al., 2024), LLaVA-CoT (Xu et al., 2024a), and Qwen2.5-VL variants.

Our full model, AgentThink, achieves state-of-the-art performance across all categories. It surpasses the baseline Qwen2.5-VL-7B by a wide margin, improving the overall reasoning score from 51.77 to 79.68 (+51.9%), and final answer accuracy from 37.81% to 71.35% (+33.5%). Compared to the strongest prior system, DriveLMM-o1, which already integrates some reasoning abilities, AgentThink further improves by +5.9% in reasoning and +9.0% in final answer accuracy, demonstrating the advantage of learned tool-use over static CoT or imitation-based methods.

**Performance Breakdown.** In addition to reasoning and accuracy, AgentThink consistently outperforms others in driving-specific metrics (risk assessment, traffic rule adherence, and scene understanding), as well as perception-related categories (relevance and missing detail detection). These gains reflect its ability to leverage dynamic tool invocation and feedback to ground its reasoning more effectively in visual context.

**Key Insight.** Unlike traditional CoT or prompt-based methods, AgentThink learns *when* and *why* to invoke external tools, enabling more adaptive and context-aware reasoning. This leads to better decision quality, fewer hallucinations, and improved trustworthiness in safety-critical driving scenarios. We provide the case in the Appendix D.

### 4.2 Tool-Use Analysis

We analyze how different training strategies influence tool-use behavior during reasoning. Table 5 reports results on these three dimensions: (1) Tool usage appropriateness, (2) Tool chain coherence, and (3) Perception-guided alignment.

The DirectTool baseline, which enforces tool in-

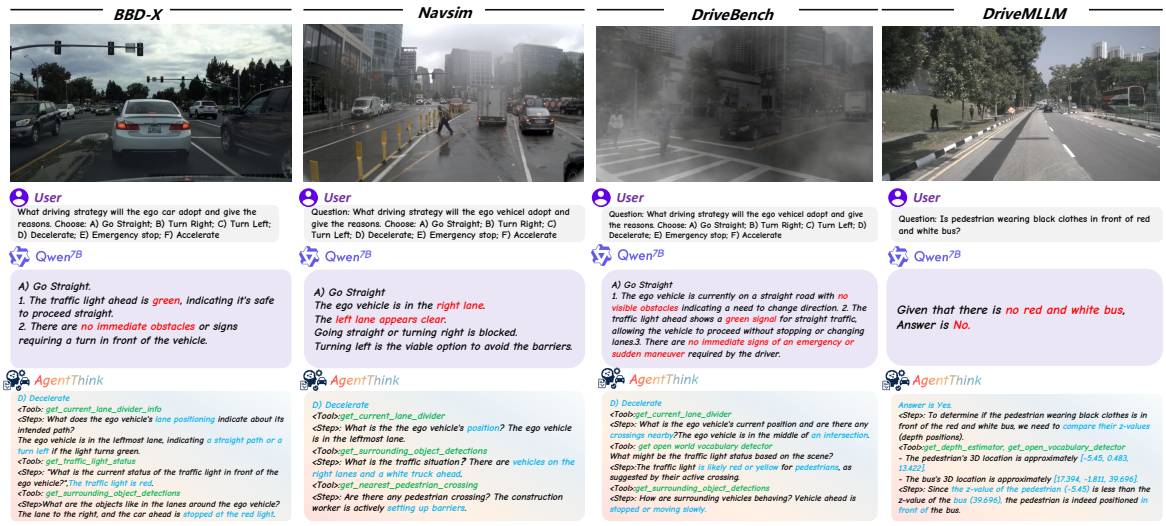


Figure 5: Zero-shot qualitative comparison with Qwen2.5VL-7B on BDD-X, Navsim, DriveBench and DriveMLLM.

vocation via prompt without reasoning structure, shows moderate chain coherence but lower appropriateness and alignment—indicating that forced tool use often lacks purpose. Adding SFT improves both appropriateness and alignment, but lacks feedback on tool quality, limiting further gains. GRPO with structured rewards leads to significant improvements, teaching the model to invoke tools selectively and integrate outputs coherently. Our full model, combining SFT and GRPO with full reward, achieves the best performance across all metrics. This demonstrates that both supervision and reward shaping are essential for learning effective, context-aware tool use. We also evaluate the impact of the training data scale, as detailed in Appendix E.

### 4.3 Ablation Study

In Table 4, we conduct comprehensive ablations to examine the effect of different reward signals and training stages in AgentThink. Using SFT or GRPO alone with either final-answer or step-reasoning reward provides moderate gains over the base model, improving task accuracy and reasoning coherence respectively. However, their effects are limited when applied in isolation.

We find that combining SFT with GRPO (without the tool-use reward) delivers better performance, which shows that warm-up reasoning is crucial before reinforcement tuning. Our complete AgentThink model, which incorporates all three reward components, attains the optimal results. It greatly enhances both reasoning quality and answer accuracy, thus emphasizing the importance of using tools and grounding reasoning in visual context.

### 4.4 Generalization Evaluation

We assessed AgentThink's generalization ability on a new DriveMLLM benchmark under zero-shot and one-shot settings against a range of strong baselines, including prominent VLMs and task-specific variants (details in Table 6). The evaluation metrics are detailed in Appendix F.

AgentThink achieves state-of-the-art performance with **zero-shot** (26.52) and **one-shot** (47.24) scores, surpassing GPT-4o and LLaVA-72B. While baseline methods like `DirectTool` demonstrate strong perception task results (*e.g.*, `RHD` 89.2 vs. 86.1, `BBox` precision 92.4% vs. 91.7%) through hard-coded tool prompts, they suffer from contextual rigidity and fragmented reasoning-perception alignment. Our model demonstrates a superior balance by effectively coordinating explicit reasoning with learned, adaptive tool use grounded in perceptual context. This underscores the advantages of its learned tool-use mechanism over static prompting or sheer model scale for robust generalization.

