

DriveVLM-RL: Neuroscience-Inspired Reinforcement Learning with Vision-Language Models for Safe and Deployable Autonomous Driving

Zilin Huang^a, Zihao Sheng^a, Zhengyang Wan^a, Yansong Qu^b, Junwei You^a, Sicong Jiang^c, Sikai Chen^{a,*}

^aDepartment of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, WI, 53706, USA

^bLyles School of Civil and Construction Engineering, Purdue University, West Lafayette, IN 47907, USA

^cDepartment of Civil Engineering, McGill University, Montreal, QC, H3A 0C3, Canada

Abstract

Ensuring safe decision-making in autonomous vehicles remains a fundamental challenge despite rapid advances in end-to-end learning approaches. Traditional reinforcement learning (RL) methods rely on manually engineered rewards or sparse collision signals, which fail to capture the rich contextual understanding required for safe driving and make unsafe exploration unavoidable in real-world settings. Recent vision-language models (VLMs) offer promising semantic understanding capabilities; however, their high inference latency and susceptibility to hallucination hinder direct application to real-time vehicle control. To address these limitations, this paper proposes **DriveVLM-RL**, a neuroscience-inspired framework that integrates VLMs into RL through a dual-pathway architecture for safe and deployable autonomous driving. Inspired by the human brain’s habitual and deliberative visual processing, DriveVLM-RL decomposes semantic reward learning into a **Static Pathway** for continuous spatial safety assessment via CLIP-based contrasting language goals, and a **Dynamic Pathway** for attention-gated multi-frame semantic risk reasoning via a lightweight detection model and large VLM (LVLM). A hierarchical reward synthesis mechanism fuses these signals with vehicle state information, while an asynchronous training pipeline decouples expensive LVLM inference from environment interaction. Critically, all VLM components operate exclusively during offline training and are completely removed at deployment, eliminating inference latency at test time. Extensive experiments in the CARLA simulator demonstrate that DriveVLM-RL significantly outperforms state-of-the-art baselines in collision avoidance, task success, and generalization across diverse traffic scenarios. Notably, even under extreme “no-reward-after-collision” settings where explicit collision penalties are removed, DriveVLM-RL maintains low collision rates through semantic risk reasoning alone, accumulating 67% fewer collisions than penalty-dependent baselines, and further demonstrates robustness across unseen towns and algorithm-agnostic transferability to on-policy learning. These results demonstrate that DriveVLM-RL provides a practical paradigm for integrating foundation models into autonomous driving without compromising real-time feasibility. The demo video and code are available at: <https://zilin-huang.github.io/DriveVLM-RL-website/>.

Keywords: Autonomous Driving, Vision-Language Models, Reinforcement Learning, Reward Design, Real-World Deployment

1. Introduction

The deployment of autonomous vehicles (AVs) in real-world traffic environments has accelerated rapidly in recent years, transitioning from controlled testing scenarios to large-scale urban operations. In late 2025, Tesla released version 14 of its Full Self-Driving (FSD) system, representing a significant advancement in end-to-end neural network-based driving pipelines (Tesla, 2025). Currently, Waymo has expanded its robotaxi services in multiple U.S. cities (Kolodny, 2025), while in China, Baidu’s Apollo Go scaled its operations to 22 cities and launched international deployments in Dubai and Abu Dhabi (CarNewsChina, 2025). However, despite these advances, ensuring safe and reliable decision-making in open-world environments remains a fundamental challenge that directly impacts public trust and regulatory acceptance (Feng et al., 2023; Jiao et al., 2025; Luo et al., 2025). In complex traffic scenarios that involve diverse road users, ambiguous intent, and long-tail events, autonomous driving systems must consider semantic risks beyond purely geometric perception (Han et al., 2025).

Two dominant paradigms have emerged for end-to-end learning-based AV decision-making: imitation learning (IL) and reinforcement learning (RL), as shown in Fig. 1 (a) (Huang et al., 2024). IL methods learn driving policies by mimicking expert demonstrations, offering simplicity and benefiting from abundant naturalistic driving data. Yet,

*Corresponding author: Sikai Chen.

Email addresses: zilin.huang@wisc.edu (Zilin Huang), zihao.sheng@wisc.edu (Zihao Sheng), zhengyang.wan@wisc.edu (Zhengyang Wan), qu120@purdue.edu (Yansong Qu), jyou38@wisc.edu (Junwei You), sicong.jiang@mail.mcgill.ca (Sicong Jiang), sikai.chen@wisc.edu (Sikai Chen)

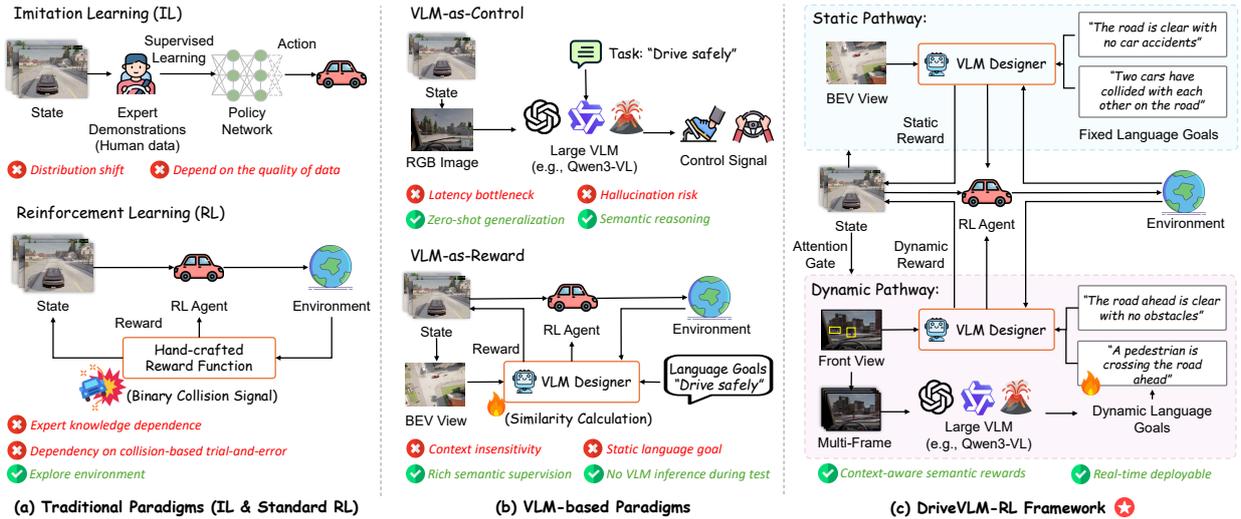


Figure 1: Comparative learning paradigms for autonomous driving. (a) Traditional policy learning approaches, including IL and RL, which rely on expert demonstrations or hand-crafted rewards. (b) Foundation model-based approaches, including VLM-as-Control and VLM-as-Reward paradigms. (c) The proposed DriveVLM-RL framework, which integrates a dual-pathway architecture to enable dynamic, context-aware semantic rewards while remaining real-time deployable.

IL suffers from well-known limitations (Ross et al., 2011; De Haan et al., 2019): (1) distribution shift, where the learned policy encounters states not represented in the training data and fails to recover; (2) bounded performance, meaning the policy cannot surpass the capability of the demonstrator; and (3) causal confusion, in which spurious correlations in demonstrations lead to brittle or unsafe behaviors. These limitations are especially problematic for safety-critical scenarios that rarely appear in naturalistic data but are essential for robust autonomous driving. In contrast, RL offers a principled alternative by enabling agents to learn through trial-and-error interaction with the environment, potentially discovering novel strategies that surpass human performance (Sutton et al., 1998). RL has achieved superhuman capabilities in domains ranging from board games to robotic manipulation (Mnih et al., 2015). For autonomous driving, RL holds the promise of learning adaptive policies that can handle rare but critical scenarios through closed-loop exploration (Huang et al., 2025a; He et al., 2024; Aradi, 2020). However, RL’s effectiveness critically depends on the design of reward functions that accurately encode desired driving behaviors.

Reward function design plays a pivotal role in RL, as it directly shapes the agent’s behavior and determines the quality of the learned driving policy (Sutton et al., 1998; Lu et al., 2025). In autonomous driving, reward functions are generally hand-crafted based on expert intuition, typically combining sub-objectives such as speed maintenance, lane tracking, and collision avoidance. Prior studies (Knox et al., 2023; Abouelazm et al., 2024) highlight several inherent limitations, including dependence on expert knowledge, conflicting objectives, and poor generalization. More critically, traditional reward functions rely on binary collision signals to learn safety, meaning the agent must physically experience crashes to recognize dangerous situations. This creates a fundamental barrier to real-world deployment: *allowing an AV to learn from actual collisions is unacceptable* (Huang et al., 2025a; Wu et al., 2024; Garcia and Fernández, 2015). Moreover, safe driving involves nuanced judgments that are difficult to encode in hand-crafted reward functions (Han et al., 2024; Zhou et al., 2024b). For instance, a pedestrian standing on the sidewalk versus stepping onto the road requires fundamentally different responses, yet both scenarios may appear similar in terms of distance-based metrics. Human drivers rely on rich semantic understanding, including intent, context, and social norms, which cannot be represented by simple geometric or physics-based reward terms.

The emergence of foundation models, such as large language models (LLMs) and vision-language models (VLMs), provide a promising alternative for addressing this limitation (Cui et al., 2024; Jiang et al., 2025; HAZRA et al., 2025; Sheng et al., 2026; Pang et al., 2026). By jointly reasoning over visual observations and natural language, VLMs can infer high-level semantic concepts such as risk, intent, and social context, which are difficult to encode through conventional reward engineering. Several recent studies have explored “VLM-based control” paradigms (Fig. 1 (b), top), where VLMs directly map visual inputs to driving actions or generate real-time control commands (Tian et al., 2025; Qian et al., 2025; Zhou et al., 2025; You et al., 2026). However, these paradigms suffer from two fundamental issues: (1) the computational latency of 500–2000 ms per inference far exceeds the 20–100 ms control cycles required for safe vehicle operation (Cui et al., 2024; Zhou et al., 2024a). (2) VLMs are susceptible to hallucination, producing outputs that may be inconsistent with the visual input, which could lead to catastrophic failures when used directly for vehicle control (Xie et al., 2025; Meng et al., 2025). Recently, researchers have proposed an alternative paradigm: rather than using VLMs for direct control, VLMs are integrated with RL to provide semantic understanding that shapes reward signals and guides policy learning (Fig. 1 (b), bottom). This allows policies to leverage VLM semantics while avoiding the latency and reliability constraints of real-time control.

This “VLM-as-Reward” paradigm has shown promising results in robotic domain (Lee et al., 2026), such as

VLM-SR (Baumli et al., 2023), RoboCLIP (Sontakke et al., 2023), DriveMind (Wasif et al., 2025), and VLM-RM (Rocamonde et al., 2024), which leverage contrastive language-image pre-training (CLIP)’s semantic embeddings to measure goal achievement from visual observations. However, unlike robotic domain where goals can be precisely specified, driving objectives expressed in natural language (e.g., “drive safely”) are inherently ambiguous and difficult to translate into dense, informative reward signals. To address this ambiguity, LORD (Ye et al., 2025) proposed using negative language goals to describe dangerous states. Our previous work (Huang et al., 2025b) further proposed contrasting language goals (CLGs) that leverage both positive and negative descriptions. Despite these advances, several challenges remain unresolved: (1) Most existing methods use CLIP to compute similarity between observations and fixed language goals. Such static formulations lack contextual awareness and cannot capture the dynamic, evolving, and temporally dependent nature of traffic risk. (2) A natural solution is to adopt large VLM (LVLM), such as GPT (Achiam et al., 2023) or Qwen-VL (Yang et al., 2025), to perform multi-frame semantic reasoning. However, invoking LVLM for every frame during RL training is computationally prohibitive, creating severe scalability bottlenecks when millions of environment interactions are required.

To address these challenges, we draw inspiration from the human brain’s visual processing architecture, which has evolved to efficiently balance routine perception and context-dependent reasoning. As shown in Fig. 2, during routine driving, the brain operates in an efficient habitual mode via sensorimotor loops mediated by the parietal cortex (Goodale and Milner, 1992). When safety-critical situations occur, such as the sudden appearance of a pedestrian, the brain’s selective attention network rapidly engages to redirect cognitive focus (Corbetta and Shulman, 2002), triggering higher-order semantic reasoning in the prefrontal cortex (Miller and Cohen, 2001), such as “A pedestrian is stepping into the roadway; I must slow down and prepare to stop”. The critical insight is that the brain does not perform deep, energy-intensive analysis on every visual frame; instead, it employs an attention gate (Desimone et al., 1995) to determine when to invoke slower but more powerful contextual reasoning processes. Inspired by the brain’s dual-pathway cognitive architecture, we propose **DriveVLM-RL**, a neuroscience-inspired cognitive framework that integrates VLMs into RL for safe and deployable autonomous driving. As illustrated in Fig. 1 (c), DriveVLM-RL fundamentally treats VLMs as semantic teachers rather than real-time decision-makers, alleviating the reliance of traditional RL on collision-based learning while avoiding the latency and hallucination issues of “VLM-for-Control”.

The main contributions of this work are summarized as follows:

- We propose DriveVLM-RL, the first framework to explicitly integrate the human brain’s dual-pathway cognitive architecture into the VLM-as-Reward paradigm for autonomous driving. The **Static Pathway** (using CLIP and fixed language goals) simulates the brain’s dorsal stream for continuous spatial awareness, while the **Dynamic Pathway** (using attention-gated LVLM reasoning) simulates the brain’s selective attention-prefrontal cortex circuit for higher-order semantic risk reasoning. This hierarchical design enables the RL agent to learn complex safety maneuvers through semantic understanding, alleviating the fundamental limitations of traditional collision-based reward functions.
- We design a novel **attention gating mechanism** that simulates the brain’s selective attention to address the computational complexity and high inference latency inherent in VLM-as-Reward paradigms. A lightweight perception model first analyzes foreground scenes, effectively filtering routine driving frames and triggering computationally expensive LVLM inference only when safety-critical objects are detected. Critically, the VLM is invoked exclusively during the RL training phase; once training is complete, no VLM calls are required during deployment, eliminating the infeasible latency that plagues VLM-for-Control methods.
- We introduce a hierarchical reward synthesis mechanism that fuses static visual-language similarity, dynamic multi-frame semantic reasoning, and vehicle state information into dense and proactive reward signals. Furthermore, we design an asynchronous training pipeline that decouples expensive VLM inference from environment interaction, enabling scalable learning despite the high computational cost of large models. Critically, all VLM components operate exclusively during offline training and are completely removed during deployment, allowing the final driving policy to execute with low latency.
- Extensive experiments conducted in CARLA simulator (Dosovitskiy et al., 2017) demonstrate that DriveVLM-RL significantly improves driving safety, robustness, and generalization across diverse traffic scenarios. Remarkably, even under extreme “no-reward-after-collision” settings where explicit collision penalties are removed, agents trained with DriveVLM-RL still learn to avoid collisions through semantic risk reasoning alone. These results indicate that DriveVLM-RL provides a practical and generalizable paradigm for leveraging foundation models to train autonomous driving policies that are both safe and deployable in real-world systems.

The remainder of this paper is organized as follows. Section 2 presents the preliminaries on RL formulation and the VLM-as-Reward paradigm. Section 3 details the proposed DriveVLM-RL framework. Section 4 presents the experimental setup and results. Section 5 concludes the paper and outlines future research directions.

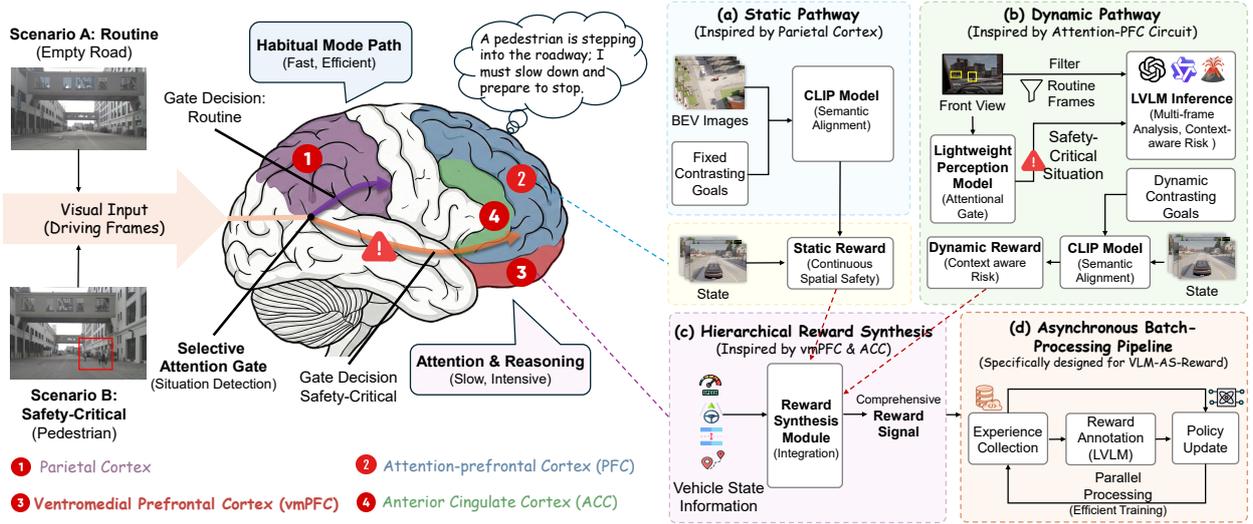


Figure 2: Neuroscience-inspired motivation of DriveVLM-RL. The framework is inspired by the brain’s habitual and deliberative visual processing: routine scenes are handled by a fast pathway, while safety-critical situations trigger attention and higher-level semantic reasoning, motivating a dual-pathway reward learning design.

2. Preliminaries

2.1. Markov Decision Process Formulation

We model the autonomous driving decision-making task as a Partially Observable Markov Decision Process (POMDP) (Kaelbling et al., 1998), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O}, R, \phi, \gamma, d_0)$. Here, \mathcal{S} denotes the state space, \mathcal{A} is the action space, $\mathcal{T}(s' | s, a)$ is the state transition function, $R(s, a)$ is the reward function, \mathcal{O} is the observation space, $\phi(o | s)$ is the observation emission function, $\gamma \in [0, 1)$ is the discount factor, and $d_0(s)$ is the initial state distribution. At each timestep t , the agent receives an observation $o_t \in \mathcal{O}$ from the environment and selects an action $a_t \in \mathcal{A}$ according to its policy $\pi(a_t | o_t)$. In our framework, the observation o_t comprises bird’s-eye-view (BEV) representations and front-view camera images, while the action $a_t = (a_t^{\text{throttle}}, a_t^{\text{steer}})$ consists of continuous throttle and steering commands. The environment transitions to a new state $s_{t+1} \sim \mathcal{T}(\cdot | s_t, a_t)$, and the agent receives a scalar reward $r_t = R(o_t, a_t)$, where we use observation o_t in place of the latent state s_t due to partial observability. The learning objective is to find an optimal policy π^* that maximizes the expected discounted return: $\pi^* = \arg \max_{\pi} G(\pi) = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t r_t \right]$.

2.2. VLM-as-Reward Paradigm

VLMs are models capable of jointly processing language inputs $l \in \mathcal{L}^{\leq n}$ and visual inputs $i \in \mathcal{I}^{\leq m}$, where \mathcal{L} denotes a finite vocabulary and \mathcal{I} the space of RGB images. A prominent class of VLMs is based on CLIP (Radford et al., 2021). CLIP consists of a language encoder $f_L : \mathcal{L}^{\leq n} \rightarrow \mathcal{E}$ and an image encoder $f_I : \mathcal{I} \rightarrow \mathcal{E}$, both mapping inputs into a shared embedding space $\mathcal{E} \subseteq \mathbb{R}^d$. These encoders are jointly trained via contrastive learning on large-scale image-caption pairs, minimizing the cosine distance for semantically aligned pairs while maximizing it for mismatched pairs. The alignment capability of CLIP enables the VLM-as-Reward paradigm, where semantic similarity between visual observations and language goals serves as a reward signal for RL training. Given an image encoder f_I , a language encoder f_L , a visual observation o_t , and a language goal l , the VLM-based reward is usually defined as $r_t^{\text{VLM}} = \text{sim}(f_I(o_t), f_L(l))$, where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity. This formulation provides the technical foundation for the VLM-as-Reward paradigm.

2.3. Problem Statement

A key challenge is designing an effective reward function $R(o_t, a_t)$ that guides the agent toward safe and efficient behaviors. Traditional reward engineering requires manual specification and extensive tuning of multiple sub-objectives, which is labor-intensive, error-prone, and difficult to generalize across diverse driving scenarios. The VLM-as-Reward paradigm offers a promising alternative by leveraging VLMs to provide semantically grounded reward signals. Ideally, we seek a reward function of the form:

$$R_{\text{VLM}}(o_t) = \Phi(l, o_t, c; \theta_{\text{VLM}}) \quad (1)$$

where l is a linguistic goal specification, o_t is the current observation, $c \in \mathcal{C}$ is optional contextual information (e.g., multi-frame history or scene description), and θ_{VLM} denotes the frozen VLM parameters. When $c = \emptyset$, the formulation reduces to the standard CLIP-based reward $r_t^{\text{VLM}} = \text{sim}(f_I(o_t), f_L(l))$.

