Enhancing Autonomous Driving with VLM-RL Framework

Veda Sahaja Bandi
Department of Data Science, University of Massachusetts Dartmouth
North Dartmouth, MA 02747

Abstract

This study presents the implementation and evaluation of a Vision-Language Model-Reinforcement Learning (VLM-RL) framework for autonomous driving, building upon existing research [1]. The framework integrates semantic reward signals generated through the Contrasting Language Goals (CLG) paradigm, guiding the agent to make safer decisions. We analyze how this reward system, which compares image observations with textual goals, influences the agent's behavior in terms of both safety and efficiency. Our comparative analysis between the VLM-RL framework and a traditional RL-only approach demonstrates that incorporating language understanding significantly improves key safety metrics, such as collision avoidance and lane-keeping precision, while introducing trade-offs in speed and efficiency. Experiments conducted in the CARLA simulated environment show that the VLM-RL framework reduces collisions by approximately 74% compared to RL-only approach and enhances lane-keeping, though the agent operates at a slower speed. These findings underscore the trade-off between safety and efficiency, highlighting the potential of semantic language goals in safety-critical applications.

1 Introduction

Autonomous Driving Systems have made remarkable strides in recent years. Yet, achieving human-level safety, reliability, and adaptability in real-world environments remains a critical challenge. Among the various approaches, reinforcement learning (RL) has emerged as a promising solution for enabling autonomous agents to navigate complex, dynamic settings. However, traditional RL models often suffer from slow convergence, poor generalization, and difficulty balancing safety, efficiency, and adherence to traffic rules. Designing effective reward functions in RL is particularly challenging, as they tend to be scenario-specific, require extensive engineering, and often fail to capture the nuanced behaviors required for safe driving. These limitations become even more apparent in unstructured scenarios such as those involving ambiguous pedestrian intent, where predicting erratic human behavior is difficult; dynamic roadblocks, which demand rapid adaptation to unexpected obstacles; and ethical dilemmas, where conventional models lack reasoning frameworks for morally complex decisions. These challenges stem from limited contextual understanding and the lack of explainability in traditional Autonomous Driving Systems (ADS). Recent advancements in Vision-Language Models (VLMs) offer a compelling solution to these issues by bridging the gap between visual perception and semantic understanding [2]. By enabling autonomous systems to interpret scenes in human-like terms (e.g., ”yield to pedestrians”), VLMs enhance safety, adaptability, and transparency. However, fully integrating VLMs into RL frameworks remains an open research problem.

In this study, we implement and evaluate a novel VLM-RL framework that incorporates OpenAI's CLIP model to generate semantic rewards based on the Contrastive Language Goals (CLG) paradigm [3]. The CLG approach contrasts positive and negative textual descriptions to guide the agent toward safer and more compliant driving behavior. Using the CARLA simulator, we compare this VLM-RL framework against a baseline traditional RL model to evaluate its impact on driving performance. Our results show that while the VLM-RL agent demonstrates significantly improved safety metrics, particularly in collision avoidance and lane keeping, it does so at the cost of reduced driving speed. This trade-off between safety and efficiency highlights the potential of incorporating semantic understanding in reinforcement learning for autonomous driving applications.

2 Methodology

This section outlines the VLM-RL framework implemented in this study, following the architecture proposed in the original work, "*VLM-RL: A Unified Vision Language Models and Reinforcement Learning Framework for Safe Autonomous Driving*" by Huang et al. (2024) [1]. The framework integrates semantic understanding from Vision-Language Models (VLMs) into a traditional Reinforcement Learning (RL) setup, enabling safer and more interpretable autonomous driving behavior. Figure 1 illustrates the architecture of the VLM-RL framework, highlighting the interaction between language-guided semantic goals, visual perception, reward computation, and agent training.