Shown in Table 7, in the zero-shot experiments on DriveBench (Xie et al., 2025), AgentThink achieved state-of-the-art performance on 3 out of 5 tasks (External, Sensor, and Transmission) and ranked within the top 3 on the remaining two (Weather and Motion). This demonstrates its strong generalization ability, showing that it can maintain high performance across diverse unseen tasks and scenarios, surpassing both general-purpose and specialist models in Driving VQA.

Qualitatively, as illustrated in Fig. 5, Agent-

| Type | Model | Metric Scores (%) ↑ | | | | | | | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | L/R | F/B | RHD | RD | PPos | BBox | CVD | CD | AccS | |
| Zero-shot | GPT-4o (Islam and Moushi, 2024) | 91.72 | 67.60 | 9.58 | 14.69 | 40.90 | 4.07 | 46.11 | 70.65 | 43.16 | 25.63 |
| | GPT-4o-mini | 67.67 | 50.13 | 70.44 | 0.00 | 29.28 | 3.78 | 0.00 | 46.40 | 33.46 | 16.68 |
| | LLaVA-ov-72B (Li et al., 2024) | 85.42 | 49.48 | 13.76 | 45.27 | 16.46 | 0.00 | 42.97 | 27.09 | 35.06 | 21.10 |
| | Qwen2.5-VL-7B (Bai et al., 2025) | 76.55 | 55.24 | 7.14 | 17.11 | 55.97 | 38.31 | 55.94 | 51.52 | 44.72 | 13.36 |
| | Qwen + CoT | **87.06** | **63.09** | 16.69 | 22.56 | 52.51 | 38.87 | **76.90** | 38.71 | 49.55 | 19.31 |
| | Qwen + DirectTool | 78.95 | 48.96 | **58.43** | **67.57** | 58.20 | 42.22 | 51.76 | 51.38 | **57.18** | 24.05 |
| | **AgentThink (Ours)** | 82.33 | 54.40 | 56.14 | 61.45 | **70.45** | **56.23** | 23.09 | **51.60** | 56.96 | **26.52** |
| One-shot | GPT-4o | 91.08 | 69.37 | 36.51 | 71.17 | 42.44 | 5.10 | 0.00 | 63.88 | 47.44 | 33.17 |
| | GPT-4o-mini | 66.00 | 48.95 | 83.02 | 58.47 | 25.71 | 3.97 | 52.73 | 55.23 | 49.26 | 22.13 |
| | LLaVA-ov-72B(Li et al., 2024) | 79.12 | 62.97 | 49.26 | 68.04 | 28.57 | 2.20 | 53.12 | 60.90 | 50.52 | 36.66 |
| | Qwen2.5-VL-7B (Bai et al., 2025) | 80.30 | 53.14 | 36.96 | 39.13 | 62.69 | 22.63 | 49.88 | 48.32 | 49.13 | 33.53 |
| | Qwen + CoT | **86.35** | **59.95** | 43.29 | 31.81 | 53.64 | 26.93 | 51.02 | 42.30 | 49.41 | 32.06 |
| | Qwen + DirectTool | 84.57 | 55.50 | **67.32** | 59.54 | **85.58** | 26.07 | **52.34** | **53.25** | 60.52 | 42.27 |
| | **AgentThink (Ours)** | 78.71 | 48.46 | 60.64 | **60.71** | 72.36 | **64.46** | 52.26 | 52.04 | **61.21** | **47.24** |

Table 6: Zero-shot and one-shot performance comparison across multiple metrics on the DriveMLLM benchmark.

| Model | Size | Type | BenchDrive Scores (%) ↑ | | | | |
|---|---|---|---|---|---|---|---|
| | | | Weather | External | Sensor | Motion | Transmission |
| GPT-4o (Islam and Moushi, 2024) | — | Commercial | 57.2 | 29.3 | 44.3 | 34.3 | 36.8 |
| LLaVA-1.5 (Li et al., 2024) | 7 B | Open | 69.7 | 26.5 | 18.8 | 71.3 | 10.2 |
| LLaVA-1.5 | 13 B | Open | 61.6 | 15.5 | 24.1 | **79.8** | 15.5 |
| LLaVA-NeXT | 7 B | Open | 69.7 | 48.5 | 21.8 | 66.0 | 11.8 |
| InternVL2 (Chen et al., 2024) | 8 B | Open | 59.9 | 50.8 | 29.9 | 68.3 | 30.0 |
| Phi-3 | 4.2 B | Open | 40.0 | 25.0 | 16.8 | 31.3 | 27.7 |
| Phi-3.5 | 4.2 B | Open | 60.6 | 21.3 | 25.6 | 33.0 | 39.7 |
| Qwen2-VL (Bai et al., 2025) | 7 B | Open | **76.7** | 37.5 | 22.8 | 57.0 | 35.8 |
| Qwen2-VL | 72 B | Open | 59.8 | 45.5 | 52.3 | 58.3 | 44.8 |
| DriveLM (Sima et al., 2024) | 7 B | Specialist | 21.2 | 21.3 | 9.0 | 22.3 | 17.5 |
| Dolphins (Ma et al., 2024) | 7 B | Specialist | 54.3 | 30.0 | 9.4 | 9.3 | 21.5 |
| **AgentThink (Ours)** | 7 B | Ours | 64.8 | **68.2** | **56.8** | 71.8 | **61.2** |

Table 7: DriveBench results. AgentThink achieves SOTA on 3/5 tasks (External, Sensor, Transmission), and ranks top-3 on the remaining 2 (Weather, Motion), indicating strong generalization.

Think successfully navigates challenging zero-shot corner cases on diverse benchmarks (BDD-X (Kim et al., 2018), Navsim (Dauner et al., 2024), DriveBench (Xie et al., 2025), DriveMLLM (Guo et al., 2024)). In these cases, the base Qwen model often fails to gather sufficient information or generates hallucinates during reasoning, leading to incorrect outputs. In contrast, AgentThink adeptly invokes tools to acquire critical decision-making information, thereby correctly answering these difficult questions. This further highlights the practical benefits of its dynamic, tool-augmented reasoning in unfamiliar contexts.