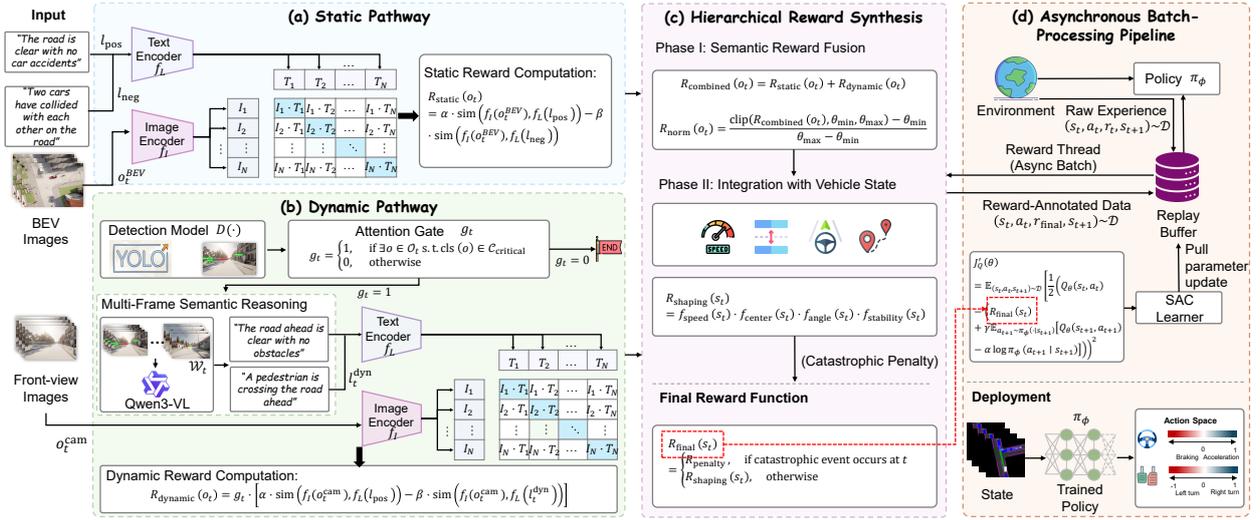


Figure 3: Overview of DriveVLM-RL. (a) Static Pathway: CLIP-based semantic alignment with contrasting language goals to provide continuous spatial safety assessment. (b) Dynamic Pathway: an attention-gated mechanism triggers multi-frame LVL reasoning only in safety-critical situations. (c) Hierarchical reward synthesis: static and dynamic semantic signals are fused and integrated with vehicle-state factors to produce the final shaping reward. (d) Asynchronous training pipeline: reward computation is decoupled from environment interaction and policy learning.

Most existing methods employ CLIP with fixed language goals (negative or positive) to compute reward signals based on text–image similarity. However, traffic scenes involve rich temporal and contextual variations, and this setting cannot capture factors such as pedestrian trajectory, motion intent, or evolving environmental conditions. A natural extension is to use LVL to generate dynamic language goals conditioned on the current scene. Given an image o_t and a text prompt p , an LVL generates a response $y = \text{LVL}(o_t, p)$. While LVL can analyze driving scenarios and provide nuanced semantic assessments, they incur substantially higher inference latency compared to CLIP. This latency far exceeds the requirements for real-time RL training, where millions of environment steps must be evaluated, making per-frame LVL inference computationally infeasible. This motivates the attention-gated dual-pathway design of DriveVLM-RL, detailed in Section 3.

3. Framework: DriveVLM-RL

3.1. Overview

The DriveVLM-RL framework is a neuro-inspired cognitive architecture designed to address the reward design challenge in RL-based autonomous driving while overcoming the semantic and contextual limitations of existing VLM-as-Reward methods. As illustrated in Fig. 2, the framework comprises four main components: **(1) Static Pathway.** Simulating the brain’s dorsal visual stream (parietal cortex), this pathway utilizes a pre-trained CLIP model to compute semantic alignment between BEV images and fixed CLGs, providing continuous spatial safety assessment. **(2) Dynamic Pathway.** Simulating the brain’s attention-prefrontal cortex (PFC) circuit, this pathway employs a lightweight perception model as an attentional gate to filter routine frames, triggering computationally expensive LVL inference only when safety-critical situations are detected. The LVL analyzes multi-frame sequences to generate dynamic, context-aware risk descriptions. **(3) Hierarchical Reward Synthesis.** Simulating the ventromedial prefrontal cortex (vmPFC) and anterior cingulate cortex (ACC), this module integrates static and dynamic rewards with vehicle state information to produce comprehensive reward signals. **(4) Asynchronous Batch-Processing Pipeline.** To enable efficient training despite LVL latency, this pipeline decouples reward computation from environment interaction, allowing parallel processing of experience collection, reward annotation, and policy updates. We describe each component in detail in the following subsections, as shown in Fig. 3.

3.2. Static Pathway

The Static Pathway simulates the parietal cortex (dorsal visual stream), which mediates habitual sensorimotor processing for spatial tasks such as lane keeping and distance maintenance. This pathway generates continuous reward signals based on foundational spatial safety assessment.

3.2.1. Static Reward Computation

The input for this pathway is the agent’s BEV image o_t^{BEV} , as this top-down representation provides unambiguous spatial relationships without the occlusion inherent in first-person views—analogueous to the parietal cortex’s integrated spatial map. However, a single abstract goal (e.g., “drive safely”) is semantically ambiguous and provides weak reward signals (Ye et al., 2025). We therefore introduce a CLG formulation (Huang et al., 2025b) that compares desired and undesired outcomes to produce more discriminative rewards.

Definition 1 (Static Contrasting Language Goal). Given the driving task, the Static CLG is a fixed pair $(l_{pos}, l_{neg}) \in \mathcal{L}^{\leq n} \times \mathcal{L}^{\leq n}$, where l_{pos} describes the desired baseline state and l_{neg} describes the fundamental undesired state. For this pathway, we define:

- Positive Goal (l_{pos}): “The road is clear with no car accidents.”
- Negative Goal (l_{neg}): “Two cars have collided with each other on the road.”

Based on this definition, we can formalize the static reward function. Specifically, we employ the CLIP model (Radford et al., 2021) as the foundation for semantic reward computation.

Definition 2 (Static Reward). Given the pre-trained CLIP model with image encoder $f_I : \mathcal{I} \rightarrow \mathcal{E}$ and language encoder $f_L : \mathcal{L}^{\leq n} \rightarrow \mathcal{E}$ mapping into a shared latent space $\mathcal{E} \subseteq \mathbb{R}^d$, the static CLG pair (l_{pos}, l_{neg}) , the static reward for observation o_t^{BEV} is:

$$R_{static}(o_t) = \alpha \cdot \text{sim}(f_I(o_t^{BEV}), f_L(l_{pos})) - \beta \cdot \text{sim}(f_I(o_t^{BEV}), f_L(l_{neg})) \quad (2)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity between embeddings:

$$\text{sim}(v_1, v_2) = \frac{v_1^\top v_2}{\|v_1\| \|v_2\|} \quad (3)$$

and $\alpha, \beta > 0$ are weighting factors with $\alpha + \beta = 1$. For simplicity, we set $\alpha = \beta = 0.5$ in this work. This formulation yields a continuous reward $R_{static} \in [-1, 1]$, encouraging states semantically similar to l_{pos} while penalizing those similar to l_{neg} .

3.2.2. Theoretical Properties

We then establish theoretical guarantees for this static reward formulation.

Lemma 1 (Boundedness). For any observation o_t and CLG pair (l_{pos}, l_{neg}) , the static reward is bounded: $R_{static}(o_t) \in [-1, 1]$.

Lemma 2 (Discriminability). The CLG formulation provides strictly greater reward discrimination than single-goal formulations. Specifically, for observations o_1, o_2 where $\text{sim}(f_I(o_1), f_L(l_{pos})) = \text{sim}(f_I(o_2), f_L(l_{pos}))$ but $\text{sim}(f_I(o_1), f_L(l_{neg})) \neq \text{sim}(f_I(o_2), f_L(l_{neg}))$, we have $R_{static}(o_1) \neq R_{static}(o_2)$, even when single-goal similarity fails to distinguish the two states.

Building on these properties, we establish that the CLG formulation induces a well-defined preference ordering over states.

Theorem 1 (Reward-Induced State Ordering). Let \mathcal{S} be the state space and define the binary relation \succeq on \mathcal{S} such that $s_1 \succeq s_2$ if and only if $R_{static}(s_1) \geq R_{static}(s_2)$. Then \succeq is a total preorder (reflexive, transitive, and total), inducing a consistent preference ranking over states aligned with the semantic safety specification (l_{pos}, l_{neg}) .

The proofs of Lemmas 1–2 follow directly from the cosine similarity bounds established in our previous work (Huang et al., 2025b). The proof of Theorem 1 is provided in Appendix A.

3.3. Dynamic Pathway

The Static Pathway provides spatial safety assessment but cannot handle complex events requiring semantic understanding. The Dynamic Pathway addresses this limitation by simulating the brain’s attention-PFC circuit, which operates on a “when-needed” basis: a fast attentional mechanism identifies salient stimuli and gates the activation of slower, high-level reasoning. As illustrated in Fig. 4, this attention-gated VLM reasoning mechanism selectively triggers expensive semantic analysis only when safety-critical situations are detected, achieving computational efficiency while preserving information fidelity for critical scenarios.

3.3.1. Attentional Gate

The human brain does not expend cognitive resources processing all visual input through the PFC; subcortical structures filter stimuli and forward only salient information. We implement this mechanism using a lightweight object detection model.

Definition 3 (Attentional Gate). Let o_t^{cam} be the front-view camera image at time t . A detection model $D(\cdot)$ produces detected objects $\mathcal{O}_t = D(o_t^{cam})$. Given a predefined set of safety-critical classes, the binary gate g_t is defined as:

$$g_t = \begin{cases} 1, & \text{if } \exists o \in \mathcal{O}_t \text{ s.t. } \text{cls}(o) \in \mathcal{C}_{critical} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $\text{cls}(o)$ returns the class label of object o .

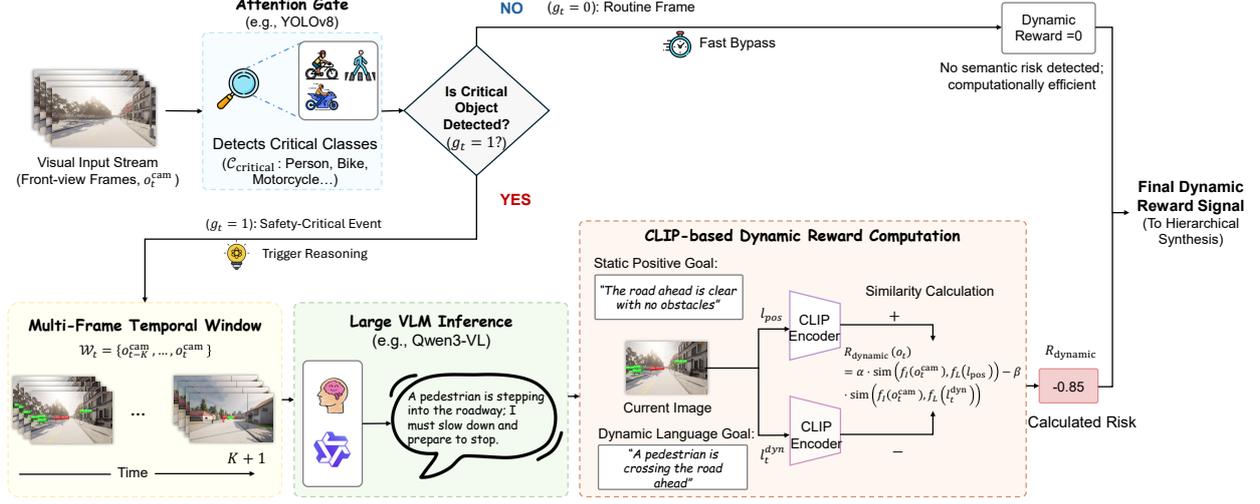


Figure 4: Attention-gated dynamic reward generation in DriveVLM-RL. Routine frames bypass semantic reasoning, while safety-critical frames trigger multi-frame LVLMM inference to produce a risk description, which is converted into a dynamic reward via CLIP-based semantic similarity.

We employ YOLOv8 (Jocher et al., 2023) as the detection model $D(\cdot)$. We set $\mathcal{C}_{critical}$ to include 11 safety-critical object categories: {person, bicycle, motorcycle, dog, horse, sheep, cow, elephant, bear, zebra, giraffe}. This set covers a wide variety of road users whose unpredictable behavior most benefits from semantic reasoning beyond spatial proximity. While the core safety-critical categories are person, bicycle, and motorcycle, we include additional animal categories to improve robustness against rare but high-risk long-tail scenarios. Vehicle-to-vehicle conflicts are primarily captured by the Static Pathway’s BEV-based spatial assessment, which provides sufficient signal for structured traffic scenarios.

3.3.2. Multi-Frame Semantic Reasoning

When $g_t = 1$, the framework simulates the PFC for semantic reasoning and the hippocampus for temporal context integration. We construct a causal temporal window of the most recent $K + 1$ frames: $\mathcal{W}_t = \{o_{t-K}^{cam}, \dots, o_t^{cam}\}$, where K is a hyperparameter controlling the temporal context window size. In our implementation, we set $K = 3$, providing a 3-frame temporal window. This window enables the VLM to understand motion dynamics and intentions rather than static snapshots.

Definition 4 (Dynamic Language Goal). The dynamic language goal l_t^{dyn} is generated by a LVLMM $F_{LVLMM}(\cdot)$ conditioned on the temporal window and detected objects:

$$l_t^{dyn} = F_{LVLMM}(\mathcal{W}_t, \mathcal{O}_t) \quad (5)$$

This output serves as a semantic hypothesis about the current risk (e.g., “A pedestrian is crossing the road ahead”).

We employ Qwen3-VL (Yang et al., 2025) as $F_{LVLMM}(\cdot)$ in our implementation. Note that $F_{LVLMM}(\cdot)$ is a LVLMM that produces natural language descriptions, distinct from the CLIP encoders used for reward computation.

3.3.3. Dynamic Reward Computation

The dynamic reward converts the VLM’s semantic understanding into a numerical signal. Unlike the Static Pathway’s fixed goal, this pathway uses the dynamically generated l_t^{dyn} as the context-specific risk description. The reward is computed using the same CLIP encoders as the Static Pathway, ensuring consistent semantic alignment.

Definition 5 (Dynamic Reward). Given the attentional gate g_t , dynamic language goal l_t^{dyn} , and static positive goal l_{pos} , the dynamic reward is:

$$R_{dynamic}(o_t) = g_t \cdot \left[\alpha \cdot \text{sim}(f_I(o_t^{cam}), f_L(l_{pos})) - \beta \cdot \text{sim}(f_I(o_t^{cam}), f_L(l_t^{dyn})) \right] \quad (6)$$

This formulation ensures $R_{dynamic} = 0$ for non-critical frames ($g_t = 0$). When triggered, it provides sparse but semantically rich penalty signals that capture complex risks beyond spatial proximity.

3.3.4. Theoretical Properties

We analyze the computational efficiency and information-theoretic properties of the attentional gating mechanism.

Lemma 3 (Computational Efficiency). *Let $p = P(g_t = 1)$ be the probability of gate activation, and let T_{LVLM} , T_{det} denote the inference time of the LVLM and detection model respectively. The expected per-frame computation time of the Dynamic Pathway is $T_{det} + p \cdot T_{LVLM}$, compared to T_{LVLM} for ungated approaches. When $p \ll 1$ and $T_{det} \ll T_{LVLM}$, this yields relative computational savings of approximately $(1 - p) \times 100\%$ compared to ungated LVLM inference.*

Theorem 2 (Information Preservation under Gating). *Let $\mathcal{S}_{critical} \subseteq \mathcal{S}$ denote the set of safety-critical states, and let μ be a distribution over $\mathcal{S}_{critical}$. Assume the detection model $D(\cdot)$ achieves recall ρ on $\mathcal{S}_{critical}$, assume $R_{LVLM}(s) \geq 0$ for all $s \in \mathcal{S}_{critical}$, and $\mathbb{E}_\mu[R_{LVLM} | g = 1] \geq \mathbb{E}_\mu[R_{LVLM}]$. Let $g(s) \in \{0, 1\}$ denote the gating indicator for state s . Then:*

$$\mathbb{E}_{s \sim \mu}[g(s) \cdot R_{LVLM}(s)] \geq \rho \cdot \mathbb{E}_{s \sim \mu}[R_{LVLM}(s)] \quad (7)$$

Combining Lemma 3 and Theorem 2, the performance of the Dynamic Pathway depends on the detector recall ρ and the gate activation probability p . In our experimental setting, the detector achieves high recall (approximately $\rho \approx 0.95$), which corresponds to 70–80% computational savings and comparable preservation of semantic information for safety-critical states. A case can be found in Appendix B

Remark 1. *The gating mechanism may fail to trigger VLM reasoning when safety-critical objects fall outside the predefined class set $\mathcal{C}_{critical}$, or when detection recall degrades due to occlusion, adverse weather, or domain shift. In such cases, during training-time reward synthesis, we fall back to the Static Pathway, which provides baseline spatial safety assessment. This graceful degradation ensures the framework remains functional, albeit with reduced semantic understanding, rather than failing catastrophically.*

The proofs are provided in Appendix C.

3.4. Hierarchical Reward Synthesis

The Static and Dynamic Pathways provide parallel assessments of spatial and semantic risk. As illustrated in Fig. 2, in the brain, such information is integrated by the vmPFC and ACC (Rangel et al., 2008), which synthesize diverse value signals into unified judgments guiding behavior. Our Hierarchical Reward Synthesis module performs this integration through a two-phase process.

3.4.1. Phase I: Semantic Reward Fusion

The first phase combines pathway outputs into a unified semantic score:

$$R_{combined}(o_t) = R_{static}(o_t) + R_{dynamic}(o_t) \quad (8)$$

This additive formulation naturally handles attentional gating: when $g_t = 0$, $R_{dynamic} = 0$ and the score defaults to spatial assessment alone.

The combined score is then normalized to $[0, 1]$:

$$R_{norm}(o_t) = \frac{\text{clip}(R_{combined}(o_t), \theta_{min}, \theta_{max}) - \theta_{min}}{\theta_{max} - \theta_{min}} \quad (9)$$

where $\text{clip}(x, a, b) = \min(\max(x, a), b)$, and $\theta_{min}, \theta_{max}$ are empirically determined hyperparameters (e.g., $\theta_{min} = -0.1$, $\theta_{max} = 0.2$). This normalized score R_{norm} represents a unified semantic safety assessment.

Corollary 1 (Bounded Final Reward). *The normalized reward satisfies $R_{norm}(o_t) \in [0, 1]$ by construction of the clipping operation in Eq. (9). Since each factor $f_{speed}, f_{center}, f_{angle}, f_{stability} \in [0, 1]$ (the former by the $\max(0, \cdot)$ operator and $v_{desired} \leq v_{max}$; the latter three by definition), their product satisfies $R_{shaping}(o_t) \in [0, 1]$. The final reward R_{final} is therefore bounded: $R_{final}(o_t) \in [R_{penalty}, 1]$.*

3.4.2. Phase II: Integration with Vehicle State

The second phase uses the normalized safety score to modulate low-level control objectives, enabling hierarchical behavior guidance.

Definition 6 (Shaping Reward). The shaping reward integrates semantic safety with vehicle state factors:

$$R_{shaping}(o_t) = f_{speed}(o_t) \cdot f_{center}(o_t) \cdot f_{angle}(o_t) \cdot f_{stability}(o_t) \quad (10)$$

where:

- $f_{\text{speed}}(o_t) = \max\left(0, 1 - \frac{|v_{\text{actual}} - v_{\text{desired}}|}{v_{\text{max}}}\right)$ measures speed tracking, with $v_{\text{desired}} = R_{\text{norm}}(o_t) \cdot v_{\text{max}}$. Here, $v_{\text{actual}} \in [0, v_{\text{max}}]$ is the current vehicle speed, ensuring $f_{\text{speed}}(o_t) \in [0, 1]$ by construction. This design encodes a safety-speed trade-off: higher semantic safety scores permit higher desired speeds, while perceived risk naturally induces conservative speed targets, down to $v_{\text{desired}} = 0$ in critical scenarios. Note that $R_{\text{norm}}(o_t)$ is computed prior to $R_{\text{shaping}}(o_t)$, using only the semantic reward components R_{static} and R_{dynamic} from Phase I;
- $f_{\text{center}}(o_t)$ evaluates lateral deviation from lane center;
- $f_{\text{angle}}(o_t)$ measures heading alignment with road direction;
- $f_{\text{stability}}(o_t)$ penalizes lateral oscillation.

Each factor is bounded in $[0, 1]$.

Remark 2. We adopt multiplicative composition rather than weighted summation for combining reward factors. This design choice ensures joint constraint satisfaction: if any factor approaches zero (e.g., severe lane deviation), the entire reward diminishes regardless of other factors. In contrast, additive formulations require careful calibration of relative weights and may allow agents to exploit high rewards in some dimensions to compensate for unsafe behaviors in others, leading to reward hacking.