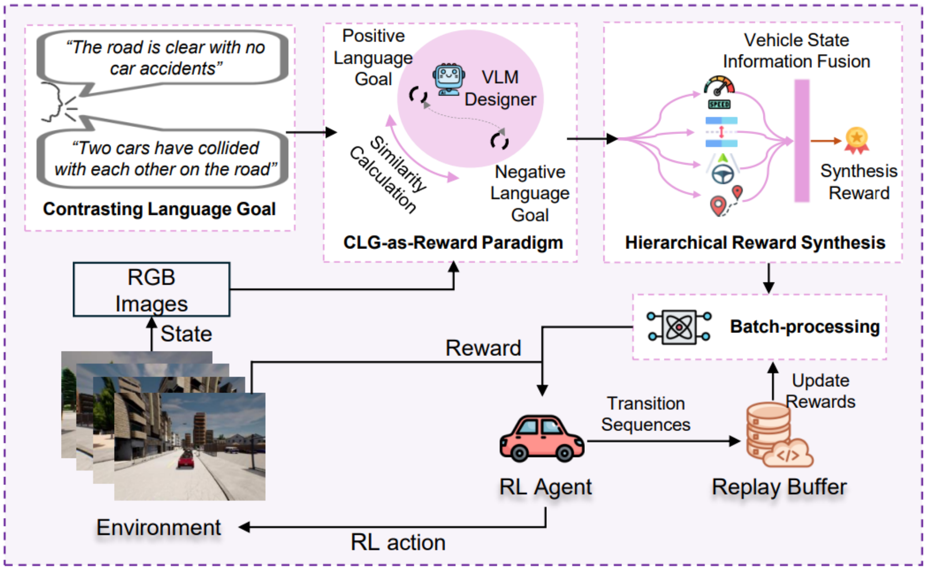

Figure 1: VLM-RL Framework Architecture

2.1 Framework Overview

The VLM-RL framework operates through four key components, which are executed iteratively as the agent interacts with the driving environment. These components collectively support the integration of semantic reasoning into the RL pipeline:

1. **Contrasting Language Goal:** This component defines pairs of positive and negative language descriptions that encapsulate desirable and undesirable driving behaviors, respectively. These textual descriptions serve as semantic references for the agent, guiding its behavior by providing clear examples of what constitutes safe or unsafe driving actions.

2. **CLG-based Semantic Reward Computation:** Utilizing a pre-trained Vision-Language Model (specifically, OpenAI's CLIP), this component computes the semantic similarity between the agent's current visual observation and the predefined CLG descriptions. The similarity scores are used to generate semantic rewards, encouraging the agent to align its behavior with positive descriptions while avoiding actions that resemble negative ones.

3. **Hierarchical Reward Synthesis:** To create a comprehensive reward signal, this component combines the semantic rewards derived from the CLG comparisons with traditional vehicle state information, such as speed and steering angle. This hierarchical approach ensures that the agent receives feedback that accounts for both high-level semantic alignment and low-level physical dynamics, promoting more stable and effective learning.

4. **Batch Processing:** To enhance computational efficiency, the framework employs a batch-processing technique. Instead of computing rewards in real-time for each observation, the system periodically processes batches of stored observations from the replay buffer. This approach reduces computational overhead and allows for more efficient policy training without compromising the quality of the reward signals.

These components, as depicted in Figure 1, work in concert to train an RL agent capable of aligning its behavior with human-like driving principles expressed through natural language.

2.2 CLG as a Reward Paradigm

The Contrasting Language Goals (CLG) approach introduces a novel reward paradigm that leverages natural language understanding to shape the reward function. The mathematical formulation of this paradigm is built around the computation of semantic similarities between visual observations and language descriptions [3]. The paradigm introduces language-based guidance into the reinforcement learning process by defining contrasting textual descriptions that represent desired and undesired driving outcomes:

Positive Description: ”The road is clear with no car accidents”

Negative Description: ”Two cars have collided with each other on the road”

These language descriptions serve as semantic anchors for the agent’s learning process, enabling it to ground its actions in human-understandable concepts. The use of contrasting goals helps the system capture task nuances more effectively by distinguishing between desirable and undesirable behaviors.

The Vision-Language Model (VLM) functions as a semantic interpreter, acting as a bridge between natural language descriptions and visual observations. Pre-trained on a large corpus of image-text pairs, the OpenAI CLIP maps both modalities into a shared, high-dimensional embedding space. This shared space is essential for comparing the semantic meaning of the observed scene with that of the language goals.

The mathematical formulation of the CLG reward function is as follows:

- At each timestep, the agent processes its current visual observation (an RGB image from the environment) through the CLIP visual encoder, while the positive and negative language goals are processed through the CLIP text encoder. The resulting embeddings are then used to compute semantic similarity, forming the basis for reward generation.
- The semantic similarity between the agent’s current observation and the CLG descriptions is computed using the cosine similarity metric. Given the embedding of the current observation $\phi(o_t)$, the positive language goal $\phi(g^+)$, and the negative language goal $\phi(g^-)$, the cosine similarities are calculated as follows:

$$\text{sim}^+ = \cos(\phi(o_t), \phi(g^+)) = \frac{\phi(o_t)^\intercal \cdot \phi(g^+)}{\|\phi(o_t)\| \cdot \|\phi(g^+)\|}$$

$$\text{sim}^- = \cos(\phi(o_t), \phi(g^-)) = \frac{\phi(o_t)^\intercal \cdot \phi(g^-)}{\|\phi(o_t)\| \cdot \|\phi(g^-)\|}$$

- The semantic reward r^{sem} is determined by the difference between the positive and negative similarities:

$$r^{\text{sem}} = \alpha \cdot \text{sim}^+ - \beta \cdot \text{sim}^-$$

  where $\alpha, \beta > 0$ are weighting factors satisfying $\alpha + \beta = 1$. If $\alpha > \beta$, the agent focuses more on achieving the positive goal, while if $\alpha < \beta$, the agent emphasizes steering clear of negative outcomes. For simplicity, we set $\alpha = \beta = 0.5$, i.e., the two goals are equally prioritized.