## 5 Conclusion

We present **AgentThink**, the first unified framework that tightly fuses CoT reasoning with agent-style tool invocation for autonomous driving. With a scalable tool-augmented dataset and a two-stage SFT with GRPO pipeline, AgentThink raises DriveLMM-o1 reasoning score from **51.77** to **79.68** and answer accuracy from **37.81%** to **71.35%**, outperforming the strongest prior model by +5.9% and +9.0%. Beyond improved performance, AgentThink demonstrates stronger interpretability by making each reasoning step grounded in tool outputs. Notably, as a driving-scene VQA system, AgentThink aligns with industrial practices where such models are used off the control loop, for example, for mining corner cases or providing high-level feedback in a dual-system setup, and thus operating under relaxed latency constraints compared to real-time planning. Results validate that coupling explicit reasoning with learned tool use is a promising path toward safer, more robust language-model-centric driving tasks. We believe this framework lays the foundation for building trustworthy VLM-based agents capable of generalizing to complex, dynamic real-world driving environments.

## Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (52372414, 52394264, 52472449). Zilin Huang did not receive any funding for this work. We extend our gratitude to Xiaomi for their contribution to this work. Xiaomi generously provided computational resources that significantly accelerated our experiments and model training. Additionally, their supportive internship environment enabled close collaboration between academic and industrial researchers, fostering innovation in real-world autonomous driving challenges.

## Limitations

**Data scale.** Our tool-augmented corpus totals 18k annotated instances, limiting exposure to long-tail or rare driving events. A substantially larger and more diverse dataset is required for the model to internalize a broader spectrum of real-world scenarios.

**Model size.** Ours relies on *qwen2.5-VL-7B*; the 7B-parameter footprint incurs non-trivial memory and latency overhead on embedded automotive hardware. Future work should investigate lighter backbones (e.g., ~3B) that preserve reasoning capability while easing on-board resource constraints.

**Lack of temporal context.** The model under discussion processes single-frame, multi-view images as inputs. However, in the absence of sequential information, it may misinterpret scenarios that rely on temporal cues, such as changing traffic lights. To address this issue, incorporating video tokens or employing recurrent memory could be effective solutions.

**Missing 3-D modalities.** The absence of LiDAR or point-cloud data deprives the model of precise spatial geometry, introducing uncertainty in distance-related reasoning. Fusing additional modalities is expected to enhance robustness.

## Ethics Statement

All data come from publicly released driving datasets that anonymise personally identifiable information; no private or crowd-sourced data were collected. The study involves no human subjects, and every experiment is run offline or in simulation. Model checkpoints are released under a non-commercial licence that prohibits deployment in safety-critical vehicles without additional validation. The work follows the ACL Code of Ethics and does not rely on sensitive data or models.

## References

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, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. *arXiv preprint arXiv:1907.05774*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Jie Cheng, Yingbing Chen, and Qifeng Chen. 2024. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint arXiv:2404.14327*.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, and 1 others. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979.

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, and 1 others. 2024. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719.

Xinpeng Ding, Jianhua Han, Hang Xu, Xiaodan Liang, Wei Zhang, and Xiaomeng Li. 2024. Holistic autonomous driving understanding by bird's-eye-view injected multi-modal large models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13668–13677.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in

llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Zilin Huang, Zihao Sheng, Yansong Qu, Junwei You, and Sikai Chen. 2024. Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving. *arXiv preprint arXiv:2412.15544*.

JJ Hwang, R Xu, H Lin, WC Hung, J Ji, K Choi, D Huang, T He, P Covington, B Sapp, and 1 others. Emma: End-to-end multimodal model for autonomous driving. arxiv 2024. *arXiv preprint arXiv:2410.23262*.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, and 1 others. 2024. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*.

Ayesha Ishaq, Jean Lahoud, Ketan More, Omkar Thawakar, Ritesh Thawkar, Dinura Dissanayake, Noor Ahsan, Yuhao Li, Fahad Shahbaz Khan, Hisham Cholakkal, and 1 others. 2025. Drivelmm-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. *arXiv preprint arXiv:2503.10621*.

Raisa Islam and Owana Marzia Moushi. 2024. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*.

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

Sicong Jiang, Zilin Huang, Kangan Qian, Ziang Luo, Tianze Zhu, Yang Zhong, Yihong Tang, Menglin Kong, Yunlong Wang, Siwen Jiao, and 1 others. 2025b. A survey on vision-language-action models for autonomous driving. *arXiv preprint arXiv:2506.24044*.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. 2018. Textual explanations for self-driving vehicles. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Shiyin Lu, Yang Li, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, and Han-Jia Ye. 2024. Ovis: Structural embedding alignment for multimodal large language model. *arXiv preprint arXiv:2405.20797*.

Yingzi Ma, Yulong Cao, Jiachen Sun, Marco Pavone, and Chaowei Xiao. 2024. Dolphins: Multimodal language model for driving. In *European Conference on Computer Vision*, pages 403–420. Springer.

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

Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. 2023b. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*.

Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, and 1 others. 2024. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pages 252–269. Springer.

Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. 2024. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *European Conference on Computer Vision*, pages 292–308. Springer.

OpenAI. 2023. Gpt-4 technical report. https://arxiv.org/abs/2303.08774.

Kangan Qian, Ziang Luo, Sicong Jiang, Zilin Huang, Jinyu Miao, Zhikun Ma, Tianze Zhu, Jiayin Li, Yangfan He, Zheng Fu, and 1 others. 2025a. Fasionad++: Integrating high-level instruction and information bottleneck in fat-slow fusion systems for enhanced safety in autonomous driving with adaptive feedback. *arXiv preprint arXiv:2503.08162*.