3.4.3. Final Reward Function

The final reward combines the dense shaping signal with a sparse penalty for catastrophic events:

$$R_{\text{final}}(o_t) = \begin{cases} R_{\text{penalty}}, & \text{if catastrophic event occurs at } t \\ R_{\text{shaping}}(o_t), & \text{otherwise} \end{cases} \quad (11)$$

where $R_{\text{penalty}} \ll 0$ is a large negative constant applied upon collision with vehicles, pedestrians, or obstacles.

Importantly, our framework does not solely rely on this explicit punishment. As demonstrated in Section 4, the agent learns safe policies even when $R_{\text{penalty}} = 0$ under “no-reward-after-collision” settings, validating that R_{shaping} provides sufficient proactive guidance to anticipate and avoid risks.

3.4.4. Theoretical Properties

We establish that the hierarchical reward synthesis preserves the convergence properties of the underlying RL algorithm.

Theorem 3 (Policy Improvement Guarantee). Let π_k denote the policy at iteration k , and π_{k+1} the updated policy obtained under the hierarchical reward R_{final} . Under standard assumptions of soft actor-critic learning, including bounded rewards, sufficient exploration, and stable function approximation, the policy update satisfies

$$J(\pi_{k+1}) \geq J(\pi_k) - \epsilon_k \quad (12)$$

where $J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t R_{\text{final}}(o_t) \right]$, and ϵ_k denotes a bounded approximation error that diminishes as training progresses.

This guarantee follows from the SAC policy improvement theorem (Haarnoja et al., 2018), combined with the bounded reward property (Corollary 1) ensuring stable Q-value estimation. The hierarchical structure of R_{final} does not interfere with convergence because all component rewards are bounded and the shaping reward R_{shaping} is state-dependent only. While these assumptions are standard in theoretical RL analysis (Haarnoja et al., 2018), we empirically verify convergence behavior in Section 4 through training curves reported across three independent random seeds. The proofs are provided in Appendix D.

3.5. Asynchronous Batch-Processing Pipeline

The LVM inference required for R_{dynamic} is computationally expensive and unsuitable for tight closed-loop interaction, making synchronous per-step reward computation impractical within the environment loop. We address this challenge through an asynchronous batch-processing pipeline that decouples reward calculation from experience collection.

3.5.1. RL Algorithm

We employ Soft Actor-Critic (SAC) (Haarnoja et al., 2018) as the backbone RL algorithm due to its sample efficiency and stability in continuous control. Importantly, the DriveVLM-RL framework is fundamentally algorithm-agnostic: since our hierarchical reward synthesis operates independently of the policy optimization procedure, it can be seamlessly integrated with virtually any standard RL algorithm. This plug-and-play compatibility enables practitioners to leverage advances in RL algorithms while benefiting from our semantic reward design.

SAC maximizes the entropy-regularized objective:

$$J(\pi_\phi) = \mathbb{E}_{\pi_\phi} \left[\sum_{t=0}^T \gamma^t (R(o_t, a_t) + \lambda \mathcal{H}(\pi_\phi(\cdot | o_t))) \right] \quad (13)$$

Its Q -function parameters θ are updated by minimizing the standard soft Bellman residual:

$$J_Q(\theta) = \mathbb{E}_{(o_t, a_t, r_t, o_{t+1}) \sim \mathcal{D}} \left[\frac{1}{2} (Q_\theta(o_t, a_t) - (r_t + \gamma \mathbb{E}_{a_{t+1} \sim \pi_\phi(\cdot | o_{t+1})} [Q_\theta(o_{t+1}, a_{t+1}) - \lambda \log \pi_\phi(a_{t+1} | o_{t+1})]))^2 \right] \quad (14)$$

Our core modification is to replace the standard immediate reward r_t with our asynchronously computed hierarchical reward $R_{\text{final}}(o_t)$ from Eq. (11), yielding the modified Bellman residual:

$$J'_Q(\theta) = \mathbb{E}_{(o_t, a_t, o_{t+1}) \sim \mathcal{D}} \left[\frac{1}{2} (Q_\theta(o_t, a_t) - (R_{\text{final}}(o_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi_\phi(\cdot | o_{t+1})} [Q_\theta(o_{t+1}, a_{t+1}) - \lambda \log \pi_\phi(a_{t+1} | o_{t+1})]))^2 \right] \quad (15)$$

3.5.2. Pipeline Architecture

To populate the replay buffer \mathcal{D} with these $R_{\text{final}}(o_t)$ values, the pipeline operates in three parallel processes:

1. **Interaction Thread.** The agent interacts with the environment, storing transitions $(o_t, a_t, \text{images}_t)$ in \mathcal{D} with placeholder rewards ($r_t \leftarrow \text{NaN}$, $\text{ready} = 0$). This thread runs at the maximum possible environmental speed to rapidly collect raw experience.
2. **Reward Thread.** This thread runs in parallel with a separate worker. It continuously samples mini-batches $\{(o_i, a_i, o_{i+1}, \text{images}_i)\}_{i=1}^B$ from \mathcal{D} . For each transition in the batch, it executes the full hierarchical reward computation (Eqs. 2,6,8–11) and updates placeholder rewards with computed R_{final} values.
3. **Learner Thread.** The SAC learner preferentially samples reward-annotated transitions (i.e., those with $\text{ready} = 1$) from \mathcal{D} and performs policy and Q -function updates using Eq. (15). To mitigate reward staleness, the learner only begins policy updates once at least N_{warmup} transitions have been reward-annotated, ensuring Q -value estimates are predominantly trained on accurate reward signals.

This design enables experience collection to proceed without waiting for LVLM inference, maintaining high training throughput while preserving reward quality.

3.5.3. Deployment

Once training is complete, the entire VLM-based reward apparatus is discarded. During deployment, DriveVLM-RL executes only the learned policy network π_ϕ . The detector D , CLIP encoders (f_I, f_L), and the LVLM F_{LVLM} are used *only* for offline training-time reward synthesis and are not executed at test time, achieving the goal of leveraging foundation model reasoning without incurring any deployment latency. The complete training procedure is outlined in Appendix E.

4. Experiments and Results

The experiments are structured to address the following research questions: **Q1:** How does DriveVLM-RL compare with state-of-the-art reward design methods in terms of safety, efficiency, and task completion? **Q2:** Can DriveVLM-RL learn safe driving behaviors without explicit collision penalties through semantic understanding alone? **Q3:** Can the learned policy generalize to unseen environments and traffic conditions?

4.1. Experimental Setup

4.1.1. Simulation Environment

We conduct simulation experiments using the CARLA simulator (Dosovitskiy et al., 2017), a high-fidelity open-source platform for autonomous driving research. CARLA provides photorealistic rendering, accurate vehicle dynamics, and diverse urban environments essential for evaluating end-to-end driving policies.



Figure 5: Multi-modal observations of the ego vehicle in urban traffic, comprising BEV representation, semantic segmentation, and camera views with diverse traffic participants (signals, motorcyclists, cyclists, and pedestrians).

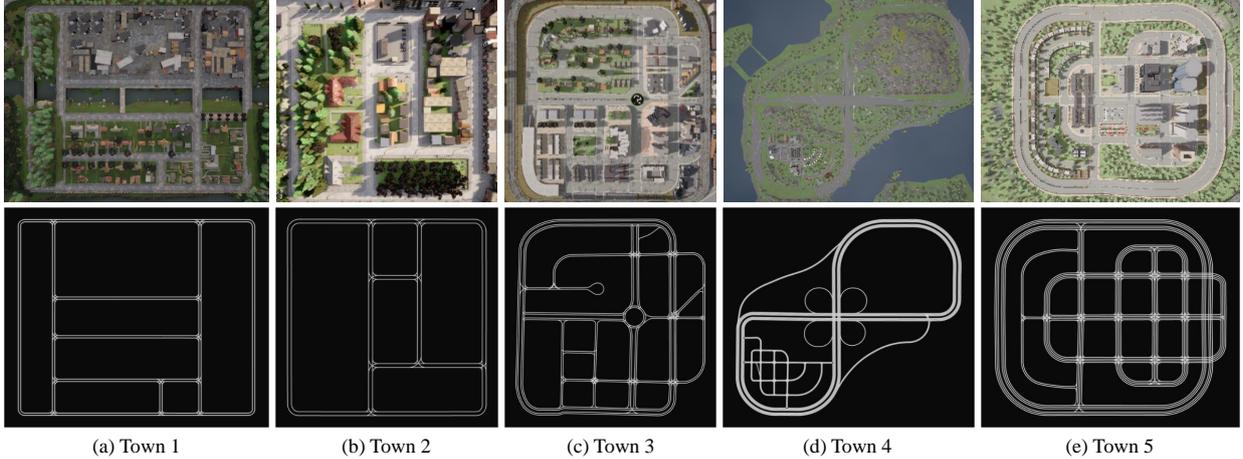


Figure 6: CARLA towns used for training and evaluation, covering diverse urban layouts and road topologies. The top row shows aerial views of the environments, and the bottom row presents the corresponding lane network structures.

1) *Training Environment.* All models in our comparative study are trained exclusively in CARLA Town 2 to ensure fair comparison and isolate the effects of different reward design approaches. Town 2 is a compact European-style urban layout featuring residential districts, commercial zones, single-lane roads, and signalized intersections. This environment provides diverse driving conditions including straight roads, curved segments, T-junctions, and varying road geometries within a manageable scale.

2) *Traffic Configuration.* Different from VLM-RL (Huang et al., 2025b), whose experimental setting contains only vehicle interactions without traffic lights or other types of road users, we construct a more complex and heterogeneous traffic environment. To evaluate robustness under realistic urban conditions, the simulation includes multiple classes of dynamic agents. Specifically, 20 vehicles generate natural traffic flow interactions; 20 pedestrians with randomized walking speeds (0.8–1.5 m/s) move around sidewalks and crosswalks to stress-test pedestrian safety; 20 motorcycles operate with short following distances (2.0 m) and $\pm 30\%$ speed variance, frequently producing cut-in behaviors; and 20 bicycles travel at substantially reduced speeds (approximately 80% lower than the speed limit), requiring safe and patient overtaking.

3) *Navigation Routes.* We employ dynamic route assignment during both training and evaluation. At each episode reset, we randomly select two distinct spawn points from the 101 predefined locations in Town 2 and compute the shortest path using the A* algorithm. Episodes continue until the cumulative driving distance reaches 3000 m, allowing comprehensive evaluation across diverse navigation scenarios within a single episode.

4) *Episode Termination.* Each training and evaluation episode continues until one of the following three termination conditions is satisfied: (i) a collision with static infrastructure, other vehicles, pedestrians, cyclists, or motorcyclists is detected; (ii) the agent becomes stuck, defined as maintaining a speed below 1 km/h for more than 90 consecutive seconds; or (iii) the lateral deviation from the lane center exceeds 3 meters, indicating loss of lane-keeping control or off-road driving.

4.1.2. Observation and Action Spaces

1) *Observation Space.* The RL agent receives: (i) a BEV semantic segmentation image rendered at a resolution of 224×224 pixels, generated by projecting CARLA ground-truth semantic labels onto a local coordinate frame centered on the ego vehicle; (ii) an ego-state vector capturing the vehicle’s dynamic state, consisting of the current steering angle (normalized to $[-1, 1]$), the throttle/brake command (normalized to $[-1, 1]$), and the instantaneous vehicle speed in km/h; and (iii) a navigation context representation composed of the next 15 future waypoints along the planned route, expressed as (x, y) coordinates in the ego-centric reference frame with the x -axis aligned with the vehicle’s heading. Waypoints are sampled at 2 m intervals, providing approximately 30 m of route preview.

2) *Action Space.* We employ a continuous two-dimensional action space $\mathcal{A} = [-1, 1]^2$. The first dimension controls steering angle (-1 : maximum left, $+1$: maximum right), while the second dimension combines throttle and

brake control (positive values map to throttle intensity, negative values to brake intensity). This end-to-end perception-to-control formulation enables direct policy deployment without any auxiliary reasoning modules at inference time.

4.1.3. Language Goal Configuration

Following the CLG paradigm, we adopt a simple yet effective contrastive prompt design. Specifically, we use “the road is clear with no car accidents” as the positive goal and “two cars have collided with each other on the road” as the negative goal. These prompts remain fixed across all experiments, highlighting the zero-shot nature of DriveVLM-RL framework and eliminating the need for task-specific prompt engineering. For the dynamic reward component, Qwen3-VL generates a context-specific risk description l_t^{dyn} conditioned on the temporal window \mathcal{W}_t and detected objects \mathcal{O}_t , following Definition 4. To constrain the output space and ensure consistent CLIP-compatible semantic embeddings, we provide the LVLM with a reference vocabulary of 10 canonical scene descriptions covering common driving risk scenarios (e.g., “a pedestrian is crossing the road ahead”, “a cyclist is merging into the ego lane”). The LVLM may draw from or adapt these descriptions when generating l_t^{dyn} , while retaining the flexibility to produce novel descriptions for out-of-distribution scenarios.

4.1.4. VLM Configuration

We employ OpenCLIP’s ViT-bigG-14 model (Ilharco et al., 2021) pre-trained on the LAION-2B dataset with 2.32 billion English image–text pairs. The model uses a patch size of 14×14 pixels and accepts 224×224 pixel images. All CLIP components remain frozen during training to ensure stable semantic reward generation. We use YOLOv8-small as the lightweight detection model for the attention gate. While YOLO can detect all 80 COCO object classes, we selectively use 11 safety-critical classes {person, bicycle, motorcycle, dog, horse, sheep, cow, elephant, bear, zebra, giraffe} to trigger VLM inference. For semantic reasoning, we employ Qwen3-VL-4B-Instruct as the LVLM, configured with a temporal window of $K = 3$ frames. Unlike CLIP’s fixed text-image matching, Qwen3-VL can generate detailed scene descriptions capturing dynamic elements. During training, the Reward Worker Thread processes stored transitions in mini-batches every $\Delta = 10$ control steps, invoking Qwen3-VL for reward annotation at an effective rate of approximately 1 Hz. YOLO pre-filtering via the attentional gate further reduces unnecessary Qwen3-VL calls by skipping transitions where no safety-critical objects are detected ($g_t = 0$).

4.2. Evaluation Metrics

We employ a set of quantitative metrics to evaluate both driving efficiency and safety performance:

- **Driving Efficiency Metrics.** *Average Speed (AS)* measures the mean vehicle speed over an episode. *Route Completion (RC)* is defined as the number of successfully completed navigation routes within a single episode. *Total Distance (TD)* records the cumulative distance traveled by the ego vehicle.
- **Safety Metrics.** The *Collision Rate (CR)* measures the percentage of episodes in which a collision with other vehicles or obstacles occurs, including rear-end and side collisions. To further characterize collision frequency, we report *Time-based Collision Frequency (TCF)*, defined as the number of collisions per 1000 time steps, and *Distance-based Collision Frequency (DCF)*, defined as the number of collisions per kilometer traveled. To quantify collision severity, we record the *Collision Speed (CS)* at the moment of impact. We additionally compute the *Inter-Collision Time (ICT)*, defined as the average number of time steps between consecutive collisions, which reflects the temporal distribution of safety-critical events.
- **Task Success.** During the test phase, we report the *Success Rate (SR)*, defined as the fraction of trials in which the agent successfully reaches the destination across 10 predefined evaluation routes. *Average Collision (AC)* represents the average number of collisions per episode.

Detailed metric definitions follow (Huang et al., 2025b).

4.3. Baseline Methods

We compare against 13 representative methods spanning three reward design paradigms. To assess the applicability of reward designs across both on-policy and off-policy algorithms, we additionally implement PPO variants for methods originally proposed with SAC, enabling evaluation of reward transferability across learning paradigms. All baselines use identical network architectures, observation/action spaces, and training hyperparameters to ensure fair comparison.

1) *Expert-Designed Reward Methods.* We implement the following baselines with manually crafted reward functions:

- **TIRL-SAC/PPO** (Cao et al., 2022): Binary reward with -1 for collision and 0 otherwise, representing minimal reward informativeness.
- **Chen-SAC** (Chen et al., 2021): Hand-tuned weighted reward balancing collision penalty, speed incentive, lane centering, and steering smoothness.

- **ASAP-RL-PPO** (Wang et al., 2023): Skill-based reward providing positive incentives for route progress, destination arrival, and overtaking, with penalties for collisions and boundary violations.
- **ChatScene-SAC/PPO** (Zhang et al., 2024): Smoothness-focused reward penalizing longitudinal acceleration, lateral acceleration, and abrupt steering changes, with a constant baseline signal to stabilize learning.

2) *LLM-Designed Reward Methods*. We compare against recent approaches that leverage LLMs for automated reward generation:

- **Revolve / Revolve-Auto** (HAZRA et al., 2025): An evolutionary framework using LLMs to generate reward function code guided by human feedback. We adopt their best-performing reward function for comparison.

3) *VLM-Designed Reward Methods*. We compare against five VLM-based reward shaping approaches:

- **VLM-SR** (Baumli et al., 2023): Binary reward using CLIP similarity thresholding to determine goal achievement.
- **RoboCLIP** (Sontakke et al., 2023): Episodic reward computing average CLIP similarity between trajectory frames and a task descriptor.
- **VLM-RM** (Rocamonde et al., 2024): Generates continuous reward by projecting the current state embedding onto the direction vector between baseline and target state descriptions.
- **LORD** (Ye et al., 2025): Penalizes similarity to dangerous states using negative language goals.
- **VLM-RL** (Huang et al., 2025b): A contrasting language goal formulation encouraging similarity to safe states while penalizing similarity to dangerous states.

4.4. Implementation Details

Our implementation is built upon the Stable-Baselines3 library (Raffin et al., 2021), which provides reliable implementations of modern RL algorithms. The standard implementations of SAC and PPO are extended to incorporate our dual-pathway reward computation architecture during the training process. The policy network accommodates heterogeneous input modalities: a 6-layer convolutional neural network (CNN) extracts visual features from the BEV semantic segmentation images, while separate multi-layer perceptrons (MLPs) process the ego-state variables and future navigation waypoints. The resulting feature embeddings are concatenated and passed to a shared policy head to generate the final control actions. All experiments are conducted on a workstation equipped with three NVIDIA RTX A6000 GPUs (each with 48 GB memory, 10,752 CUDA cores), an AMD Ryzen Threadripper Pro 7985WX processor (64 cores, 128 threads), and 512 GB system memory.

4.5. Main Results

Table 1 reports the mean and standard deviation of key metrics at the final training checkpoint, averaged over three independent runs with different random seeds to ensure robustness and reliability. The results are organized by reward design category to facilitate analysis of different paradigm strengths and limitations. We also report both training dynamics (Figs. 7–9) and testing performance (Table 2) to provide a complete picture of learned policy quality.

4.5.1. Training Performance Analysis

1) *Comparison with expert-designed binary rewards*. Binary-reward methods such as TIRL-SAC appear to achieve relatively low collision rates early in training (Fig. 7(a)). However, this behavior is misleading. The corresponding efficiency metrics in Figs. 7(d)–(f) reveal that the agent hardly moves, resulting in minimal completed routes and short traveled distance. This indicates a well-known failure mode of sparse penalties: the agent minimizes risk by avoiding driving. DriveVLM-RL avoids this local optimum. Instead of suppressing exploration, the agent progressively increases average speed and route completion while keeping collision rate low. This demonstrates that the learned policy is not merely risk-averse but capable of purposeful navigation.

2) *Comparison with summation-reward methods*. As illustrated in Fig. 8 and Table 1, Chen employs a weighted reward that balances speed incentives, collision penalties, lane centering, and steering smoothness. This method achieves the highest average speed among baselines at 25.06 km/h, demonstrating effective speed optimization. However, this aggressive driving style comes at a significant safety cost: the collision rate reaches 0.293, with a DCF of 2.71 collisions per kilometer. The learning curves in Figs. 8(a)–(c) show that Chen-SAC maintains consistently high collision rates throughout training, suggesting that its reward function over-emphasizes efficiency at the expense of safety.

ASAP takes a more conservative approach with explicit penalties for boundary violations and careful reward shaping for overtaking maneuvers. While ASAP achieves an extremely low collision speed (CS = 0.04 km/h), its high collision rate (0.403) reveals a tendency to creep into obstacles at negligible speed—indicating that the agent avoids

Table 1: Performance comparison with baselines during training. Mean and standard deviation over 3 seeds. The best results are marked in **bold**.