This integrated reward function enables the agent to learn a driving policy that optimizes for both semantic safety criteria and operational efficiency metrics.

2.3 Hierarchical Reward Synthesis and Batch Processing

In the Hierarchical Reward Synthesis step, the semantic reward r^{sem} is first normalized and then combined with low-level vehicle state information, such as speed and steering angle, to produce a final synthesized reward signal. This hybrid structure enables the agent to learn both high-level semantic alignment with human-like driving behaviors and low-level control stability. We adopt Proximal Policy Optimization (PPO) for policy training in the reinforcement learning algorithm, training both the policy and value networks to maximize the synthesized reward and encourage safer, more context-aware driving.

To reduce the computational overhead of CLIP-based semantic inference, we implement an efficient batch-processing strategy. During training, observations, state information, and rewards are stored in a replay buffer. At regular intervals, a batch of observations is sampled and processed through the CLIP encoder. The embeddings for the positive and negative language goals are computed once at the beginning of training and remain fixed throughout. Synthesized rewards are then computed and used to update the corresponding transitions in the buffer. PPO samples these updated transitions during policy optimization, enabling reward computation to run asynchronously without disrupting the learning loop.

2.4 Evaluation Metrics

To ensure a fair comparison between approaches, we evaluated both frameworks using the same set of metrics and scenarios. Each model was tested across multiple episodes with varying traffic conditions and route complexities. The evaluation metrics were designed to capture both safety aspects (collision rate, deviation from centerline) and efficiency aspects (task completion, distance traveled, average speed).

3 Experimental Setup

We conducted our experiments using the CARLA simulator, a high-fidelity platform tailored for autonomous driving research. This simulator offers a realistic environment equipped with collision detection, traffic simulation, and route planning capabilities, providing a comprehensive testing ground for evaluating our Vision-Language Reinforcement Learning (VLM-RL) framework.

3.1 Simulation Environment

The simulation environment was configured to ensure a challenging and realistic evaluation of autonomous driving systems. Comprehensive collision detection and handling were implemented to accurately identify and respond to interactions with obstacles, vehicles, and pedestrians. To simulate realistic traffic conditions, 20 vehicles were populated in the environment, operating in autopilot mode via CARLA's built-in traffic manager, following traffic rules, and performing basic collision avoidance. Episodes were terminated under the following conditions: (a) collision with any object, (b) deviation from the road center by more than 3 meters, or (c) the vehicle speed falling below 1 km/h for over 90 consecutive seconds, indicating the agent was stuck. For navigation, we dynamically generated routes using the A* algorithm, connecting randomly selected start and end points from among 101 predefined spawn points. New routes were continuously generated until the cumulative driving distance reached 3000 meters within an episode.

3.2 Agent Configuration

The VLM-RL agent was designed to process three primary input types to facilitate effective decision-making for navigation and control. First, bird's-eye view (BEV) semantic segmentation images (224×224 pixels) were used to capture the surrounding environment, including drivable areas, lane boundaries, and traffic participants, providing crucial spatial information. Second, ego state information comprising the current steering angle, throttle value, and vehicle speed, reflected the vehicle's dynamic state. Third, the navigation information included the next 15 waypoints along the planned route, represented as (x, y) coordinates relative to the vehicle's current position. This input helped the agent follow its desired trajectory. The agent's action space was defined as a continuous 2-dimensional space $[-1, 1]^2$. The first dimension controlled the steering angle, with -1 for maximum left, 0 for straight, and +1 for maximum right. The second dimension controlled throttle and brake, with positive values indicating throttle and negative values indicating braking.