Kangan Qian, Ziang Luo, Sicong Jiang, Zilin Huang, Jinyu Miao, Zhikun Ma, Tianze Zhu, Jiayin Li, Yangfan He, Zheng Fu, and 1 others. 2025b. Fasionad++: Integrating high-level instruction and information bottleneck in fat-slow fusion systems for enhanced safety in autonomous driving with adaptive feedback. *arXiv preprint arXiv:2503.08162*.

Kangan Qian, Jinyu Miao, Xinyu Jiao, Ziang Luo, Zheng Fu, Yining Shi, Yunlong Wang, Kun Jiang, and Diange Yang. 2024a. Priormotion: Generative class-agnostic motion prediction with raster-vector motion field priors. *arXiv preprint arXiv:2412.04020*.

Kangan Qian, Jinyu Miao, Ziang Luo, Zheng Fu, Yining Shi, Yunlong Wang, Kun Jiang, Mengmeng Yang, Diange Yang, and 1 others. 2025c. Lego-motion: Learning-enhanced grids with occupancy instance modeling for class-agnostic motion prediction. *arXiv preprint arXiv:2503.07367*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yining Shi, Kun Jiang, Ke Wang, Jiusi Li, Yunlong Wang, Mengmeng Yang, and Diange Yang. 2024. Streamingflow: Streaming occupancy forecasting with asynchronous multi-modal data streams via neural ordinary differential equation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14833–14842.

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

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, and 1 others. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. 2024. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. 2024a. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. 2025. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. *arXiv preprint arXiv:2504.04348*.

Tianqi Wang, Enze Xie, Ruihang Chu, Zhenguo Li, and Ping Luo. 2024b. Drivecot: Integrating chain-of-thought reasoning with end-to-end driving. *arXiv preprint arXiv:2403.16996*.

Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, and 1 others. 2023. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*.

Yunlong Wang, Hongyu Pan, Jun Zhu, Yu-Huan Wu, Xin Zhan, Kun Jiang, and Diange Yang. 2022. Besti: Spatial-temporal integrated network for class-agnostic motion prediction with bidirectional enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17093–17102.

Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. 2023. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. 2025. Are vlms ready for autonomous driving? an empirical study from the reliability, data, and metric perspectives. *arXiv preprint arXiv:2501.04003*.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024a. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

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

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and 1 others. 2024. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2025. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26183–26191.

# Appendix

## A Tool Library

This tool library is designed to support autonomous driving function calls by providing a comprehensive set of tools for various tasks including visual detection, object detection, trajectory prediction, map information querying, and more. Below is an introduction to the main categories and specific tools available in the library:

**Visual info Functions**

- **get_open_world_vocabulary_detection**: Given a list of object words, this function detects the objects in the image and returns their 2D positions and sizes within the camera coordinate system. If no related object is found, it returns None.
- **get_3d_loc_in_cam**: Given an input image and object-related keywords, this function calculates the depth value for each pixel and determines the 3D locations of the specified objects within the camera coordinate system.
- **Resize_image_info**: defines a function to resize an image to specified dimensions, supporting various interpolation methods. It requires the input image path, output path, and target size with, interpolation method as an optional parameter.
- **Crop_image_info**: defines a function to crop a rectangular region from an image. It requires the input image path, output path, and the coordinates of the crop region.

**Detection Functions**

- **get_leading_object_detection**: Detects the leading object, returning its ID, position, and size. If no leading object exists, it returns None.
- **get_surrounding_object_detections**: Detects surrounding objects within a 20m×20m range, providing a list of object IDs along with their positions and sizes. Returns None if no surrounding objects are present.
- **get_front_object_detections**: Identifies objects within a 10m×20m range in front of the vehicle, returning their IDs, positions, and sizes. Returns None if no such objects exist.
- **get_object_detections_in_range**: Detects objects within a specified range (x_start, x_end)×(y_start, y_end)m², returning a list of object IDs and their corresponding positions and sizes. Returns None if no objects are found in the range.
- **get_all_object_detections**: Retrieves detections for all objects in the entire scene, providing a list of object IDs and their positions and sizes. However, it is recommended to avoid using this function if more specific alternatives are available.

**Prediction Functions**

- **get_leading_object_future_trajectory**: Provides the predicted future trajectory of the leading object. If no leading object exists, it returns None.
- **get_future_trajectories_specific_objects**: Returns the future trajectories for specific objects identified by their object IDs. If no such objects exist, it returns None.
- **get_future_trajectories_in_range**: Retrieves trajectories where any waypoint falls within a given range (x_start, x_end)×(y_start, y_end)m². This function will return None if no trajectories meet the criteria.
- **get_future_waypoint_of_specific_objects_at_timestep**: Obtains the future waypoints of specific objects at a particular timestep. If an object does not have a waypoint at the specified timestep or if no such object exists, it returns None.
- **get_all_future_trajectories**: Provides the predicted future trajectories for all objects in the scene.

**Occupancy Functions**

- **get_occupancy_at_locations_for_timestep**: Determines the probability of occupancy for a list of locations at a specific timestep. Returns None if a location is outside the occupancy prediction scope.
- **check_occupancy_for_planned_trajectory**: Evaluates whether a planned trajectory collides with other objects.

**Map Functions**

- **get_drivable_at_locations**: Checks the drivability of specific locations. Returns None if a location is outside the map scope.
- **get_lane_category_at_locations**: Retrieves the lane category for specific locations. If the location is outside the map scope, it returns None.
- **get_distance_to_shoulder_at_locations**: Calculates the distance to both sides of road shoulders for specific locations. Returns None if a location is outside the map scope.
- **get_current_shoulder**: Provides the distance to both sides of road shoulders for the current ego-vehicle location.
- **get_distance_to_lane_divider_at_locations**: Computes the distance to both sides of road lane dividers for specific locations. Returns None if a location is outside the map scope.
- **get_current_lane_divider**: Returns the distance to both sides of road lane dividers for the current ego-vehicle location.
- **get_nearest_pedestrian_crossing**: Identifies the location of the nearest pedestrian crossing to the ego-vehicle. Returns None if no pedestrian crossing exists.