Model	Reference	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	CR \downarrow	ICT \uparrow	DCF \downarrow	TCF \downarrow
<i>Expert-designed Reward Methods (Binary Rewards)</i>									
TIRL-SAC	TR-C'22	15.20 \pm 5.11	0.26 \pm 0.25	14.18 \pm 6.51	4.02 \pm 5.48	0.083 \pm 0.07	36935 \pm 58721	130.77 \pm 28.73	3.028 \pm 2.15
<i>Expert-designed Reward Methods (Summation Rewards)</i>									
Chen	T-ITS'22	25.06 \pm 0.20	2.28 \pm 0.76	560.92 \pm 265.32	3.02 \pm 1.25	0.293 \pm 0.05	2071 \pm 593	2.71 \pm 0.75	1.833 \pm 0.75
ASAP	RSS'23	18.66 \pm 3.37	0.75 \pm 0.39	67.17 \pm 46.87	0.04 \pm 0.05	0.403 \pm 0.16	21398 \pm 14202	34.71 \pm 25.75	0.426 \pm 0.25
ChatScene-SAC	CVPR'24	18.00 \pm 0.07	5.54 \pm 2.14	2116.2 \pm 777.70	0.33 \pm 0.30	0.800 \pm 0.11	4967 \pm 2390	0.62 \pm 0.06	0.287 \pm 0.01
<i>LLM-based Reward Methods</i>									
Revolve	ICLR'25	18.46 \pm 0.71	2.78 \pm 0.63	910.12 \pm 283.56	0.53 \pm 0.56	0.767 \pm 0.13	2493 \pm 730	1.18 \pm 0.39	0.493 \pm 0.10
Revolve-auto	ICLR'25	17.92 \pm 2.06	3.65 \pm 0.44	1283.8 \pm 313.86	0.50 \pm 0.25	0.930 \pm 0.03	4702 \pm 2254	0.81 \pm 0.19	0.281 \pm 0.17
<i>VLM-based Reward Methods (Robotic)</i>									
VLM-SR	NeurIPS'23	1.49 \pm 1.54	0.51 \pm 0.36	53.44 \pm 61.79	0.02 \pm 0.02	0.024 \pm 0.04	869244 \pm 35550	42.73 \pm 30.12	0.192 \pm 0.14
RoboCLIP	NeurIPS'23	11.05 \pm 5.74	0.73 \pm 0.51	130.17 \pm 67.78	0.008 \pm 0.01	0.097 \pm 0.10	290455 \pm 438767	47.17 \pm 46.46	0.391 \pm 0.21
VLM-RM	ICLR'24	10.86 \pm 4.57	0.66 \pm 0.24	61.81 \pm 48.32	0.14 \pm 0.19	0.067 \pm 0.06	101193 \pm 62897	35.26 \pm 24.13	0.211 \pm 0.05
<i>VLM-based Reward Methods (Autonomous Driving)</i>									
LORD	WACV'25	15.77 \pm 8.13	0.72 \pm 0.32	111.10 \pm 130.47	0.56 \pm 0.91	0.063 \pm 0.03	45880 \pm 29377	60.92 \pm 74.85	0.398 \pm 0.21
VLM-RL	TR-C'25	22.53 \pm 0.57	2.77 \pm 0.33	806.97 \pm 190.56	0.008 \pm 0.01	0.407 \pm 0.01	5017 \pm 1297	1.50 \pm 0.48	0.404 \pm 0.05
DriveVLM-RL	Ours	25.10 \pm 3.59	4.61 \pm 0.54	1282.7 \pm 104.19	0.005 \pm 0.01	0.126 \pm 0.09	11730 \pm 1968	0.33 \pm 0.51	0.065 \pm 0.11

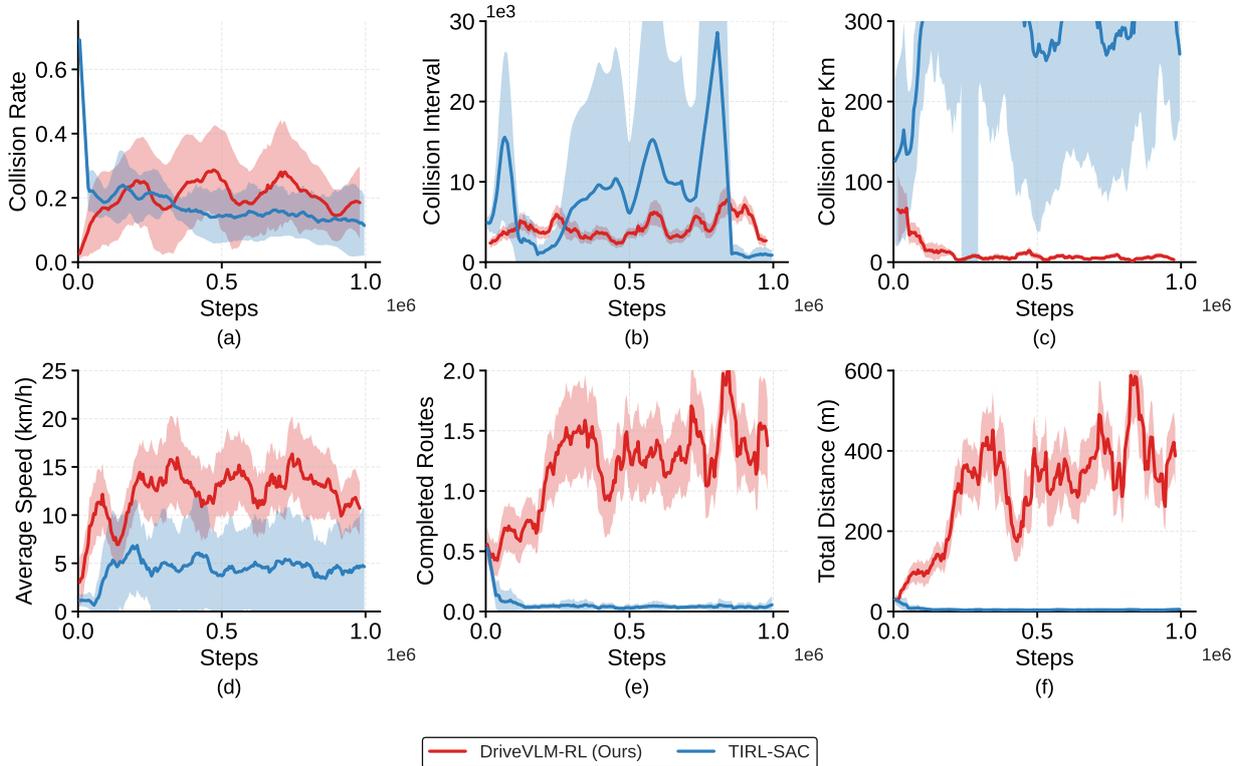


Figure 7: Training curves for expert-designed binary reward baselines. (a) Collision rate; (b) Collision interval; (c) Collision per km; (d) Average speed; (e) Completed routes; (f) Total distance. DriveVLM-RL (red) progressively improves navigation capability with route completion reaching 1.5–2.0 and total distance exceeding 400 m, while TIRL-SAC (blue) converges to a near-stationary policy with negligible route completion and extremely high collision-per-km despite a similar collision rate, confirming the sparse penalty failure mode.

high-speed impacts but cannot maintain clearance from surrounding objects without semantic scene understanding. The overall driving capability remains limited, with only 0.75 route completions and 67.17 meters traveled per episode.

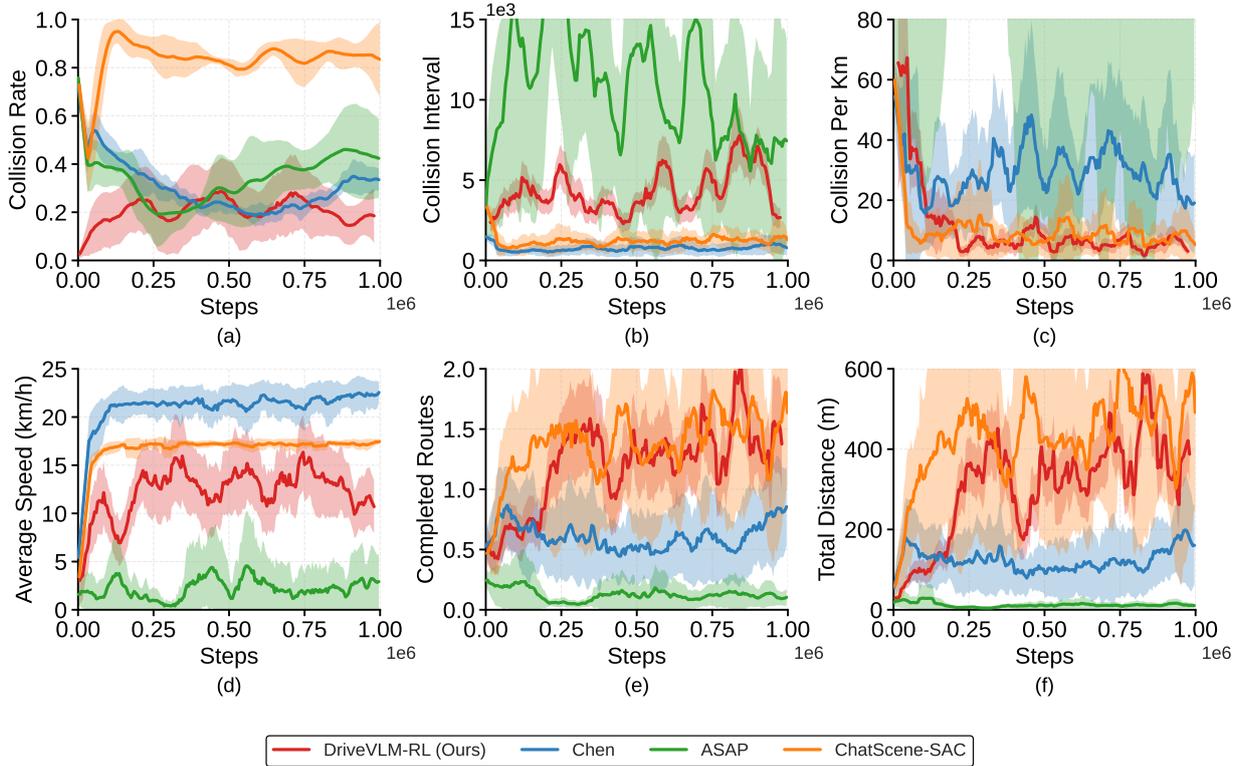


Figure 8: Training curves for expert-designed summation reward baselines. (a) Collision rate; (b) Collision interval; (c) Collision per km; (d) Average speed; (e) Completed routes; (f) Total distance. DriveVLM-RL (red) progressively reduces its collision rate to 0.15–0.20 while maintaining competitive route completion, whereas ChatScene-SAC (orange) and Chen (blue) sustain high collision rates throughout training despite achieving high speeds, and ASAP (green) nearly stalls with negligible forward progress.

ChatScene-SAC represents a smoothness-focused design that penalizes abrupt control actions. This method achieves high route completion (5.54) and total distance (2116.2 m) by sacrificing safety—a strategy incompatible with real-world deployment. The safety profile reveals critical concerns: ChatScene-SAC exhibits a collision rate of 0.800, with a DCF of 0.62 and TCF of 0.287. Fig. 8 shows that while ChatScene-SAC quickly learns to complete routes, it maintains persistently high collision rates throughout training, never developing the semantic risk anticipation capabilities that our method exhibits.

DriveVLM-RL achieves a more balanced performance profile. With 4.61 route completions and 1282.7 m traveled, our method demonstrates strong navigation capability while maintaining substantially superior safety metrics. The collision rate of 0.126 represents an 84% reduction compared to ChatScene-SAC (0.800→0.126) and a 57% reduction compared to Chen-SAC (0.293→0.126). More critically, the DCF of 0.33 collisions/km and TCF of 0.065 collisions per 1000 steps indicate that DriveVLM-RL encounters far fewer safety-critical situations during actual driving.

3) *Comparison with LLM-designed methods.* As shown in Fig. 9 and Table 1, both Revolve and Revolve-auto leverage LLMs to generate reward function code through evolutionary search with human feedback. These methods achieve moderate performance with Revolve completing 2.78 routes at 18.46 km/h and Revolve-auto completing 3.65 routes at 17.92 km/h. However, both methods exhibit high collision rates (0.767 and 0.930 respectively) and relatively high collision speeds (0.53 and 0.50 km/h).

DriveVLM-RL demonstrates clear advantages over LLM-based approaches in both safety and efficiency metrics. Our method achieves an 83% reduction in collision rate compared to Revolve (0.767→0.126) and an 86% reduction compared to Revolve-auto (0.930→0.126), while simultaneously completing 66% and 26% more routes respectively. This performance gap can be attributed to fundamental differences in reward design philosophy. LLM-based methods rely on language models to translate high-level safety specifications into reward function code, but this translation is constrained by the LLM’s limited understanding of driving dynamics and its inability to ground abstract safety concepts in visual observations. In contrast, DriveVLM-RL leverages VLMs to directly assess semantic safety from visual inputs, enabling grounded reasoning about traffic situations that is difficult to capture in hand-written code.

4) *Comparison with VLM-Based Robotic methods.* We evaluate three prominent approaches: VLM-SR, RoboCLIP, and VLM-RM. As shown in Fig. 10 and Table 1, all three methods exhibit severe performance degradation in our driving environment. VLM-SR, which uses CLIP similarity thresholding to generate binary rewards, achieves an average speed of only 1.49 km/h, completing 0.51 routes and traveling 53.44 m per episode. RoboCLIP and VLM-RM show similarly poor navigation capabilities with average speeds of 11.05 km/h and 10.86 km/h respectively. The learning curves in Figs. 10(d)–(f) reveal that these methods struggle to discover basic driving behaviors, with speed and distance metrics remaining near zero throughout most of the training process.

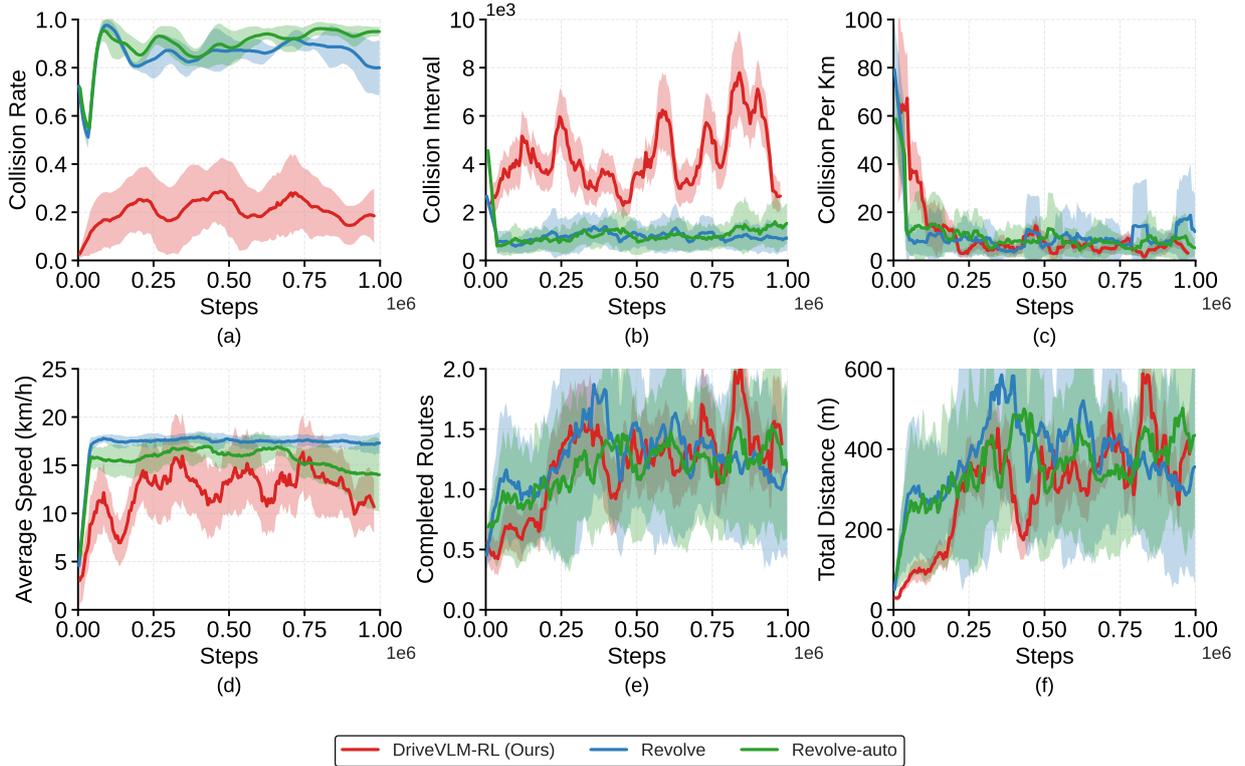


Figure 9: Training curves for LLM-designed reward baselines. (a) Collision rate; (b) Collision interval; (c) Collision per km; (d) Average speed; (e) Completed routes; (f) Total distance. DriveVLM-RL (red) maintains a consistently lower collision rate (0.15–0.25) than Revolve (blue) and Revolve-auto (green), which both plateau near 1.0 throughout training, while achieving comparable route completion and distance despite the substantially safer driving profile.

This failure of robotic VLM methods highlights a fundamental domain gap between manipulation and autonomous driving. In robotic tasks, goal states can be precisely specified with language descriptions such as “grasp the red cube” or “place the object in the bowl,” and VLMs can effectively measure progress toward these well-defined visual goals. However, driving objectives such as “drive safely” are inherently ambiguous and cannot be reduced to simple visual pattern matching. The dynamic, multi-agent nature of traffic environments, combined with the need for continuous control at varying speeds, requires reward signals that capture temporal evolution, motion dynamics, and context-dependent risk assessment—capabilities that single-frame CLIP similarity scores cannot provide.

5) *Comparison with VLM-Based Driving Methods.* We compare against recent VLM-reward methods specifically designed for autonomous driving: LORD and our previous work VLM-RL. As shown in Fig. 11 and Table 1, these methods represent the current state of the art in applying VLMs to driving reward design.

LORD, which uses negative language goals to describe dangerous states, demonstrates learning instability in our complex traffic environment. The method achieves only 0.72 route completions at 15.77 km/h with 111.10 m traveled per episode. While it maintains a relatively low collision rate of 0.063, the extremely limited navigation capability suggests overly conservative behavior—a failure mode analogous to TIRL-SAC, where the agent avoids collisions by suppressing movement rather than through genuine risk reasoning.

VLM-RL, our previous work using contrasting language goals with CLIP, achieves substantially better performance with 2.77 route completions at 22.53 km/h and 806.97 m traveled. However, the collision rate of 0.407 and collision speed of 0.008 km/h indicate that static CLG formulations have limited capacity to handle the dynamic, context-dependent safety reasoning required in heterogeneous traffic with pedestrians, motorcycles, and bicycles. Because VLM-RL relies exclusively on fixed language goals and single-frame BEV images, it cannot perceive evolving risk situations such as a pedestrian approaching the curb or a cyclist preparing to merge.

DriveVLM-RL demonstrates significant improvements over both driving-specific baselines. Compared to VLM-RL, our dual-pathway architecture achieves a 69% reduction in collision rate (0.407→0.126), a 66% increase in route completion (2.77→4.61), and a 59% increase in distance traveled (806.97→1282.7 m). Compared to LORD (0.063), DriveVLM-RL achieves a substantially higher collision rate but dramatically better navigation capability (4.61 vs 0.72 routes), confirming that our method achieves a genuine safety–efficiency balance rather than trading mobility for collision avoidance. This progressive improvement over prior VLM-based driving methods indicates that the Dynamic Pathway successfully captures evolving traffic situations through attention-gated multi-frame reasoning, enabling the agent to learn context-aware risk anticipation rather than merely pattern-matching to static safety descriptions.

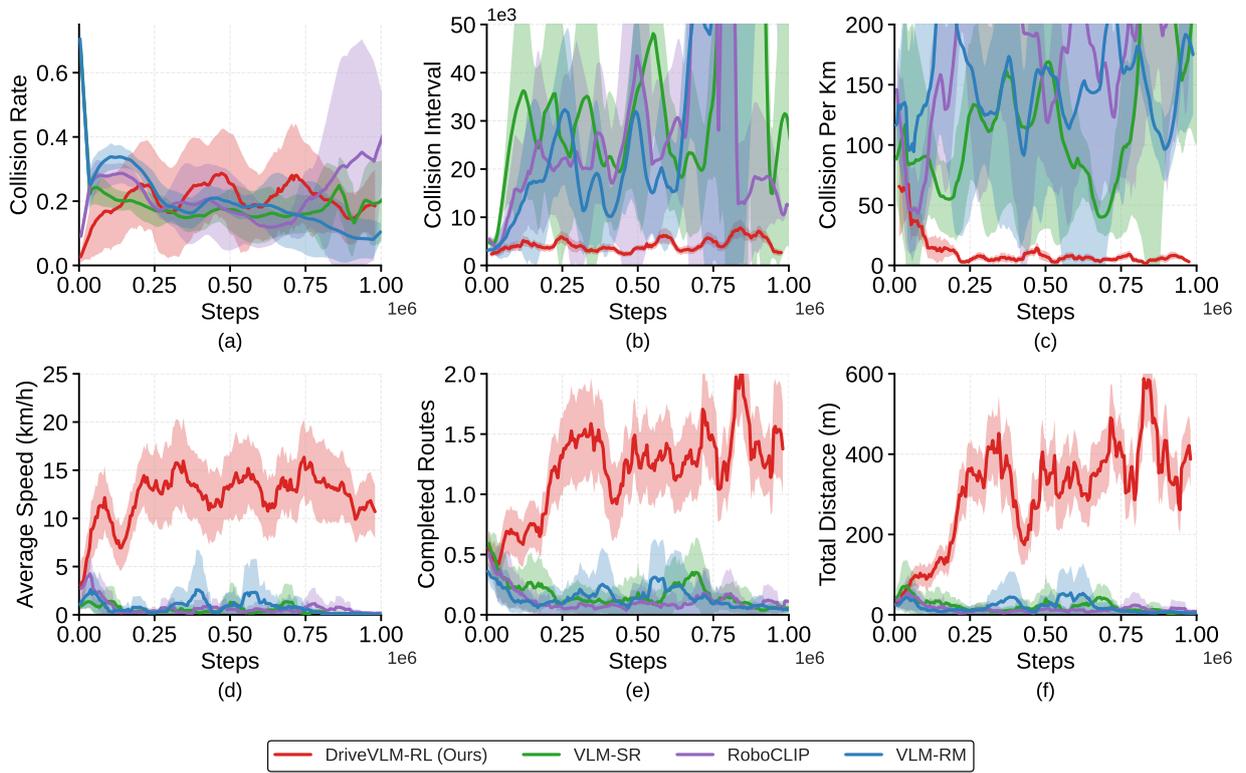


Figure 10: Training curves for VLM-based robotic reward baselines. (a) Collision rate; (b) Collision interval; (c) Collision per km; (d) Average speed; (e) Completed routes; (f) Total distance. DriveVLM-RL (red) demonstrates substantially higher navigation capability across all metrics, while VLM-SR (green), RoboCLIP (purple), and VLM-RM (blue) fail to develop meaningful driving behaviors, remaining near-stationary throughout training despite low collision rates.