# B Implementation Details

In our experiments, we utilized the MS-SWIFT framework to finish our experiment with the following parameter configuration.

## B.1 SFT Phase

We trained the model for 20 epochs using the LoRA (Low-Rank Adaptation) fine-tuning method, setting the LoRA rank to 8 and the LoRA alpha to 32. To manage GPU memory usage, the per-device training and evaluation batch sizes were both set to 2, with gradient accumulation enabled over 16 steps. The learning rate was set to $1 \times 10^{-4}$, and we used the bfloat16 data type to enhance computational efficiency. In order to constrain GPU memory consumption, we froze the parameters of the Vision Transformer (ViT) and set the maximum sequence length to 4096. During training, model evaluation and saving were performed every 1000 steps, with a limit of 5 saved models. Logging was conducted every 5 steps to monitor the training process closely. We employed the DeepSpeed ZeRO-2 optimizer to optimize training performance and disabled the reentrancy of gradient checkpointing for efficiency. Additionally, we configured the warm-up ratio to 0.05 and allocated 4 worker processes for both the data loader and dataset processing.

## B.2 GRPO Phase

The RLHF type was designated as "grpo" to utilize the GRPO algorithm for reinforcement learning with human feedback. We activated LoRA training mode(same as SFT Phase in Sec. B.1) and set the PyTorch data type to "bfloat16" for efficient computation. The maximum sequence length was configured to 2048, with a maximum completion length of 1024. Training was conducted for 8 epoch, with per-device training and evaluation batch sizes both set to 1. The learning rate was set at $1 \times 10^{-5}$, and we employed 8 gradient accumulation steps. Model evaluation occurred every 500 steps, with model saving every 100 steps, and we restricted the total number of saved models to 20. Logging was performed at every step to closely monitor the training process. The warm-up ratio for the learning rate was configured to 0.01. We allocated 4 workers to the DataLoader to expedite data loading. For generation, we allowed 2 generations with a temperature of 1.2 to enhance output diversity. The system was prompted with the instruction: "You are a helpful assistant. You first think about the reasoning process in the mind and then provide the user with the answer." We leveraged DeepSpeed's ZeRO-3 stage optimizer to optimize training performance and enabled logging of model completions for further analysis. The beta value was set to 0.001 for specific algorithmic adjustments, and the entire process was conducted for 1 iteration.

# C Evaluation Metric in the DriveLMM-o1 Benchmark

To comprehensively evaluate both quantitative performance and qualitative decision-making capabilities in autonomous driving scenarios, we adopt the evaluation metrics proposed by (Ishaq et al., 2025).

**Risk Assessment** examines the model's capacity to prioritize high-risk objects or scenarios, ensuring critical situations receive appropriate urgency in decision-making processes. Accuracy quantifies the precision of environmental perception through correct identification and classification of relevant

elements.

**Traffic Rule Adherence** measures compliance with established traffic regulations and domain-specific best practices, reflecting real-world operational fidelity.

**Scene Awareness** and **Object Understanding** jointly assess contextual interpretation depth, encompassing accurate perception of spatial relationships, dynamic object behaviors, and complex environmental interactions.

**Relevance** evaluates alignment between model outputs and scenario-specific requirements relative to ground truth annotations, ensuring contextually appropriate responses.

**Missing Details** identifies critical information gaps through systematic analysis of perceptual blind spots in situational understanding. These complementary metrics collectively establish a holistic framework for evaluating system reliability, safety margins, and environmental adaptability across diverse driving conditions.

| Metric | Description |
|---|---|
| Risk Assessment Accuracy | Evaluates if the model correctly prioritizes high-risk objects or scenarios. |
| Traffic Rule Adherence | Scores how well the response follows traffic laws and driving best practices. |
| Scene Awareness and Object Understanding | Measures how well the response interprets objects, their positions, and actions. |
| Relevance | Measures how well the response is specific to the given scenario and ground truth. |
| Missing Details | Evaluates the extent to which critical information is missing from the response. |

Table 8: Evaluation metrics for quantitative performance and qualitative decision-making capabilities

DriveLMM-o1(Ishaq et al., 2025) utilize GPT-4o-mini to complete the evaluation steps. The prompt we employ is as follows:

*You are an autonomous driving reasoning evaluator. Your task is to assess the alignment, coherence, and quality of reasoning steps in text responses for safety-critical driving scenarios.*

You will evaluate the model-generated reasoning using the following metrics:

1. **Faithfulness-Step (1-10)**: Measures how well the model's reasoning steps align with the ground truth.
   - 9-10: All steps correctly match or closely reflect the reference.
   - 7-8: Most steps align, with minor deviations.
   - 5-6: Some steps align, but several are incorrect or missing.
   - 3-4: Few steps align; most are inaccurate or missing.
   - 1-2: Majority of steps are incorrect.
2. **Informativeness-Step (1-10)**: Measures completeness of reasoning.
   - 9-10: Captures almost all critical information.
   - 7-8: Covers most key points, with minor omissions.
   - 5-6: Missing significant details.
   - 3-4: Only partial reasoning present.
   - 1-2: Poor extraction of relevant reasoning.
3. **Risk Assessment Accuracy (1-10)**: Evaluates if the model correctly prioritizes high-risk objects or scenarios.
   - 9-10: Correctly identifies and prioritizes key dangers.
   - 7-8: Mostly accurate, with minor misprioritizations.
   - 5-6: Some important risks are overlooked.
   - 3-4: Significant misjudgments in risk prioritization.
   - 1-2: Misidentifies key risks or misses them entirely.
4. **Traffic Rule Adherence (1-10)**: Evaluates whether the response follows traffic laws and driving best practices.
   - 9-10: Fully compliant with legal and safe driving practices.
   - 7-8: Minor deviations, but mostly correct.
   - 5-6: Some inaccuracies in legal/safe driving recommendations.
   - 3-4: Several rule violations or unsafe suggestions.
   - 1-2: Promotes highly unsafe driving behavior.

5. **Scene Awareness & Object Understanding (1-10)**: Measures how well the response interprets objects, their positions, and actions.
    - 9-10: Clearly understands all relevant objects and their relationships.
    - 7-8: Minor misinterpretations but mostly correct.
    - 5-6: Some key objects misunderstood or ignored.
    - 3-4: Many errors in object recognition and reasoning.
    - 1-2: Misidentifies or ignores key objects.
6. **Repetition-Token (1-10)**: Identifies unnecessary repetition in reasoning.
    - 9-10: No redundancy, very concise.
    - 7-8: Minor repetition but still clear.
    - 5-6: Noticeable redundancy.
    - 3-4: Frequent repetition that disrupts reasoning.
    - 1-2: Excessive redundancy, making reasoning unclear.
7. **Hallucination (1-10)**: Detects irrelevant or invented reasoning steps not aligned with ground truth.
    - 9-10: No hallucinations, all reasoning is grounded.
    - 7-8: One or two minor hallucinations.
    - 5-6: Some fabricated details.
    - 3-4: Frequent hallucinations.
    - 1-2: Majority of reasoning is hallucinated.
8. **Semantic Coverage-Step (1-10)**: Checks if the response fully covers the critical reasoning elements.
    - 9-10: Nearly complete semantic coverage.
    - 7-8: Good coverage, some minor omissions.
    - 5-6: Partial coverage with key gaps.
    - 3-4: Major gaps in reasoning.
    - 1-2: Very poor semantic coverage.
9. **Commonsense Reasoning (1-10)**: Assesses the use of intuitive driving logic in reasoning.
    - 9-10: Displays strong commonsense understanding.
    - 7-8: Mostly correct, with minor gaps.
    - 5-6: Some commonsense errors.
    - 3-4: Frequent commonsense mistakes.
    - 1-2: Lacks basic driving commonsense.
10. **Missing Step (1-10)**: Evaluates if any necessary reasoning steps are missing.
    - 9-10: No critical steps missing.
    - 7-8: Minor missing steps, but answer is mostly intact.
    - 5-6: Some important steps missing.
    - 3-4: Many critical reasoning gaps.
    - 1-2: Response is highly incomplete.
11. **Relevance (1-10)**: Measures how well the response is specific to the given scenario and ground truth.
    - 9-10: Highly specific and directly relevant to the driving scenario. Captures critical elements precisely, with no unnecessary generalization.
    - 7-8: Mostly relevant, but some minor parts may be overly generic or slightly off-focus.
    - 5-6: Somewhat relevant but lacks precision; response contains vague or general reasoning without clear scenario-based details.
    - 3-4: Mostly generic or off-topic reasoning, with significant irrelevant content.
    - 1-2: Largely irrelevant, missing key aspects of the scenario and failing to align with the ground truth.
12. **Missing Details (1-10)**: Evaluates the extent to which critical information is missing from the response, impacting the reasoning quality.
    - 9-10: No significant details are missing; response is comprehensive and complete.
    - 7-8: Covers most important details, with minor omissions that do not severely impact reasoning.
    - 5-6: Some essential details are missing, affecting the completeness of reasoning.
    - 3-4: Many critical reasoning steps or contextual details are absent, making the response incomplete.
    - 1-2: Response is highly lacking in necessary details, leaving major gaps in understanding.

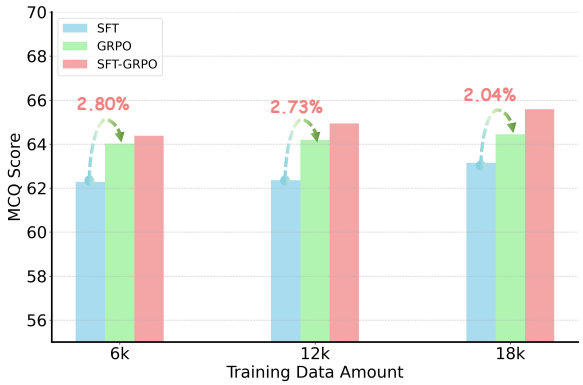


Figure 6: Our GRPO method with structured rewards setting and two-stage training strategy significantly boost final answer accuracy. With just 6k samples, it outperforms SFT, showing strong performance even with limited data. As data increase, AlphaDrive consistently leads in MCQ performance.

### Final Evaluation

Compute the Overall Score as the average of all metric scores.

```
{
  "Faithfulness-Step": 6.0,
  "Informativeness-Step": 6.5,
  "Risk Assessment Accuracy": 7.0,
  "Traffic Rule Adherence": 7.5,
  "Object Understanding": 8.0,
  "Repetition-Token": 7.0,
  "Hallucination": 8.5,
  "Semantic Coverage-Step": 7.5,
  "Commonsense Reasoning": 7.0,
  "Missing Step": 8.5,
  "Relevance": 8.5,
  "Missing Details": 7.0,
  "Overall Score": 7.42
}
```

### Guidelines

- Avoid subjective interpretation and adhere to the given thresholds.
- Always strictly follow these scoring guidelines.
- Do not add any additional explanations beyond the structured JSON output.

## D Visualization of Main Experiment

Here we provide the visualizations for different scenarios (Fig 7, 8). Correct scene understanding outputs are highlighted in green, while erroneous interpretations are marked in red for comparative analysis. Notably, AgentThink demonstrates accurate analytical capabilities and avoid overly conservative decision-making tendencies (e.g. unnecessary braking), which compromise driving comfort despite maintaining safety margins.