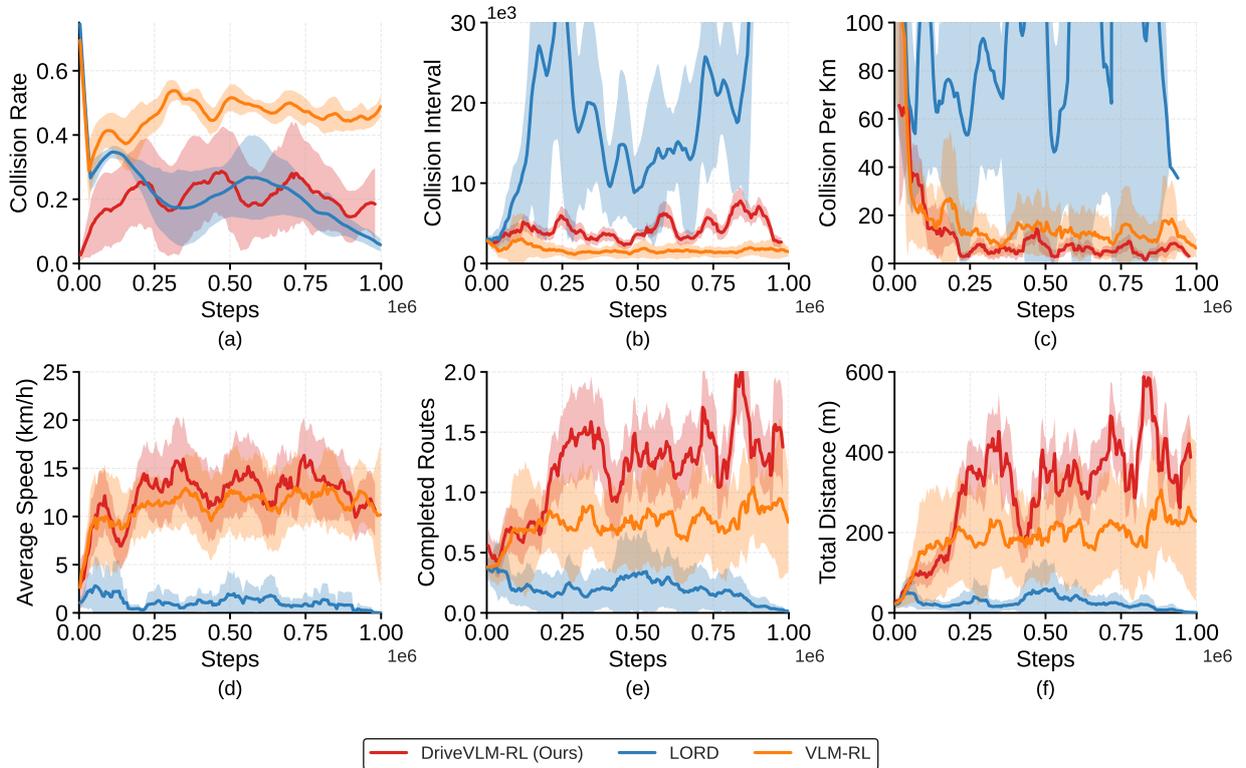


Figure 11: Training curves for VLM-based autonomous driving reward baselines. (a) Collision rate; (b) Collision interval; (c) Collision per km; (d) Average speed; (e) Completed routes; (f) Total distance. DriveVLM-RL (red) achieves the lowest collision rate and highest navigation performance, while LORD (blue) nearly stalls with minimal forward progress despite low collision frequency.

Table 2: Performance comparison with baselines during testing. Mean and standard deviation over 3 seeds. The best results are marked in **bold**.

Model	Reference	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
<i>Expert-designed Reward Methods</i>							
TIRL-SAC	TR-C’22	0.45 \pm 0.77	0.01 \pm 0.01	1.49 \pm 2.32	0.29 \pm 0.50	0.00 \pm 0.00	0.07 \pm 0.12
Chen-SAC	T-ITS’22	24.32 \pm 0.46	0.49 \pm 0.08	162.01 \pm 17.67	16.04 \pm 2.51	0.50 \pm 0.10	0.50 \pm 0.10
ASAP	RSS’23	11.53 \pm 10.22	0.12 \pm 0.11	25.00 \pm 24.92	7.07 \pm 5.96	0.00 \pm 0.00	0.67 \pm 0.32
ChatScene-SAC	CVPR’24	17.47 \pm 0.10	0.45 \pm 0.03	161.78 \pm 13.56	10.69 \pm 5.02	0.57 \pm 0.12	0.43 \pm 0.12
<i>LLM-based Reward Methods</i>							
Revolve	ICLR’25	17.42 \pm 0.80	0.40 \pm 0.12	134.37 \pm 15.26	10.33 \pm 2.25	0.40 \pm 0.20	0.60 \pm 0.20
Revolve-auto	ICLR’25	13.54 \pm 3.38	0.39 \pm 0.04	159.91 \pm 28.35	8.21 \pm 1.74	0.57 \pm 0.15	0.43 \pm 0.15
<i>VLM-based Reward Methods (Robotic)</i>							
VLM-SR	NeurIPS’23	0.06 \pm 0.05	0.01 \pm 0.00	2.26 \pm 1.26	0.66 \pm 1.14	0.00 \pm 0.00	0.07 \pm 0.12
RoboCLIP	NeurIPS’23	0.13 \pm 0.09	0.02 \pm 0.01	3.46 \pm 2.32	0.01 \pm 0.02	0.00 \pm 0.00	0.03 \pm 0.06
VLM-RM	ICLR’24	0.08 \pm 0.01	0.02 \pm 0.00	3.60 \pm 0.38	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00
<i>VLM-based Reward Methods (Autonomous Driving)</i>							
LORD	WACV’25	0.36 \pm 0.59	0.03 \pm 0.03	4.10 \pm 6.11	1.52 \pm 2.63	0.00 \pm 0.00	0.07 \pm 0.12
VLM-RL	TR-C’25	14.38 \pm 1.53	0.51 \pm 0.08	138.08 \pm 16.68	10.09 \pm 11.93	0.40 \pm 0.00	0.10 \pm 0.10
DriveVLM-RL	Ours	14.13 \pm 1.07	0.45 \pm 0.01	174.69 \pm 8.96	0.29 \pm 0.43	0.60 \pm 0.00	0.17 \pm 0.06

4.5.2. Performance Evaluation in Testing

To assess the generalization and robustness of learned policies, we evaluate all methods on 10 predefined routes in Town 2 that were not encountered during training. Table 2 presents the testing results averaged over three random seeds, with particular focus on SR (the fraction of routes completed without collision) and AC per route.

TIRL-SAC, which already struggled during training, shows even worse test performance with an average speed of only 0.45 km/h, 0.01 route completions, and 0% success rate. Chen-SAC maintains its high speed (24.32 km/h) but achieves only a 50% success rate with 0.50 average collisions per route. ChatScene-SAC demonstrates better generalization with a 57% success rate but still experiences 0.43 collisions per route on average, suggesting that smoothness-based rewards cannot fully capture safety requirements in unseen scenarios.

Revolve achieves a 40% success rate while Revolve-auto reaches 57%. However, their collision speeds (10.33 and 8.21 km/h respectively) remain concerningly high, indicating that these methods fail to develop the anticipatory safety behaviors required for robust deployment. Despite generating reward functions through evolutionary LLM search, both approaches show that code-level reward specification cannot substitute for visual semantic grounding in novel traffic conditions.

VLM-SR, RoboCLIP, and VLM-RM all achieve 0% success rates with minimal forward progress, confirming that robotic VLM reward methods do not transfer to autonomous driving scenarios. Notably, VLM-RM achieves AC = 0.00 and CS = 0.00 km/h by remaining nearly stationary, reflecting the same risk-avoidance failure mode observed in TIRL-SAC during training. LORD similarly collapses to 0% success rate at test time despite its low training collision rate, suggesting that its negative-goal formulation overfits to training distribution and fails to generalize.

Compared to our previous work VLM-RL, DriveVLM-RL achieves a 50% higher success rate (0.60 vs 0.40) and substantially lower collision speed (0.29 vs 10.09 km/h), while traveling the greatest total distance among all methods (174.69 m vs 138.08 m). The dramatic improvement in collision speed confirms that the dual-pathway architecture with dynamic semantic reasoning enables substantially safer behavior in novel scenarios.

The strong testing performance can be attributed to three key factors in our framework. First, the Static Pathway provides a robust foundation of spatial safety assessment that generalizes across scenarios. Second, the Dynamic Pathway with attention-gated LVLM reasoning enables flexible adaptation to diverse traffic situations: rather than memorizing specific scenarios, the LVLM analyzes multi-frame observations to generate context-aware risk descriptions, allowing the policy to reason about novel pedestrian behaviors, motorcycle cut-ins, or bicycle interactions that may not have been frequently encountered during training. Third, the hierarchical reward synthesis ensures that semantic understanding is grounded in vehicle dynamics through the multiplicative reward formulation, preventing the agent from developing high-level safety concepts that are disconnected from actual control capabilities—a common failure mode in methods that rely solely on semantic rewards.

Table 3: No-reward-after-collision testing performance. Mean and standard deviation over three random seeds. The best results in each column are highlighted in **bold**.

Model	Reference	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
ChatScene-SAC	CVPR'24	17.29 \pm 0.15	0.45 \pm 0.03	164.77 \pm 7.49	10.76 \pm 2.38	0.52 \pm 0.08	0.47 \pm 0.06
VLM-RL	TR-C'25	14.59 \pm 0.51	0.44 \pm 0.06	124.03 \pm 22.14	11.54 \pm 8.33	0.40 \pm 0.10	0.17 \pm 0.06
DriveVLM-RL	Ours	14.72 \pm 2.17	0.44 \pm 0.06	166.56 \pm 15.67	0.49 \pm 0.30	0.53 \pm 0.06	0.22 \pm 0.08

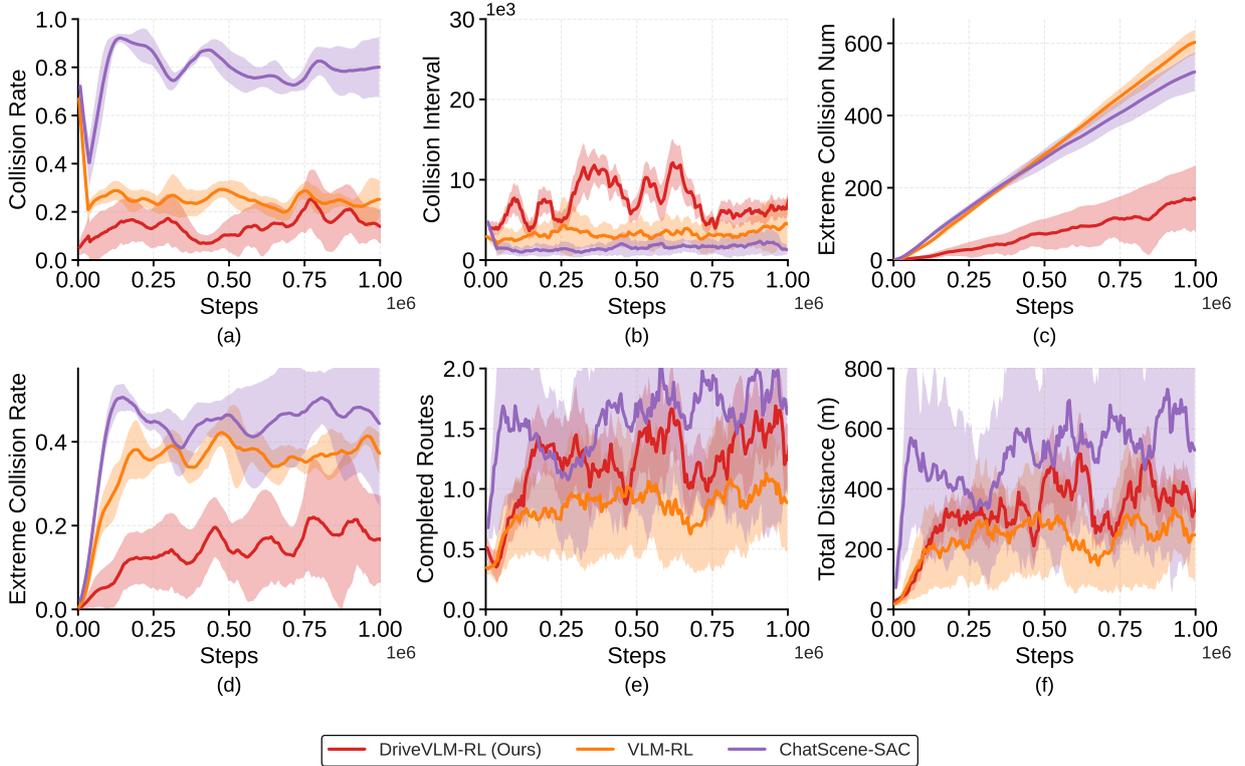


Figure 12: Training curves under the no-reward-after-collision setting ($R_{\text{penalty}} = 0$). (a) Collision rate; (b) Collision interval; (c) Cumulative extreme collision count; (d) Extreme collision rate; (e) Completed routes; (f) Total distance. DriveVLM-RL (red) consistently maintains the lowest collision rate and count throughout training, demonstrating proactive semantic safety without any explicit penalty signal.

4.6. No-Reward-After-Collision Experiment

Traditional RL approaches depend on collision-based trial-and-error, creating an insurmountable barrier to real-world deployment where physical crashes are unacceptable. To investigate whether DriveVLM-RL can learn safe behaviors through semantic understanding alone, we conduct an extreme ablation study by completely removing the collision penalty term, modifying Eq. (11) to:

$$R_{\text{final}}(o_t) = R_{\text{shaping}}(o_t), \quad R_{\text{penalty}} = 0 \quad (16)$$

We compare DriveVLM-RL with ChatScene-SAC and VLM-RL, as they achieve the strongest overall performance in the main experiments. Table 3 presents the testing results under this extreme condition.

DriveVLM-RL maintains competitive overall performance, achieving the highest success rate (SR = 0.53), the lowest collision severity (CS = 0.49 km/h), and the longest travel distance (TD = 166.56 m). In contrast, VLM-RL exhibits high average speed but suffers from severe collisions (CS = 11.54 km/h), indicating that without penalty signals the policy drives aggressively without recognizing hazardous situations. ChatScene-SAC achieves the highest average speed and route completion but still produces frequent collisions (CS = 10.76 km/h). Notably, removing explicit collision penalties causes clear safety degradation in both baselines, whereas the impact on DriveVLM-RL is substantially smaller.

ChatScene-SAC without collision penalty. As shown in Fig. 12(a), the collision rate rapidly increases to nearly 1.0 within 250k steps and remains persistently high throughout training. The cumulative extreme collision count steadily exceeds 600 by 1M steps (Fig. 12(c)), and the extreme collision rate stabilizes around 0.4–0.5 (Fig. 12(d)). These results demonstrate that without explicit penalties, the smoothness-based reward inadvertently encourages maintaining steady speeds regardless of obstacles, as sudden avoidance maneuvers incur smoothness penalties. Although route

Table 4: Ablation study on framework components during testing. Mean and standard deviation over 3 seeds. The best results are highlighted in **bold**.

Model	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
w/o First-View	12.8 \pm 2.3	0.38 \pm 0.00	98.5 \pm 32.8	4.52 \pm 3.15	0.43 \pm 0.12	0.35 \pm 0.12
w/o Attentional Gating	9.8 \pm 3.8	0.25 \pm 0.11	62.3 \pm 38.5	6.85 \pm 4.28	0.30 \pm 0.15	0.52 \pm 0.18
w/o Reward Synthesis	14.38 \pm 1.53	0.51 \pm 0.08	138.08 \pm 16.68	10.09 \pm 11.93	0.40 \pm 0.00	0.10 \pm 0.10
DriveVLM-RL (Full)	14.13 \pm 1.07	0.45 \pm 0.01	174.69 \pm 8.96	0.29 \pm 0.43	0.60 \pm 0.00	0.17 \pm 0.06

completion and total distance (Figs. 12(e)–(f)) appear relatively high, this is achieved entirely through collision-tolerant driving rather than safe navigation.

VLM-RL without collision penalty. VLM-RL shows partial success. The collision rate decreases from approximately 0.6 to 0.3–0.4 during the first 500k steps but plateaus thereafter (Fig. 12(a)), with cumulative extreme collisions reaching approximately 500–600 by 1M steps (Fig. 12(c)). The extreme collision rate similarly plateaus around 0.4 (Fig. 12(d)), and route completion stabilizes around 1.0–1.5 (Fig. 12(e)). This plateau reveals the fundamental limitation of static language goals: fixed CLIP-based rewards cannot capture context-dependent risks such as pedestrians stepping onto roads or motorcycles initiating cut-ins, making further collision reduction impossible without an explicit penalty signal.

DriveVLM-RL without collision penalty. DriveVLM-RL demonstrates remarkable robustness despite zero collision penalties. The collision rate remains consistently low at 0.15–0.2 throughout training (Fig. 12(a))—only slightly higher than the main experiment result of 0.126. The collision interval is substantially higher than both baselines (Fig. 12(b)), and the extreme collision count accumulates to only 150–200 by 1M steps (Fig. 12(c)), representing an approximately 67% reduction compared to both ChatScene-SAC and VLM-RL (\approx 500–600). The extreme collision rate remains below 0.2 throughout (Fig. 12(d)), and route completion reaches 1.5–1.6 (Fig. 12(e)) with total distance trending upward (Fig. 12(f)).

These observations confirm that baseline methods primarily rely on penalty-based learning: collisions are avoided only when the environment explicitly discourages them. In contrast, DriveVLM-RL continues to avoid hazardous situations even in the complete absence of penalty signals. This suggests that the policy has internalized a predictive safety criterion: because the Dynamic Pathway evaluates contextual risk semantically during training, the agent learns to anticipate unsafe situations (e.g., approaching pedestrians or conflicting vehicles) rather than merely reacting to collision outcomes.

4.7. Ablation Study

4.7.1. Component Analysis

1) *The Impact of First-View Camera Input.* In this variant, the Dynamic Pathway uses BEV observations instead of front-view camera images to compute dynamic rewards (Eq. (6)). Training performance degrades moderately (RC: 4.61 \rightarrow 3.45, CR: 12.6% \rightarrow 21.8%), and testing shows reduced success rates (SR: 60% \rightarrow 43%, AC: 0.17 \rightarrow 0.35). While BEV representations excel at spatial relationships, they lack the depth cues and perspective context necessary for interpreting pedestrian intent and vehicle motion dynamics, causing the LVLM to generate less precise risk descriptions (e.g., “A person-shaped object is detected” instead of “A pedestrian is crossing the road ahead”). Notably, this variant still outperforms the w/o Reward Synthesis baseline (SR: 43% vs. 40%), confirming that the remaining components provide robustness even when one modality is degraded.

2) *The Impact of Attentional Gating.* Removing the YOLO-based attentional gate and invoking LVLM inference on every frame severely degrades both training efficiency and deployment safety. Training performance collapses (RC: 4.61 \rightarrow 2.92, CR: 12.6% \rightarrow 38.5%), and testing reaches the lowest levels across all variants (SR: 60% \rightarrow 30%, AC: 0.17 \rightarrow 0.52). This counterintuitive result (more frequent semantic evaluation leading to worse performance) can be attributed to two factors. First, invoking LVLM on every frame floods the reward signal with low-quality annotations from routine, hazard-free transitions, where LVLM outputs are inherently noisy and unstable. These spurious rewards (e.g., “A vehicle might suddenly appear” on a clear road) corrupt the policy gradient and obscure the informative signal from genuine safety-critical events. Second, the increased annotation workload strains the asynchronous pipeline, causing reward timestamps to lag behind policy updates and introducing temporal inconsistency in training. This validates our design principle: semantic reasoning should activate selectively on safety-critical stimuli, ensuring that LVLM rewards are both informative and temporally aligned.