## E Impact of Data Size

To systematically evaluate the influence of training data scale, we conduct ablation studies on Qwen2.5-VL-7B with fixed vision encoder parameters. The base model is first finetuned via LoRA-based SFT, followed by policy optimization through GRPO algorithm. All experiments are implemented on 16× NVIDIA A800 GPUs with unified hyperparameter configurations. As presented in Tab. 9 and Fig. 6, reducing training data size from 12k to 6k samples leads to a performance decline across all metrics, with SFT-based training exhibiting a more pronounced degradation. Notably, RLFT demonstrates superior robustness to data scarcity, achieving statistically significant performance gains over SFT counterparts in low-data regimes ($52.32\%$ vs. $62.28\%$ final accuracy at 6k data scale). This suggests the inherent advantage of reinforcement learning frameworks in leveraging limited training samples for vision-language policy learning.

## F Evaluation Metric in the DriveMLLM Benchmark

The eight metrics in the DriveMLLM benchmark (Guo et al., 2024) can be described as follows:

**Left/Right (L/R)** This metric evaluates the model's ability to identify which of two objects is positioned further to the left or right in the image based on their inferred $x$-coordinates.

**Front/Back (F/B)** This assesses whether the model can determine if one object is in front of or behind another using depth cues, reflecting its understanding of $z$-coordinate relationships.

$$acc_i = \begin{cases} 1, & \text{if } p_i = y_i \\ 0, & \text{if } p_i \neq y_i \end{cases} \tag{8}$$

where: - $p_i$ is the model's predicted label for sample $i$. - $y_i$ is the ground truth label for sample $i$.

The overall accuracy $acc$ is calculated as the mean of the individual accuracies.

| ID | Amount of Data | Driving Metrics (%) ↑ | | | Scene Detail (%) ↑ | | Overall (%) ↑ | |
|---|---|---|---|---|---|---|---|---|
| | | Risk Ass. | Rule Adh. | Obj. Und. | Relevance | Missing | Reason. | MCQ |
| (a) | GRPO w/ 6k data | 68.79 | 74.93 | 71.12 | 75.44 | 67.80 | 69.09 | 64.02 |
| (b) | GRPO w/ 12k data | **69.26** | **75.44** | **71.68** | **75.85** | **68.13** | **69.48** | **64.19** |
| (c) | SFT w/ 6k data | 74.47 | 80.58 | 75.82 | 80.66 | 72.02 | 74.08 | 62.28 |
| (d) | SFT w/ 12k data | **74.81** | **80.88** | **76.21** | **80.90** | **72.42** | **74.33** | **62.36** |
| (e) | SFT-GRPO w/ 6kdata | 75.60 | 81.40 | 76.97 | 81.74 | 72.51 | 74.92 | 64.38 |
| (f) | SFT-GRPO w/ 12kdata | **75.69** | **82.29** | **76.98** | **82.58** | **72.39** | **75.08** | **64.94** |

Table 9: Impact of training data amount on key evaluation metrics (reordered by driving metrics first).

**Relative Height Difference (RHD)** It tests the model's capability to calculate the vertical distance between the camera and an object based on the object's $z$-coordinate.

**Relative Distance (RD)** This metric examines the model's ability to estimate the Euclidean distance from the camera to a specified object using inferred spatial information.

$$acc_i = \frac{1}{1 + \alpha_d \|d_i - d_i^{\text{gt}}\|_1} \tag{9}$$

where: - $d_i$ is the model's predicted distance for sample $i$. - $d_i^{\text{gt}}$ is the ground truth distance for sample $i$. - $\alpha_d$ is a scaling factor controlling the penalty for deviation, set to $\alpha_d = 0.05$.

**Positional Precision (PPos)** It evaluates the model's accuracy in identifying the exact coordinates $[x, y]$ of a specified object within an image.

$$acc_i = \frac{1}{1 + \alpha_p \|\mathbf{x}_i - \mathbf{x}_i^{\text{gt}}\|_2} \tag{10}$$

where: - $\mathbf{x}_i = (x_i, y_i)$ are the model's predicted coordinates for sample $i$. - $\mathbf{x}_i^{\text{gt}} = (x_i^{\text{gt}}, y_i^{\text{gt}})$ are the ground truth coordinates for sample $i$. - $\alpha_p$ is a scaling factor, set to $\alpha_p = 0.005$.

**Bounding Box (BBox)** This metric assesses the model's capability to determine the bounding box coordinates $[min\_x, min\_y, max\_x, max\_y]$ of a specific object.

$$acc_i = \text{IoU}(B_i, B_i^{\text{gt}}) = \frac{|B_i \cap B_i^{\text{gt}}|}{|B_i \cup B_i^{\text{gt}}|} \tag{11}$$

where: - $B_i$ is the model's predicted bounding box for sample $i$. - $B_i^{\text{gt}}$ is the ground truth bounding box for sample $i$. - $|\cdot|$ denotes the area of the bounding box.

**Camera-Object Vertical Distance (CVD)** It tests the model's ability to calculate the vertical distance from the camera to an object based on the object's $z$-coordinate.

**Camera-Object Euclidean Distance (CD)** This metric evaluates the model's skill in computing the Euclidean distance between the camera and an object using inferred spatial information. The computation of the CVD and CD metrics follows the same methodology as that of the RVD and RD metrics.

**Overall Accuracy Score (AccS)** This provides a comprehensive measure of the model's performance across all the aforementioned tasks, reflecting its overall spatial understanding capabilities.

$$acc = \frac{1}{N}\sum_{i=1}^{N} acc_i \quad \text{AccS} = \frac{1}{8}\sum_{j=1}^{8} acc_j \tag{12}$$

**Question:**
<image>
What is the best way to avoid a collision? Choose from the following answers only
A) Come to a complete stop.
B) Brake suddenly.
C) Speed up.
D) Change lane to right.
E) Change lane to left.
F) None of the options.

**Ground-Truth:**
**Identify hazards:** The front right image shows a white car moving near the curb of the road, indicating potential for collision if another vehicle is approaching.
**Analyze options:** Given the visible surroundings, change in speed (options A, B, C) or sudden braking (option B) might not be ideal. Changing lanes (option D or E) could bypass the obstacle.
**Final Answer: D) Change lane to right.**

**Baseline**

1.Identify the scenario: The image shows a street scene with a car in motion, trees, buildings, and road signs. **There is no visible indication of an immediate threat or collision.**
2.Analyze the options:
A) Come to a complete stop: This could be risky if there's a need to continue driving.
B) Brake suddenly: This can be dangerous and may cause a loss of control.
C) Speed up: This increases the risk of a collision.
**D) Change lane to right: This might not be safe if there's traffic coming from the right.**
E) Change lane to left: This could be safer if there's clear space on the left side.
F) None of the options: This would mean no action is needed, which isn't always the case.

Consider the safest option: **Given the lack of immediate danger and the need for caution, changing lanes to the left (option E) seems like the safest choice if there's clear space available.**

**Final Answer: E) Change lane to left.** ✗

**SFT+RLFT**

1.In the 'front' image, **there are pedestrians visible on the sidewalk near the road**. This indicates that there may be potential obstacles or the need to yield, which requires immediate attention to avoid a collision.

2.The 'front right' image shows **the road ahead with pedestrians crossing**. The presence of pedestrians suggests that the ego vehicle should slow down or stop to ensure their safety.

3.Considering the current speed of the ego vehicle is 9.08 km/h **and the presence of pedestrians**, the safest option to avoid a collision is to come to a complete stop.

**Final Answer: A) Come to a complete stop.**

**AgentThink**

1.Identify Hazards:
<Tool name>: get_surrounding_object, get_drivable_lane
<Reason>:**The front right image shows a white car moving near the curb of the road, which indicates a potential hazard**. This suggests that there is a risk of collision if another vehicle approaches from behind.
2. Analyze Options:
**Option A (Come to a complete stop): This could be risky because it might not allow enough time for the other vehicle to react, especially if it's approaching at high speed.**
Option B (Brake suddenly): Sudden braking can be dangerous and might cause the vehicle to skid, increasing the risk of a collision.
Option C (Speed up): Speeding up would increase the likelihood of a collision with the white car.
**Option D (Change lane to right): Changing to the right lane could help bypass the obstacle**
**Option E (Change lane to left): Changing to the left lane might not be possible if the left lane is occupied or if it's not safe to change lanes.**
Option F (None of the options): This option is not applicable since we have identified viable options.

**Final Answer**: D) Change lane to right. ✓

Figure 7: Visulization of Results. Correct scene understanding outputs are highlighted in green, while erroneous interpretations are marked in red for comparative analysis. Notably, AgentThink demonstrates accurate analytical capabilities and avoid overly conservative decision-making tendencies (e.g. unnecessary braking), which compromise driving comfort despite maintaining safety margins.

**Question:**
What is the best way to avoid a collision? Choose from the following answers only
A) Come to a complete stop.
B) Brake suddenly.
C) Speed up.
D) Change lane to right.
E) Change lane to left.
F) None of the options.

**Ground-Truth:**
1. The motorcycle is directly ahead, and altering speed suddenly could be unsafe.

2. Coming to a complete stop could prevent a collision with the motorcycle.

**Final Answer: A) Come to a complete stop.**

## Baseline

1.Identify the scenario: The image shows a nighttime road scene with vehicles in motion. There are **no visible signs of an immediate threat or collision**.
2.Analyze the options:
A) Come to a complete stop: This could be dangerous if there's traffic behind you, as it might cause a chain reaction.
B) Brake suddenly: This can be risky if you're not sure what's ahead, as it might cause a rear-end collision.
C) Speed up: This is generally unsafe and increases the risk of accidents.
D) Change lane to right: **If there's clear space and no oncoming traffic, this could be a safe option**.
E) Change lane to left: If there's clear space and no oncoming traffic, this could also be a safe option.
F) None of the options: This would mean none of the given options are suitable.

Consider the context: The image does not show any immediate danger, but the question asks for the best way to avoid a collision. Given the options, **changing lanes to either side (D or E) seems like the safest choice** if there's clear space and no oncoming traffic.
**Final Answer: D) Change lane to right.** ✗

## SFT+RLFT

1.Front-view images show **vehicles moving toward the ego vehicle**, indicating potential collision risks if it continues without intervention.

2.The **wet road surface** increases stopping distances and reduces traction.

3.**Nighttime** conditions lower visibility, and road signs suggest caution, implying a complete stop may be safest.

4.Given the oncoming traffic and hazards, **stopping is the best option to prevent collisions**, as sudden braking might cause skidding and lane changes could risk other dangers.

**Final Answer: A) Come to a complete stop.** ✓

## AgentThink

1.<Tool name>: get_surrounding_detection
There is a **motorcycle directly ahead**, which poses a potential collision risk.

2.Evaluate the Options:
A) Come to a complete stop: This option **is safe and can prevent a collision if the motorcycle is too close or if there is not enough time to react safely**.
B) Brake suddenly: This could be unsafe because it might cause the vehicle to skid, especially at night when visibility is low.
C) Speed up: This would increase the risk of a collision and is not a safe option.
D) Change lane to right: This could be risky if the motorcycle is in the same lane and changing lanes could lead to a collision.
E) Change lane to left: This could also be risky for the same reasons as changing to the right.
F) None of the options: This is not applicable since we have identified a safe option.

Conclusion: The safest option is to come to a complete stop to ensure that the motorcycle has enough space to pass or to allow the driver to react safely without the risk of sudden braking causing a skid.

**Final Answer: A) Come to a complete stop.** ✓

Figure 8: Visulization of AgentThink. Correct scene understanding outputs are highlighted in green, while erroneous interpretations are marked in red for comparative analysis.