3) *The Impact of Hierarchical Reward Synthesis.* This ablation removes the multiplicative integration with vehicle state functions (Eqs. (10)–(11)), retaining only the combined semantic reward R_{combined} from the Static and Dynamic Pathways as the sole training signal. Training converges to a suboptimal policy (CR: 40.7%), and testing shows limited success (SR: 40%, AC: 0.10). This variant achieves the *lowest* average collision count (AC = 0.10) yet also the lowest task success and high collision speeds when impacts do occur (CS = 10.09 \pm 11.93 km/h), revealing a

Table 5: YOLOv8 variant comparison for attentional gating. Infer: average inference time per frame (s); Objects: total detections across all frames; Key: detections of safety-critical classes that trigger LVLM inference; VLM%: percentage of frames invoking LVLM.

Episode (Frames)	Model	Infer (s)	Objects	Key	VLM%
Episode 1 (104)	YOLOv8n	0.020*	107	13	10.6
	YOLOv8m	0.028	231	18	15.4
	YOLOv8x	0.044	280	23	18.3
Episode 2 (142)	YOLOv8n	0.020	171	73	33.8
	YOLOv8m	0.020	320	118	48.6
	YOLOv8x	0.032	396	123	49.3
Episode 3 (202)	YOLOv8n	0.020	181	24	8.4
	YOLOv8m	0.021	402	24	10.4
	YOLOv8x	0.034	501	27	11.9
Episode 4 (522)	YOLOv8n	0.021	1388	634	54.8
	YOLOv8m	0.021	1877	717	66.7
	YOLOv8x	0.033	2099	798	70.5

*Episode 1 inference time for YOLOv8n reflects cold-start model loading; steady-state inference time is 0.020 s, consistent with subsequent episodes. Model load times: YOLOv8n = 0.047 s, YOLOv8m = 0.071 s, YOLOv8x = 0.128 s.

“defensive stagnation” failure mode. Without multiplicative coupling to speed tracking, lane centering, and heading alignment, the agent can maximize semantic safety rewards by remaining nearly stationary—a behavior that satisfies “the road is clear” semantically while failing to navigate. When the agent does attempt forward motion, the absence of dynamic state guidance produces poorly-timed maneuvers and high-speed impacts (reflected in the large standard deviation of CS). The full framework’s multiplicative composition ensures that semantic safety and vehicle dynamics objectives must be satisfied simultaneously, grounding abstract risk reasoning in concrete control requirements.

4.7.2. Attentional Gating Efficiency

We evaluated three YOLOv8 (Jocher et al., 2023) variants of increasing model capacity: YOLOv8n, YOLOv8m, and YOLOv8x. As shown in Table 5, all three variants achieve similar steady-state inference times (0.020–0.021 s per frame), but differ substantially in detection coverage. YOLOv8m detects 1.5–2× more objects than YOLOv8n across all episodes (e.g., 1877 vs. 1388 in Episode 4), while triggering LVLM inference on 66.7% of frames compared to 54.8% for YOLOv8n—a meaningful increase in safety-critical coverage at no additional runtime cost. YOLOv8x achieves marginally higher coverage (70.5%) but requires 57% more inference time (0.033 s vs. 0.021 s).

Fig. 13 illustrates qualitative detection results across four representative cases. YOLOv8n shows notable limitations: in Case (ii) it fails to detect a distant pedestrian (a false negative that suppresses LVLM inference), while in Cases (iii) and (iv) it misclassifies a fire hydrant as a pedestrian (a false positive that unnecessarily triggers LVLM reasoning). In Case (i), it mistakes a rock for a car, whereas YOLOv8m suppresses this error and YOLOv8x produces overly detailed segmentation that introduces redundant KEY detections. Based on these observations, we adopt YOLOv8m as the default detector, as it offers the best balance between detection accuracy, robustness to false positives, and runtime efficiency.

4.7.3. Impact of LVLM Backbone Capacity

Table 6 compares DriveVLM-RL performance across three Qwen3-VL variants. The results reveal that performance is relatively insensitive to LVLM capacity beyond a minimum threshold: the 4B and 8B models achieve similar success rates (SR: 0.60 vs. 0.63) and collision severities (CS: 0.29 vs. 0.24 km/h), with differences largely within one standard deviation. This robustness stems from three properties of our framework. First, the LVLM is tasked only with scene description rather than complex multi-step reasoning—a capability that is well-handled even by compact models. Second, the reference vocabulary of canonical scene descriptions (Section 4.1.3) constrains the output space, reducing the task to structured selection rather than open-ended generation and minimizing sensitivity to model capacity. Third, and most importantly, the LVLM is invoked only during training to annotate reward signals; at inference time, the deployed policy operates entirely without LVLM calls. As a result, the quality of the learned reward signal matters more than the raw capacity of the model generating it. The 2B model shows a more noticeable performance gap (SR: 0.50, CS: 1.38 km/h), suggesting that a minimum level of semantic understanding is required to generate coherent scene descriptions. We adopt Qwen3-VL-4B as our default, as it surpasses this threshold while maintaining a practical inference footprint for asynchronous reward annotation during training.

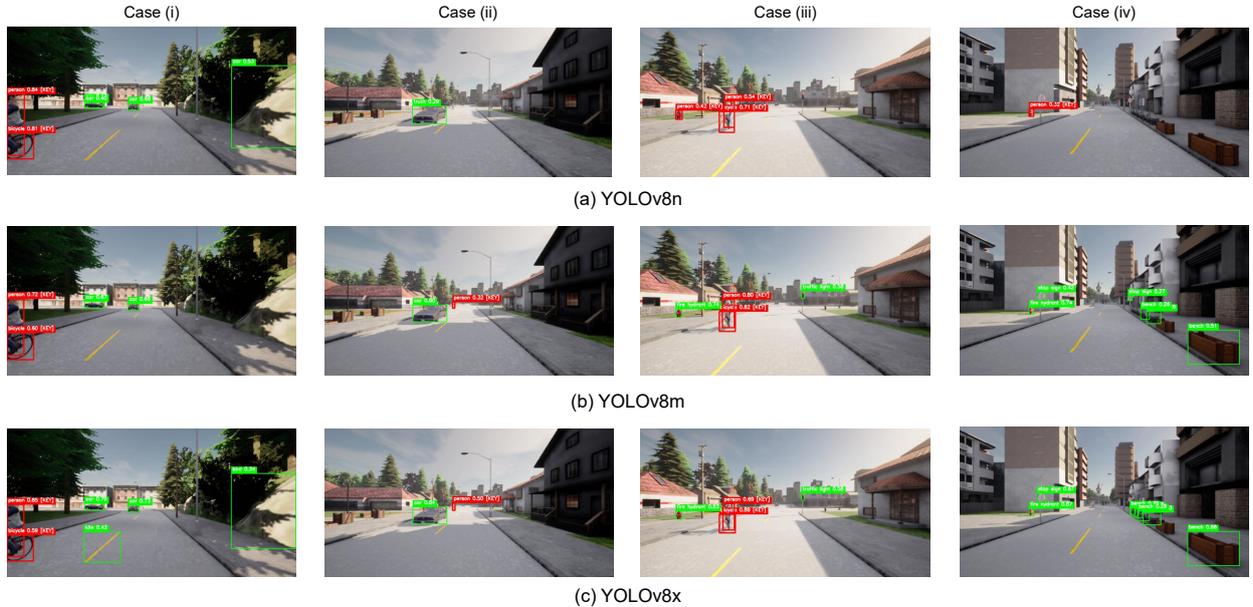


Figure 13: Visualization of YOLOv8 Detection Results in Four Driving Cases. Red bounding boxes labeled with [KEY] denote critical objects (e.g., pedestrians, bikes) that would activate VLM reasoning, while green boxes indicate non-critical detections.

Table 6: Performance comparison using different LVLM backbones. Mean and standard deviation over 3 seeds. The baseline configuration is highlighted.

LVLM Model	Params	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
Qwen3-VL-2B	2B	13.2 \pm 1.5	0.38 \pm 0.06	152.00 \pm 24	1.38 \pm 0.95	0.50 \pm 0.08	0.25 \pm 0.08
Qwen3-VL-4B (ours)	4B	14.13 \pm 1.07	0.45 \pm 0.01	174.69 \pm 8.96	0.29 \pm 0.43	0.60 \pm 0.00	0.17 \pm 0.06
Qwen3-VL-8B	8B	14.5 \pm 1.2	0.46 \pm 0.03	178.3 \pm 10.4	0.24 \pm 0.39	0.63 \pm 0.06	0.15 \pm 0.05

4.8. Generalization to Unseen Environments

4.8.1. Cross-Town Generalization

To evaluate generalization beyond the training environment, we test all methods across five distinct CARLA urban layouts (Towns 1–5), where Town 2 is the training environment and Towns 1, 3–5 are fully out-of-distribution. Results are summarized in Table 7. In Towns 1 and 3, which share compact urban geometry with Town 2, DriveVLM-RL achieves SR of 57% and 17% respectively, while ChatScene-SAC collapses to 0% in both towns despite achieving 57% in Town 2, confirming that smoothness-based rewards overfit to training road geometry. Town 4’s large-scale highway layout represents the most challenging distribution shift: DriveVLM-RL achieves SR = 0.07 (slightly below VLM-RL’s 0.13) but maintains the longest travel distance (438.40 vs. 357.78 m), with the gap in SR likely attributable to the domain shift in risk semantics from pedestrian-rich to highway-specific hazards. Town 5’s multi-level road structure constrains all methods (SR: 0.00–0.10), though DriveVLM-RL again achieves the highest SR (0.10) and distance (86.59 m). Aggregating across all out-of-distribution towns, DriveVLM-RL achieves a mean SR of 22.75% vs. VLM-RL’s 9.00% and ChatScene-SAC’s 1.75%, representing improvements of 153% and 1200% respectively, while also identifying highway and multi-level environments as directions for future work.

4.8.2. Traffic Density Robustness

We evaluate all methods under three distinct traffic densities: Empty (0 vehicles), Regular (20 vehicles, matching training), and Dense (40 vehicles). Results are summarized in Table 8. Under Empty conditions, all methods achieve zero collision rates, but DriveVLM-RL attains a perfect SR of 1.00 and the longest travel distance (208.49 m), while ChatScene-SAC and VLM-RL reach only 0.60 and 0.83 respectively, suggesting that the semantic reward structure encourages more complete route-following even in the absence of interactive hazards. Under Regular density (the training condition), DriveVLM-RL maintains the best safety profile (CS = 0.29 km/h vs. 10.09–10.69 km/h for baselines) and the highest SR (0.60). Under Dense conditions, all methods degrade, but DriveVLM-RL exhibits the most graceful degradation: SR drops from 0.60 to 0.50 (17% reduction), compared to ChatScene-SAC’s drop from 0.57 to 0.40 (30% reduction) and VLM-RL’s drop from 0.40 to 0.40 (unchanged but with substantially higher collision speeds). Critically, DriveVLM-RL’s CS remains low at 0.75 km/h under Dense traffic, compared to 13.58 km/h for ChatScene-SAC and 7.38 km/h for VLM-RL, demonstrating that the Dynamic Pathway’s attention-gated semantic reasoning scales effectively to high-density interaction scenarios.

Table 7: Cross-town generalization performance. Models are trained exclusively on Town 2 and evaluated on Towns 1–5. Mean and standard deviation over 3 seeds. The best result in each column within a town is in **bold**.

Town	Model	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
Town 1	ChatScene-SAC	13.20 \pm 1.45	0.22 \pm 0.04	122.17 \pm 18.32	12.67 \pm 0.91	0.00 \pm 0.00	0.43 \pm 0.12
	VLM-RL	14.95 \pm 0.95	0.24 \pm 0.03	191.74 \pm 14.26	5.02 \pm 3.61	0.20 \pm 0.10	0.37 \pm 0.06
	DriveVLM-RL	17.91 \pm 0.05	0.24 \pm 0.07	375.77 \pm 69.53	0.73 \pm 0.21	0.57 \pm 0.12	0.30 \pm 0.10
Town 2 [†]	ChatScene-SAC	17.47 \pm 0.10	0.45 \pm 0.03	161.78 \pm 13.56	10.69 \pm 5.02	0.57 \pm 0.12	0.43 \pm 0.12
	VLM-RL	14.38 \pm 1.53	0.51 \pm 0.08	138.08 \pm 16.68	10.09 \pm 11.93	0.40 \pm 0.00	0.10 \pm 0.10
	DriveVLM-RL	14.13 \pm 1.07	0.45 \pm 0.01	174.69 \pm 8.96	0.29 \pm 0.43	0.60 \pm 0.00	0.17 \pm 0.06
Town 3	ChatScene-SAC	10.47 \pm 1.83	0.30 \pm 0.05	85.56 \pm 12.41	12.06 \pm 2.14	0.00 \pm 0.00	0.30 \pm 0.10
	VLM-RL	16.64 \pm 1.71	0.35 \pm 0.05	111.07 \pm 31.42	10.97 \pm 5.21	0.03 \pm 0.06	0.27 \pm 0.06
	DriveVLM-RL	16.78 \pm 0.70	0.39 \pm 0.07	156.57 \pm 29.20	12.06 \pm 1.04	0.17 \pm 0.06	0.27 \pm 0.15
Town 4	ChatScene-SAC	17.74 \pm 0.20	0.11 \pm 0.03	266.40 \pm 31.25	12.56 \pm 7.16	0.00 \pm 0.00	0.23 \pm 0.06
	VLM-RL	19.39 \pm 3.27	0.21 \pm 0.03	357.78 \pm 44.84	0.65 \pm 0.13	0.13 \pm 0.06	0.20 \pm 0.06
	DriveVLM-RL	19.48 \pm 1.32	0.27 \pm 0.17	438.40 \pm 284.27	3.64 \pm 1.87	0.07 \pm 0.06	0.20 \pm 0.10
Town 5	ChatScene-SAC	17.75 \pm 0.09	0.33 \pm 0.04	75.30 \pm 10.48	10.96 \pm 4.48	0.07 \pm 0.06	0.20 \pm 0.10
	VLM-RL	16.43 \pm 4.94	0.22 \pm 0.04	53.69 \pm 11.76	5.36 \pm 9.29	0.00 \pm 0.00	0.07 \pm 0.12
	DriveVLM-RL	15.37 \pm 1.84	0.31 \pm 0.06	86.59 \pm 14.23	10.65 \pm 3.42	0.10 \pm 0.05	0.20 \pm 0.10

[†]Town 2 is the training environment; all other towns are out-of-distribution.

Table 8: Performance under different traffic densities. Mean and standard deviation over 3 seeds. The best result in each column within a density is in **bold**.

Traffic Density	Model	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
Empty	ChatScene-SAC	17.81 \pm 0.04	0.51 \pm 0.03	149.14 \pm 8.62	0.00 \pm 0.00	0.60 \pm 0.00	0.00 \pm 0.00
	VLM-RL	17.35 \pm 0.80	0.51 \pm 0.07	189.46 \pm 14.20	0.00 \pm 0.00	0.83 \pm 0.12	0.00 \pm 0.00
	DriveVLM-RL	15.82 \pm 0.63	0.54 \pm 0.02	208.49 \pm 0.54	0.00 \pm 0.00	1.00 \pm 0.00	0.00 \pm 0.00
Regular	ChatScene-SAC	17.47 \pm 0.10	0.45 \pm 0.03	161.78 \pm 13.56	10.69 \pm 5.02	0.57 \pm 0.12	0.43 \pm 0.12
	VLM-RL	14.38 \pm 1.53	0.51 \pm 0.08	138.08 \pm 16.68	10.09 \pm 11.93	0.40 \pm 0.00	0.10 \pm 0.10
	DriveVLM-RL	14.13 \pm 1.07	0.45 \pm 0.01	174.69 \pm 8.96	0.29 \pm 0.43	0.60 \pm 0.00	0.17 \pm 0.06
Dense	ChatScene-SAC	17.24 \pm 0.30	0.44 \pm 0.04	131.22 \pm 14.37	13.58 \pm 1.23	0.40 \pm 0.10	0.50 \pm 0.10
	VLM-RL	12.00 \pm 0.85	0.32 \pm 0.04	106.93 \pm 11.99	7.38 \pm 1.94	0.40 \pm 0.10	0.37 \pm 0.12
	DriveVLM-RL	10.36 \pm 1.24	0.45 \pm 0.03	136.87 \pm 2.34	0.75 \pm 0.31	0.50 \pm 0.10	0.40 \pm 0.08

4.8.3. Transferability to On-Policy Learning

To assess whether DriveVLM-RL’s reward design transfers across RL paradigms, we implement PPO variants (Schulman et al., 2017) of all three methods and evaluate on the same test routes. Results are presented in Table 9 and training dynamics in Fig. 14. DriveVLM-RL-PPO achieves the strongest overall performance, with the highest route completion (RC = 0.75), total distance (TD = 203.32 m), success rate (SR = 0.58), and lowest collision count (AC = 0.02). Compared to VLM-RL-PPO, DriveVLM-RL-PPO completes 44% more routes (0.75 vs. 0.52) and achieves 45% higher SR (0.58 vs. 0.40) with substantially fewer collisions. The training curves confirm this advantage: Fig. 14(a) shows that DriveVLM-RL-PPO maintains a consistently lower collision rate (0.15–0.25) than both ChatScene-PPO (\approx 0.75–0.85) and VLM-RL-PPO (\approx 0.35–0.45) throughout training, while Figs. 14(e)–(f) show progressive improvement in route completion and distance that neither baseline achieves. Notably, the collision interval (Fig. 14(b)) for DriveVLM-RL-PPO is substantially higher than baselines, indicating longer safe driving stretches between incidents. These results confirm that the dual-pathway semantic reward structure is algorithm-agnostic and transfers effectively from off-policy SAC to on-policy PPO training.

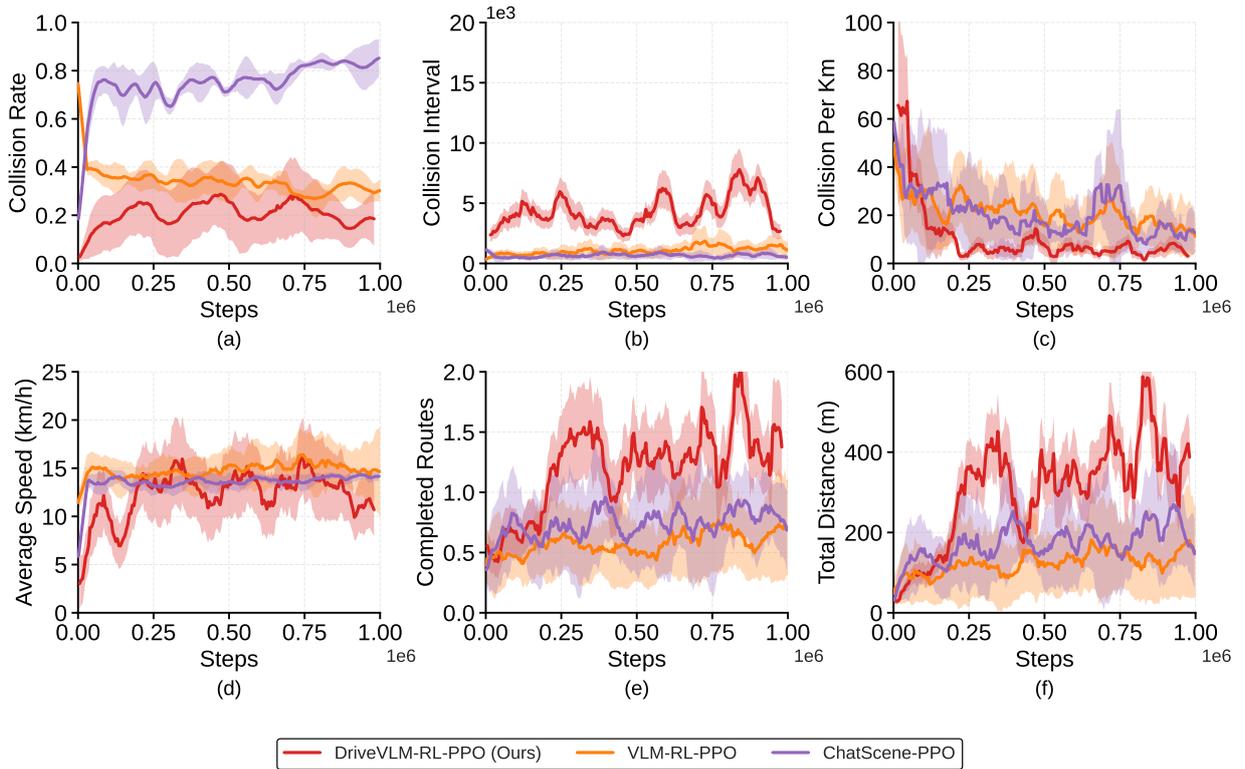


Figure 14: Training curves for PPO-based methods. (a) Collision rate; (b) Collision interval; (c) Collision per km; (d) Average speed; (e) Completed routes; (f) Total distance. DriveVLM-RL-PPO (red) consistently achieves lower collision rates and higher navigation performance than ChatScene-PPO (purple) and VLM-RL-PPO (orange).

Table 9: Comparison of PPO-based methods during testing. Mean and standard deviation over 3 seeds. The best result in each column is in **bold**.

Model	AS \uparrow	RC \uparrow	TD \uparrow	CS \downarrow	SR \uparrow	AC \downarrow
ChatScene-PPO	14.45 \pm 0.04	0.37 \pm 0.07	111.61 \pm 17.00	5.29 \pm 0.12	0.35 \pm 0.07	0.65 \pm 0.07
VLM-RL-PPO	16.65 \pm 1.22	0.52 \pm 0.10	121.99 \pm 3.95	4.26 \pm 7.38	0.40 \pm 0.10	0.03 \pm 0.06
DriveVLM-RL-PPO	15.55 \pm 0.78	0.75 \pm 0.06	203.32 \pm 20.33	4.78 \pm 0.48	0.58 \pm 0.06	0.02 \pm 0.00

5. Conclusion

This paper presented DriveVLM-RL, a neuroscience-inspired framework that integrates VLM into RL for safe and deployable autonomous driving. Motivated by the brain’s dual-pathway cognitive architecture, DriveVLM-RL decomposes semantic reward learning into a Static Pathway for continuous spatial safety assessment and a Dynamic Pathway for attention-gated, multi-frame semantic risk reasoning. A hierarchical reward synthesis mechanism fuses these signals with vehicle state information, while an asynchronous pipeline decouples expensive LVLM inference from environment interaction. Critically, all VLM components are used exclusively during training and completely removed at deployment, eliminating the latency constraints that plague existing VLM-for-control approaches.

Extensive experiments in CARLA demonstrate that DriveVLM-RL consistently outperforms 13 baseline methods across expert-designed, LLM-based, and VLM-based reward paradigms, achieving an 84% collision rate reduction over ChatScene-SAC and a 60% test success rate with collision severity reduced from 10.09 to 0.29 km/h. Under the extreme no-reward-after-collision setting, DriveVLM-RL maintains low collision rates throughout training, demonstrating that the policy internalizes predictive safety through semantic reasoning rather than penalty-driven avoidance. Cross-town generalization and PPO transferability experiments further confirm the robustness and algorithm-agnostic nature of the framework.

Several directions remain open for future work. First, performance in structurally dissimilar environments such as highway-style and multi-level road layouts remains limited, highlighting the need for richer semantic vocabularies beyond pedestrian-rich urban scenarios. Second, the current framework is validated in simulation; bridging the sim-to-real gap for deployment on physical vehicles (e.g., Sky-Drive (Huang et al., 2025c)) will require addressing sensor noise, domain shift in visual observations, and real-time safety constraints. Third, investigating online reward adaptation where the language goal vocabulary evolves during training could further improve long-tail robustness. Finally, extending the dual-pathway architecture to multi-agent settings represents a promising direction toward human-level generalization in autonomous driving.

Acknowledgment

This work was supported by the University of Wisconsin-Madison's Center for Connected and Automated Transportation (CCAT), a part of the larger CCAT consortium, a USDOT Region 5 University Transportation Center funded by the U.S. Department of Transportation, Award #69A3552348305. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein, and do not necessarily reflect the official views or policies of the sponsoring organization.

Appendix A. Static Pathway Proofs

The theoretical properties of the CLG-based static reward formulation follow from the structure of cosine similarity and real-valued arithmetic. Lemmas 1–2 extend the analysis in Huang et al. (2025b) to the specific CLG formulation defined in Definition 2.

Lemma 1 (Boundedness). *For any observation o_t and CLG pair (l_{pos}, l_{neg}) , the static reward is bounded: $R_{static}(o_t) \in [-1, 1]$.*

Proof 1. *For any two unit-normalized vectors $v_1, v_2 \in \mathbb{R}^d$, the Cauchy–Schwarz inequality gives:*

$$|v_1^\top v_2| \leq \|v_1\| \|v_2\| \quad (\text{A.1})$$

Since CLIP encoders produce ℓ_2 -normalized embeddings, $\|f_I(\cdot)\| = \|f_L(\cdot)\| = 1$, and thus:

$$\text{sim}(f_I(o_t), f_L(l)) = \frac{f_I(o_t)^\top f_L(l)}{\|f_I(o_t)\| \|f_L(l)\|} \in [-1, 1] \quad (\text{A.2})$$

Let $s^+ = \text{sim}(f_I(o_t^{BEV}), f_L(l_{pos})) \in [-1, 1]$ and $s^- = \text{sim}(f_I(o_t^{BEV}), f_L(l_{neg})) \in [-1, 1]$. Then:

$$R_{static}(o_t) = \alpha \cdot s^+ - \beta \cdot s^- \quad (\text{A.3})$$

The upper bound is achieved when $s^+ = 1$ and $s^- = -1$:

$$R_{static}(o_t) \leq \alpha \cdot 1 - \beta \cdot (-1) = \alpha + \beta = 1 \quad (\text{A.4})$$

The lower bound is achieved when $s^+ = -1$ and $s^- = 1$:

$$R_{static}(o_t) \geq \alpha \cdot (-1) - \beta \cdot 1 = -(\alpha + \beta) = -1 \quad (\text{A.5})$$

Therefore $R_{static}(o_t) \in [-1, 1]$.

Lemma 2 (Discriminability). *The CLG formulation provides strictly greater reward discrimination than single-goal formulations. Specifically, for observations o_1, o_2 where $\text{sim}(f_I(o_1), f_L(l_{pos})) = \text{sim}(f_I(o_2), f_L(l_{pos}))$ but $\text{sim}(f_I(o_1), f_L(l_{neg})) \neq \text{sim}(f_I(o_2), f_L(l_{neg}))$, we have $R_{static}(o_1) \neq R_{static}(o_2)$, even when single-goal similarity fails to distinguish the two states.*

Proof 2. *Let $s_i^+ = \text{sim}(f_I(o_i), f_L(l_{pos}))$ and $s_i^- = \text{sim}(f_I(o_i), f_L(l_{neg}))$ for $i \in \{1, 2\}$.*

By hypothesis, $s_1^+ = s_2^+$ and $s_1^- \neq s_2^-$.

A single-goal reward $r_i = \text{sim}(f_I(o_i), f_L(l_{pos})) = s_i^+$ satisfies $r_1 = r_2$, so it cannot distinguish o_1 from o_2 .

For the CLG reward:

$$R_{static}(o_1) - R_{static}(o_2) = (\alpha s_1^+ - \beta s_1^-) - (\alpha s_2^+ - \beta s_2^-) \quad (\text{A.6})$$

$$= \alpha(s_1^+ - s_2^+) - \beta(s_1^- - s_2^-) \quad (\text{A.7})$$

$$= \alpha \cdot 0 - \beta(s_1^- - s_2^-) \quad (\text{A.8})$$

$$= -\beta(s_1^- - s_2^-) \quad (\text{A.9})$$

Since $\beta > 0$ and $s_1^- \neq s_2^-$ by hypothesis, we conclude $R_{static}(o_1) \neq R_{static}(o_2)$.

Theorem 1 (Reward-Induced State Ordering). *Let \mathcal{S} be the state space and define the binary relation \succeq on \mathcal{S} such that $s_1 \succeq s_2$ if and only if $R_{static}(s_1) \geq R_{static}(s_2)$. Then \succeq is a total preorder (reflexive, transitive, and total), inducing a consistent preference ranking over states aligned with the semantic safety specification (l_{pos}, l_{neg}) .*

Proof 3. *We verify the three defining properties of a total preorder.*

(1) Reflexivity. *For any $s \in \mathcal{S}$:*

$$R_{static}(s) \geq R_{static}(s) \quad (\text{A.10})$$

holds trivially, so $s \succeq s$.

(2) Transitivity. *Suppose $s_1 \succeq s_2$ and $s_2 \succeq s_3$ for some $s_1, s_2, s_3 \in \mathcal{S}$. Then:*

$$R_{static}(s_1) \geq R_{static}(s_2) \geq R_{static}(s_3) \quad (\text{A.11})$$

By transitivity of \geq on \mathbb{R} , $R_{static}(s_1) \geq R_{static}(s_3)$, thus $s_1 \succeq s_3$.

(3) Totality. *For any $s_1, s_2 \in \mathcal{S}$, since $R_{static}(s_1), R_{static}(s_2) \in \mathbb{R}$ and \geq is a total order on \mathbb{R} :*

$$R_{static}(s_1) \geq R_{static}(s_2) \quad \text{or} \quad R_{static}(s_2) \geq R_{static}(s_1) \quad (\text{A.12})$$

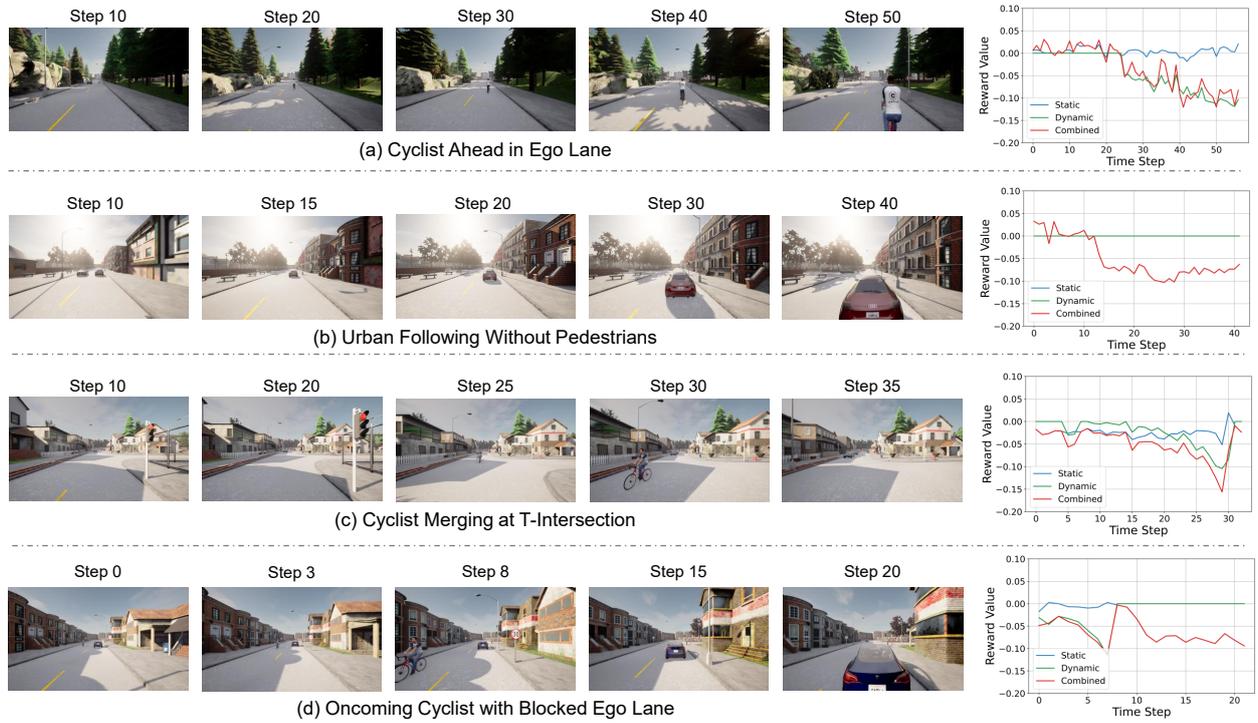


Figure B.15: Comparative reward curves and visual observations across diverse traffic scenarios. (a) A cyclist appears in the same lane ahead of the ego vehicle, requiring early deceleration and cautious following; (b) The ego vehicle follows a car in a structured urban environment with no pedestrians or dynamic obstacles; (c) At a T-junction, a cyclist enters the ego vehicle’s path from the right, demanding quick perception and safe negotiation; (d) The ego lane is blocked by a parked or slow vehicle, while a cyclist is approaching from the opposite direction in the oncoming lane.

Thus either $s_1 \succeq s_2$ or $s_2 \succeq s_1$ (or both when equality holds).

Semantic Alignment. The ordering \succeq reflects the safety specification (l_{pos}, l_{neg}) because R_{static} is monotonically increasing in $\text{sim}(f_I(o), f_L(l_{pos}))$ and monotonically decreasing in $\text{sim}(f_I(o), f_L(l_{neg}))$. Formally, for any $s_1 \succeq s_2$:

$$\alpha \cdot \text{sim}(f_I(o(s_1)), f_L(l_{pos})) - \beta \cdot \text{sim}(f_I(o(s_1)), f_L(l_{neg})) \geq \alpha \cdot \text{sim}(f_I(o(s_2)), f_L(l_{pos})) - \beta \cdot \text{sim}(f_I(o(s_2)), f_L(l_{neg})) \quad (\text{A.13})$$

This guarantees that s_1 is ranked no lower than s_2 precisely when s_1 is jointly more similar to the desired state l_{pos} and less similar to the undesired state l_{neg} , consistent with the semantic safety specification.

Therefore, \succeq is a total preorder inducing a semantically grounded preference ranking over \mathcal{S} .

Appendix B. Case Studies: Attention-Gated Reward Behavior Across Driving Scenarios

Fig. B.15 presents four representative driving scenarios that illustrate the complementary behavior of the static and dynamic reward components across varying levels of semantic complexity.

(a) Cyclist Ahead in Ego Lane. A cyclist gradually approaches the ego vehicle’s lane over 50 steps. The static reward (R_{static} , blue) remains relatively stable at a slightly negative level, reflecting persistent spatial proximity risk captured via BEV assessment. Once the cyclist enters the critical detection zone, the attentional gate activates ($g_t = 1$), triggering LVLMM inference and producing a sharply negative dynamic reward ($R_{dynamic}$, green). The combined reward ($R_{combined}$, red) consequently drops, providing a strong penalty signal that encourages the agent to decelerate and yield.

(b) Urban Following Without Pedestrians. In a structured urban scenario with no vulnerable road users present, the attentional gate remains inactive throughout ($g_t = 0$ for all t), and $R_{dynamic} = 0$ (green line flat at zero). The combined reward tracks the static reward exclusively, reflecting the designed fallback behavior: in the absence of safety-critical objects, the framework relies entirely on the Static Pathway for spatial safety assessment. This demonstrates the computational efficiency of the gating mechanism—no LVLMM inference is incurred in routine scenarios.

(c) Cyclist Merging at T-Intersection. A cyclist abruptly enters the ego vehicle’s path from the right at a T-intersection around step 20–25. The static reward declines gradually as the cyclist approaches, while the dynamic reward exhibits a sharp negative spike upon gate activation, capturing the semantic risk of the crossing maneuver (e.g., “A cyclist is merging into the ego lane from the right”). The combined reward reflects both the spatial hazard and the semantic context, providing richer guidance than either signal alone.

(d) Oncoming Cyclist with Blocked Ego Lane. From step 0, the ego lane is partially blocked by a parked vehicle while a cyclist approaches from the opposite direction. The static reward immediately registers a negative value due to the spatial obstruction visible in BEV. As the oncoming cyclist is detected, the dynamic reward drops sharply,

producing a strongly negative combined signal that discourages the agent from proceeding and encourages a cautious lane-change or stopping maneuver.

Across all four scenarios, the combined reward R_{combined} consistently provides more discriminative and semantically grounded signals than either pathway alone, validating the design rationale of the Hierarchical Reward Synthesis module described in Section 3.

Appendix C. Dynamic Pathway Proofs

Appendix C.1. Proof of Lemma 3 (Computational Efficiency)

Lemma 3 (Computational Efficiency). Let $p = P(g_t = 1)$ be the probability of gate activation, and let T_{LVLM} , T_{det} denote the inference time of the LVLM and detection model respectively. The expected per-frame computation time of the Dynamic Pathway is $T_{\text{det}} + p \cdot T_{\text{LVLM}}$, compared to T_{LVLM} for ungated approaches. When $p \ll 1$ and $T_{\text{det}} \ll T_{\text{LVLM}}$, this yields relative computational savings of approximately $(1 - p) \times 100\%$ compared to ungated LVLM inference.

Proof 4. For each frame, the Dynamic Pathway executes: (1) detection model $D(\cdot)$, always at cost T_{det} ; and (2) LVLM $F_{\text{LVLM}}(\cdot)$, only when $g_t = 1$, at cost T_{LVLM} .

By linearity of expectation:

$$\mathbb{E}[T_{\text{gated}}] = T_{\text{det}} + p \cdot T_{\text{LVLM}} \quad (\text{C.1})$$

The ungated baseline always runs the LVLM:

$$T_{\text{ungated}} = T_{\text{LVLM}} \quad (\text{C.2})$$

The relative savings are:

$$\text{Savings} = \frac{T_{\text{ungated}} - \mathbb{E}[T_{\text{gated}}]}{T_{\text{ungated}}} = 1 - p - \frac{T_{\text{det}}}{T_{\text{LVLM}}} \approx 1 - p \quad (\text{C.3})$$

where the approximation holds when $T_{\text{det}} \ll T_{\text{LVLM}}$. In our experiments, $p \approx 0.2\text{--}0.3$, yielding savings of approximately 70–80%.

Appendix C.2. Proof of Theorem 2 (Information Preservation under Gating)

Theorem 2 (Information Preservation under Gating). Let $\mathcal{S}_{\text{critical}} \subseteq \mathcal{S}$ denote the set of safety-critical states, and let μ be a distribution over $\mathcal{S}_{\text{critical}}$. Assume:

- (i) The detection model $D(\cdot)$ achieves recall $\rho = P(g(s) = 1 \mid s \in \mathcal{S}_{\text{critical}})$ on $\mathcal{S}_{\text{critical}}$;
- (ii) $R_{\text{LVLM}}(s) \geq 0$ for all $s \in \mathcal{S}_{\text{critical}}$;
- (iii) Detection misses are not systematically correlated with reward magnitude, i.e., $\mathbb{E}_{\mu}[R_{\text{LVLM}} \mid g = 1] \geq \mathbb{E}_{\mu}[R_{\text{LVLM}}]$.

Let $g(s) \in \{0, 1\}$ denote the gating indicator. Then:

$$\mathbb{E}_{s \sim \mu}[g(s) \cdot R_{\text{LVLM}}(s)] \geq \rho \cdot \mathbb{E}_{s \sim \mu}[R_{\text{LVLM}}(s)] \quad (\text{C.4})$$

Proof 5. By the law of total expectation, conditioning on $g(s)$:

$$\begin{aligned} \mathbb{E}_{s \sim \mu}[g(s) \cdot R_{\text{LVLM}}(s)] &= P(g = 1) \cdot \mathbb{E}_{\mu}[R_{\text{LVLM}} \mid g = 1] + P(g = 0) \cdot 0 \\ &= \rho \cdot \mathbb{E}_{\mu}[R_{\text{LVLM}} \mid g = 1] \end{aligned} \quad (\text{C.5})$$

where the second term vanishes because $g(s) = 0$ implies $g(s) \cdot R_{\text{LVLM}}(s) = 0$.

By assumption (iii):

$$\mathbb{E}_{\mu}[R_{\text{LVLM}} \mid g = 1] \geq \mathbb{E}_{s \sim \mu}[R_{\text{LVLM}}(s)] \quad (\text{C.6})$$

Substituting Eq. (C.6) into Eq. (C.5):

$$\mathbb{E}_{s \sim \mu}[g(s) \cdot R_{\text{LVLM}}(s)] = \rho \cdot \mathbb{E}_{\mu}[R_{\text{LVLM}} \mid g = 1] \geq \rho \cdot \mathbb{E}_{s \sim \mu}[R_{\text{LVLM}}(s)] \quad (\text{C.7})$$

which establishes the claimed inequality.

Remark 1. Assumption (iii) is mild in practice: it states that the detection model does not systematically fail on the highest-risk frames. In our implementation with YOLOv8 (Jocher et al., 2023) achieving $\rho \approx 0.95$, missed detections are primarily low-confidence borderline cases rather than high-severity scenarios, supporting the validity of this assumption empirically. Assumptions (i)–(ii) are standard; (ii) holds because R_{LVLM} is defined via cosine similarity against a positive goal l_{pos} , which is non-negative in the normalized CLIP embedding space when safety-critical states are present.

Appendix D. Hierarchical Reward Synthesis Proofs

Theorem 3 (Policy Improvement Guarantee). Let π_k denote the policy at iteration k , and π_{k+1} the updated policy obtained under the hierarchical reward R_{final} . Under standard assumptions of soft actor–critic learning, including bounded rewards, sufficient exploration, and stable function approximation, the policy update satisfies:

$$J(\pi_{k+1}) \geq J(\pi_k) - \epsilon_k \quad (\text{D.1})$$

where $J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^T \gamma^t R_{final}(o_t) \right]$, and ϵ_k denotes a bounded approximation error that diminishes as training progresses.

Proof 6. We verify that the hierarchical reward R_{final} satisfies all conditions required for the SAC policy improvement theorem (Haarnoja et al., 2018) to apply.

Step 1: Boundedness of R_{final} . We establish the reward bound by tracing through the hierarchical construction.

First, by Lemma 1, the static reward satisfies $R_{static}(o_t) \in [-1, 1]$.

Second, for the dynamic reward (Definition 5), since $g_t \in \{0, 1\}$ and the bracketed term $\alpha \cdot \text{sim}(f_I(o_t^{cam}), f_L(l_{pos})) - \beta \cdot \text{sim}(f_I(o_t^{cam}), f_L(l_t^{dyn})) \in [-1, 1]$ by the same cosine similarity argument as Lemma 1, we have:

$$R_{dynamic}(o_t) = g_t \cdot [\alpha \cdot \text{sim}(f_I(o_t^{cam}), f_L(l_{pos})) - \beta \cdot \text{sim}(f_I(o_t^{cam}), f_L(l_t^{dyn}))] \in [-1, 1] \quad (\text{D.2})$$

with $R_{dynamic} = 0$ when $g_t = 0$.

Third, since $R_{combined} = R_{static} + R_{dynamic} \in [-2, 2]$, the clipping and normalization in Eq. (9) map this to $R_{norm}(o_t) \in [0, 1]$ by construction of the clip operator.

Fourth, by Corollary 1, each factor $f_{speed}, f_{center}, f_{angle}, f_{stability} \in [0, 1]$, so their product satisfies $R_{shaping}(o_t) \in [0, 1]$.

Therefore, the final reward is bounded:

$$R_{final}(o_t) \in [R_{penalty}, 1], \quad |R_{final}(o_t)| \leq R_{max} \triangleq \max(|R_{penalty}|, 1) \quad (\text{D.3})$$

Step 2: Compatibility with SAC Policy Improvement. The SAC algorithm (Haarnoja et al., 2018) optimizes the maximum-entropy objective:

$$J(\pi_\phi) = \mathbb{E}_{\pi_\phi} \left[\sum_{t=0}^T \gamma^t (R(o_t, a_t) + \lambda \mathcal{H}(\pi_\phi(\cdot | o_t))) \right] \quad (\text{D.4})$$

where $\lambda > 0$ is the entropy regularization coefficient. The SAC policy improvement theorem guarantees that each policy update satisfies $J(\pi_{k+1}) \geq J(\pi_k) - \epsilon_k$ provided that: (a) the reward function is bounded, and (b) the policy and Q -function lie within a sufficiently expressive function approximation class.

Condition (a) is satisfied by Eq. (D.3). Condition (b) is a standard assumption on the neural network architecture, which we adopt here.

Step 3: Approximation Error Characterization. In practice, neural function approximation introduces estimation error. Let \mathcal{F} denote the function class of the critic network with pseudo-dimension $\text{Pdim}(\mathcal{F})$. Following standard analyses in approximate dynamic programming (Farahmand et al., 2010), the per-iteration approximation error can be bounded as:

$$\epsilon_k = \mathcal{O} \left(\sqrt{\frac{\text{Pdim}(\mathcal{F})}{N_k}} \right) + \epsilon_{approx} \quad (\text{D.5})$$

where N_k is the number of transitions sampled at iteration k and ϵ_{approx} is the irreducible approximation error of the critic class. As training proceeds and $N_k \rightarrow \infty$, the first term vanishes, leaving only the approximation bias ϵ_{approx} , which is bounded by the expressiveness of the chosen network architecture.

Step 4: Conclusion. Since R_{final} is bounded (Eq. D.3), preserves the POMDP structure (rewards depend only on observations o_t), and the SAC conditions are satisfied, the policy improvement bound in Eq. (D.1) holds for all k . The sequence $\{\pi_k\}$ therefore converges to a stable fixed point with bounded suboptimality ϵ_k under the hierarchical reward R_{final} .

Remark 2. The hierarchical structure of R_{final} does not interfere with convergence for three reasons: (i) all component rewards remain bounded (Corollary 1), satisfying the prerequisite of the SAC improvement theorem; (ii) the shaping reward $R_{shaping}$ is observation-dependent only, preserving the underlying POMDP structure and ensuring that the Bellman operator remains a contraction; and (iii) the asynchronous reward computation (Section 3) introduces bounded reward staleness controlled by N_{warmup} , which affects convergence speed but not the validity of the improvement bound, since the Learner Thread preferentially samples reward-annotated transitions as described in Section 3.5.2.

Appendix E. Training Procedure with Asynchronous Batch-Processing

Algorithm 1 DriveVLM-RL Training with Asynchronous Reward Synthesis

Input: Policy parameters ϕ , Q-function parameters θ , target parameters θ^- , replay buffer \mathcal{D} , batch size B , CLIP encoders f_I, f_L , generative LVLm F_{LVLm} , detection model D , language goals $(l_{\text{pos}}, l_{\text{neg}})$, safety-critical classes $\mathcal{C}_{\text{critical}}$, CLG weighting factors α, β with $\alpha + \beta = 1$, reward bounds $\theta_{\text{min}}, \theta_{\text{max}}$, maximum speed v_{max} , update interval Δ , warmup threshold N_{warmup} , entropy coefficient λ , target network smoothing factor τ

- 1: **Precompute language embeddings:**
- 2: $\mathbf{v}_{\text{pos}} \leftarrow f_L(l_{\text{pos}}), \mathbf{v}_{\text{neg}} \leftarrow f_L(l_{\text{neg}})$
- 3: $N_{\text{ready}} \leftarrow 0$ ▷ Counter for reward-annotated transitions
- 4: **for** $t = 1, 2, \dots, T$ **do**
- 5: **// Interaction Thread**
- 6: Observe $o_t = (o_t^{\text{BEV}}, o_t^{\text{cam}})$ from environment
- 7: Select action $a_t \sim \pi_\phi(\cdot | o_t)$, execute in environment, observe o_{t+1}
- 8: Store transition $(o_t, a_t, o_{t+1}, r_t \leftarrow \text{NaN}, \text{ready} \leftarrow 0)$ in \mathcal{D}
- 9: **if** $t \bmod \Delta = 0$ **then**
- 10: **// Reward Worker Thread**
- 11: Sample mini-batch $\{(o_i, a_i, o_{i+1})\}_{i=1}^B$ from \mathcal{D} where $\text{ready} = 0$
- 12: **for** each transition i in mini-batch **do**
- 13: **— Static Pathway —**
- 14: $\mathbf{v}_i^{\text{BEV}} \leftarrow f_I(o_i^{\text{BEV}})$
- 15: $R_{\text{static}} \leftarrow \alpha \cdot \text{sim}(\mathbf{v}_i^{\text{BEV}}, \mathbf{v}_{\text{pos}}) - \beta \cdot \text{sim}(\mathbf{v}_i^{\text{BEV}}, \mathbf{v}_{\text{neg}})$
- 16: **— Dynamic Pathway —**
- 17: $\mathcal{O}_i \leftarrow D(o_i^{\text{cam}})$
- 18: **if** $\exists o \in \mathcal{O}_i$ s.t. $\text{cls}(o) \in \mathcal{C}_{\text{critical}}$ **then**
- 19: Construct temporal window $\mathcal{W}_i = \{o_{i-K}^{\text{cam}}, \dots, o_i^{\text{cam}}\}$
- 20: $l_i^{\text{dyn}} \leftarrow F_{\text{LVLm}}(\mathcal{W}_i, \mathcal{O}_i)$ ▷ Generate risk description
- 21: $\mathbf{v}_i^{\text{cam}} \leftarrow f_I(o_i^{\text{cam}})$
- 22: $R_{\text{dynamic}} \leftarrow \alpha \cdot \text{sim}(\mathbf{v}_i^{\text{cam}}, \mathbf{v}_{\text{pos}}) - \beta \cdot \text{sim}(\mathbf{v}_i^{\text{cam}}, f_L(l_i^{\text{dyn}}))$
- 23: **else**
- 24: $R_{\text{dynamic}} \leftarrow 0$
- 25: **end if**
- 26: **— Hierarchical Reward Synthesis —**
- 27: $R_{\text{combined}} \leftarrow R_{\text{static}} + R_{\text{dynamic}}$
- 28: $R_{\text{norm}} \leftarrow \frac{\text{clip}(R_{\text{combined}}, \theta_{\text{min}}, \theta_{\text{max}}) - \theta_{\text{min}}}{\theta_{\text{max}} - \theta_{\text{min}}}$
- 29: $v_{\text{actual}} \leftarrow \text{speed}(o_i), v_{\text{desired}} \leftarrow R_{\text{norm}} \cdot v_{\text{max}}$
- 30: $f_{\text{speed}} \leftarrow \max\left(0, 1 - \frac{|v_{\text{actual}} - v_{\text{desired}}|}{v_{\text{max}}}\right)$ ▷ Clipped to $[0, 1]$
- 31: $R_{\text{shaping}} \leftarrow f_{\text{speed}} \cdot f_{\text{center}}(o_i) \cdot f_{\text{angle}}(o_i) \cdot f_{\text{stability}}(o_i)$
- 32: $R_{\text{final}} \leftarrow \begin{cases} R_{\text{penalty}}, & \text{if collision at step } i \\ R_{\text{shaping}}, & \text{otherwise} \end{cases}$
- 33: Update $r_i \leftarrow R_{\text{final}}, \text{ready} \leftarrow 1$ in \mathcal{D}
- 34: $N_{\text{ready}} \leftarrow N_{\text{ready}} + 1$
- 35: **end for**
- 36: **end if**
- 37: **// Learner Thread**
- 38: **if** $N_{\text{ready}} \geq N_{\text{warmup}}$ **then** ▷ Wait until sufficient annotated data
- 39: Sample mini-batch from \mathcal{D} with $\text{ready} = 1$
- 40: Update critic: minimize $J'_Q(\theta)$ from Eq. (15)
- 41: Update actor: maximize SAC objective $J(\pi_\phi)$ from Eq. (13)
- 42: Soft update target networks: $\theta^- \leftarrow (1 - \tau)\theta^- + \tau\theta$
- 43: **end if**
- 44: **end for**
- 45: **return** learned policy π_ϕ

References

- Abouelazm, A., Michel, J., Zöllner, J.M., 2024. A review of reward functions for reinforcement learning in the context of autonomous driving, in: 2024 IEEE Intelligent Vehicles Symposium (IV), IEEE. pp. 156–163.
- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al., 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774 .
- Aradi, S., 2020. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems* 23, 740–759.
- Baumli, K., Baveja, S., Behbahani, F., Chan, H., Comanici, G., Flennerhag, S., Gazeau, M., Holsheimer, K., Horgan, D., Laskin, M., et al., 2023. Vision-language models as a source of rewards. arXiv preprint arXiv:2312.09187 .
- Cao, Z., Xu, S., Jiao, X., Peng, H., Yang, D., 2022. Trustworthy safety improvement for autonomous driving using reinforcement learning. *Transportation research part C: emerging technologies* 138, 103656.
- CarNewsChina, 2025. Baidu’s apollo go robotaxi leads global autonomous driving with 17m+ orders. URL: <https://carnewschina.com/2025/11/13/baidus-apollo-go-robotaxi-leads-global-autonomous-driving-with-17m-orders-target>. carNewsChina article; accessed 2025.
- Chen, J., Li, S.E., Tomizuka, M., 2021. Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems* 23, 5068–5078.
- Corbetta, M., Shulman, G.L., 2002. Control of goal-directed and stimulus-driven attention in the brain. *Nature reviews neuroscience* 3, 201–215.
- Cui, C., Ma, Y., Cao, X., Ye, W., Zhou, Y., Liang, K., Chen, J., Lu, J., Yang, Z., Liao, K.D., et al., 2024. A survey on multimodal large language models for autonomous driving, in: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 958–979.
- De Haan, P., Jayaraman, D., Levine, S., 2019. Causal confusion in imitation learning. *Advances in neural information processing systems* 32.
- Desimone, R., Duncan, J., et al., 1995. Neural mechanisms of selective visual attention. *Annual review of neuroscience* 18, 193–222.
- Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. Carla: An open urban driving simulator, in: *Conference on robot learning*, PMLR. pp. 1–16.
- Farahmand, A.m., Szepesvári, C., Munos, R., 2010. Error propagation for approximate policy and value iteration. *Advances in neural information processing systems* 23.
- Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., Liu, H.X., 2023. Dense reinforcement learning for safety validation of autonomous vehicles. *Nature* 615, 620–627.
- García, J., Fernández, F., 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1437–1480.
- Goodale, M.A., Milner, A.D., 1992. Separate visual pathways for perception and action. *Trends in neurosciences* 15, 20–25.
- Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: *International conference on machine learning*, PMLR. pp. 1861–1870.
- Han, W., Guo, D., Xu, C.Z., Shen, J., 2025. Dme-driver: Integrating human decision logic and 3d scene perception in autonomous driving, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3347–3355.
- Han, X., Yang, Q., Chen, X., Cai, Z., Chu, X., Zhu, M., 2024. Autoreward: Closed-loop reward design with large language models for autonomous driving. *IEEE Transactions on Intelligent Vehicles* .
- HAZRA, R., Sygkounas, A., Persson, A., Loufi, A., Dos Martires, P.Z., 2025. Revolve: Reward evolution with large language models using human feedback, in: *The Thirteenth International Conference on Learning Representations*.
- He, X., Huang, W., Lv, C., 2024. Trustworthy autonomous driving via defense-aware robust reinforcement learning against worst-case observational perturbations. *Transportation Research Part C: Emerging Technologies* 163, 104632.
- Huang, Z., Sheng, Z., Chen, S., 2025a. Pe-rlhf: Reinforcement learning with human feedback and physics knowledge for safe and trustworthy autonomous driving. *Transportation Research Part C: Emerging Technologies* 179, 105262.
- Huang, Z., Sheng, Z., Ma, C., Chen, S., 2024. Human as ai mentor: Enhanced human-in-the-loop reinforcement learning for safe and efficient autonomous driving. *Communications in Transportation Research* 4, 100127.
- Huang, Z., Sheng, Z., Qu, Y., You, J., Chen, S., 2025b. Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving. *Transportation Research Part C: Emerging Technologies* 180, 105321.
- Huang, Z., Sheng, Z., Wan, Z., Qu, Y., Luo, Y., Wang, B., Li, P., Chen, Y.J., Chen, J., Long, K., et al., 2025c. Sky-drive: a distributed multiagent simulation platform for human-ai collaborative and socially aware future transportation. *Journal of Intelligent and Connected Vehicles* 8, 9210070–1.
- Ilharco, G., Wortsman, M., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., et al., 2021. Openclip. Zenodo .
- Jiang, S., Huang, Z., Qian, K., Luo, Z., Zhu, T., Zhong, Y., Tang, Y., Kong, M., Wang, Y., Jiao, S., et al., 2025. A survey on vision-language-action models for autonomous driving, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4524–4536.
- Jiao, S., Qian, K., Ye, H., Zhong, Y., Luo, Z., Jiang, S., Huang, Z., Fang, Y., Miao, J., Fu, Z., et al., 2025. Evadrive: Evolutionary adversarial policy optimization for end-to-end autonomous driving. arXiv preprint arXiv:2508.09158 .
- Jocher, G., Chaurasia, A., Qiu, J., 2023. Yolo by ultralytics. <https://github.com/ultralytics/ultralytics>. YOLOv8 implementation.
- Kaelbling, L.P., Littman, M.L., Cassandra, A.R., 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 99–134.
- Knox, W.B., Allievi, A., Banzhaf, H., Schmitt, F., Stone, P., 2023. Reward (mis) design for autonomous driving. *Artificial Intelligence* 316, 103829.
- Kolodny, L., 2025. Waymo crosses 450,000 weekly paid rides as alphabet robotaxi unit widens lead. <https://www.cnbc.com>. CNBC article reporting Waymo’s weekly paid ride milestone; accessed 2025.
- Lee, T., Wagenmaker, A., Pertsch, K., Liang, P., Levine, S., Finn, C., 2026. Roboreward: General-purpose vision-language reward models for robotics. arXiv preprint arXiv:2601.00675 .
- Lu, R., Shao, Z., Ding, Y., Chen, R., Wu, D., Su, H., Yang, T., Zhang, F., Wang, J., Shi, Y., et al., 2025. Discovery of the reward function for embodied reinforcement learning agents. *Nature Communications* 16, 11064.
- Luo, Z., Qian, K., Wang, J., Luo, Y., Miao, J., Fu, Z., Wang, Y., Jiang, S., Huang, Z., Hu, Y., et al., 2025. Mtdrive: Memory-tool synergistic reasoning for robust autonomous driving in corner cases. arXiv preprint arXiv:2509.20843 .
- Meng, X., Zhang, Y., Huang, Z., Lu, Z., Ji, Z., Yin, Y., Zhang, H., Jiang, G., Lin, Y., Chen, L., et al., 2025. Is your vlm for autonomous driving safety-ready? a comprehensive benchmark for evaluating external and in-cabin risks. arXiv preprint arXiv:2511.14592 .
- Miller, E.K., Cohen, J.D., 2001. An integrative theory of prefrontal cortex function. *Annual review of neuroscience* 24, 167–202.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. *nature* 518, 529–533.
- Pang, H., Wang, Z., Li, G., 2026. Large language model guided deep reinforcement learning for safe autonomous vehicle decision making. *Transportation Research Part C: Emerging Technologies* 184, 105511.

- Qian, K., Jiang, S., Zhong, Y., Luo, Z., Huang, Z., Zhu, T., Jiang, K., Yang, M., Fu, Z., Miao, J., et al., 2025. Agentthink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. arXiv preprint arXiv:2505.15298 .
- Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR. pp. 8748–8763.
- Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N., 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of machine learning research* 22, 1–8.
- Rangel, A., Camerer, C., Montague, P.R., 2008. A framework for studying the neurobiology of value-based decision making. *Nature reviews neuroscience* 9, 545–556.
- Rocamonde, J., Montesinos, V., Nava, E., Perez, E., Lindner, D., 2024. Vision-language models are zero-shot reward models for reinforcement learning, in: *NeurIPS 2023 Foundation Models for Decision Making Workshop*.
- Ross, S., Gordon, G., Bagnell, D., 2011. A reduction of imitation learning and structured prediction to no-regret online learning, in: *Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*. pp. 627–635.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 .
- Sheng, Z., Huang, Z., Qu, Y., Leng, Y., Bhavanam, S., Chen, S., 2026. Curriculum: Towards safe autonomous driving via personalized safety-critical curriculum learning with vision-language models. *Transportation Research Part C: Emerging Technologies* 185, 105549.
- Sontakke, S., Zhang, J., Arnold, S., Pertsch, K., Biyik, E., Sadigh, D., Finn, C., Itti, L., 2023. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems* 36, 55681–55693.
- Sutton, R.S., Barto, A.G., et al., 1998. *Reinforcement learning: An introduction*. volume 1. MIT press Cambridge.
- Tesla, I., 2025. Tesla fsd version 14: Next-generation autonomous driving. <https://www.tesla.com/autopilot>. Accessed: 2025.
- Tian, X., Gu, J., Li, B., Liu, Y., Wang, Y., Zhao, Z., Zhan, K., Jia, P., Lang, X., Zhao, H., 2025. Drivevlm: The convergence of autonomous driving and large vision-language models, in: *Conference on Robot Learning, PMLR*. pp. 4698–4726.
- Wang, L., Liu, J., Shao, H., Wang, W., Chen, R., Liu, Y., Waslander, S.L., 2023. Efficient reinforcement learning for autonomous driving with parameterized skills and priors, in: *Robotics: Science and Systems*.
- Wasif, D., Moore, T.J., Reddy, C.K., Cho, J.H., 2025. Drivemind: A dual-vlm based reinforcement learning framework for autonomous driving. arXiv preprint arXiv:2506.00819 .
- Wu, J., Huang, C., Huang, H., Lv, C., Wang, Y., Wang, F.Y., 2024. Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey. *Transportation Research Part C: Emerging Technologies* 164, 104654.
- Xie, S., Kong, L., Dong, Y., Sima, C., Zhang, W., Chen, Q.A., Liu, Z., Pan, L., 2025. Are vlms ready for autonomous driving? an empirical study from the reliability, data and metric perspectives, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6585–6597.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al., 2025. Qwen3 technical report. arXiv preprint arXiv:2505.09388 .
- Ye, X., Tao, F., Mallik, A., Yaman, B., Ren, L., 2025. Lord: Large models based opposite reward design for autonomous driving, in: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE. pp. 5072–5081.
- You, J., Jiang, Z., Huang, Z., Shi, H., Gan, R., Wu, K., Cheng, X., Li, X., Ran, B., 2026. V2x-vlm: End-to-end v2x cooperative autonomous driving through large vision-language models. *Transportation Research Part C: Emerging Technologies* 183, 105457.
- Zhang, J., Xu, C., Li, B., 2024. Chatscene: Knowledge-enabled safety-critical scenario generation for autonomous vehicles, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15459–15469.
- Zhou, X., Liu, M., Yurtsever, E., Zagar, B.L., Zimmer, W., Cao, H., Knoll, A.C., 2024a. Vision language models in autonomous driving: A survey and outlook. *IEEE Transactions on Intelligent Vehicles* .
- Zhou, Z., Cai, T., Zhao, S.Z., Zhang, Y., Huang, Z., Zhou, B., Ma, J., 2025. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. arXiv preprint arXiv:2506.13757 .
- Zhou, Z., Zhang, J., Zhang, J., He, Y., Wang, B., Shi, T., Khamis, A., 2024b. Human-centric reward optimization for reinforcement learning-based automated driving using large language models. arXiv preprint arXiv:2405.04135 .