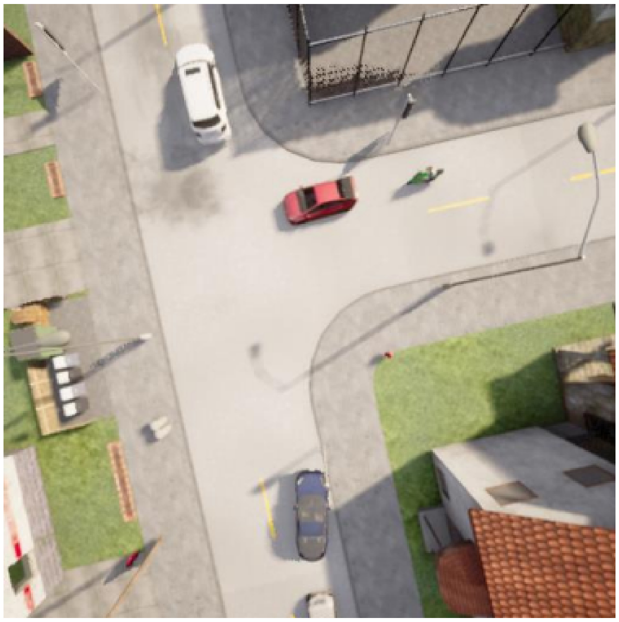

Figure 2: BEV Camera Image

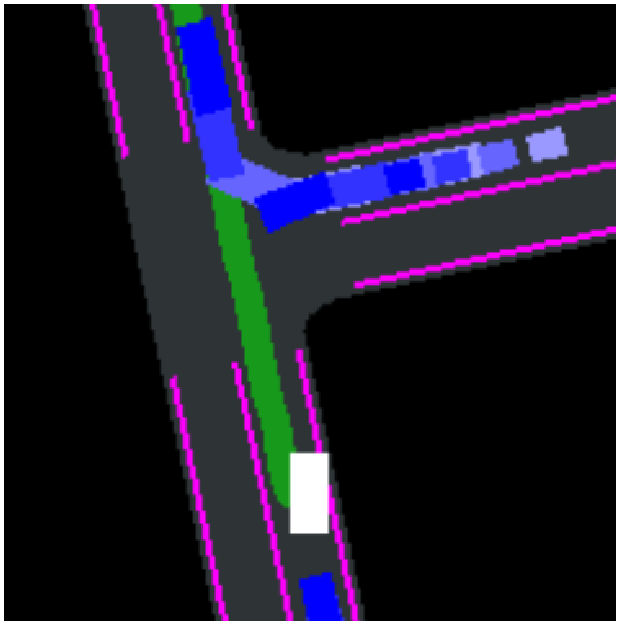

Figure 3: Semantic Image

3.3 Training Configuration

Training was conducted using the Stable-Baselines3 library, known for its reliable and modular implementation of modern RL algorithms. Due to computational constraints, we reduced the original training duration from 100,000 timesteps to 30,000 timesteps. The Proximal Policy Optimization (PPO) algorithm was selected for its efficiency and stability. Training took place in CARLA's Town 2 map, which features a typical European-style urban layout with diverse driving challenges. Generalization performance was tested in Towns 1, 3, 4, and 5. The VLM component employed OpenAI's CLIP model (ViT-B-32) to compute semantic rewards based on visual-language alignment. Consistent contrasting language descriptions were used throughout the training: ”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 scenario.

3.4 Hardware and Software Configuration

Our experiments were conducted on a high-performance setup designed for autonomous driving simulation and deep reinforcement learning. The hardware used included an NVIDIA RTX 3080 GPU with 8GB VRAM to handle the computational demands of processing visual inputs and training deep neural networks. The CARLA simulator provided a high-fidelity environment for autonomous driving tasks, featuring realistic physics, lighting, and traffic simulation. For the vision-language component, OpenAI's CLIP model (ViT-B-32) was utilized to process 224×224 pixel images and assess the alignment between observations and language descriptions.

4 Results

Our experimental evaluation compared the performance of the VLM-RL framework against a traditional RL-only baseline approach across 10 predefined routes. The results demonstrate the effectiveness of incorporating semantic rewards derived from the CLIP model into reinforcement learning for autonomous driving tasks.

During training, the VLM-RL framework showed a consistent increase in rewards over training timesteps, indicating successful learning of the desired behaviors as shown in Figure 4. However, the reward progression exhibited higher variability compared to the RL-only framework, reflecting the more complex reward landscape introduced by the semantic component.

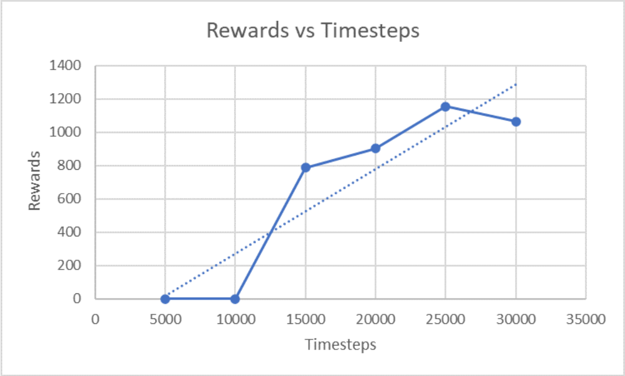

Figure 4: Rewards vs Timesteps for VLM-RL Framework

4.1 Comparative Performance

The VLM-RL framework demonstrated significant improvements in safety metrics compared to the RL-only approach, as shown in Table 1:

Metric	**VLM-RL Framework**	**RL-only Framework**
Task Completion Rate (%)	0.5 (50%)	0.2 (20%)
Collision Rate (CPM) (collisions/km)	3.13	12.07
Collision Interval (timesteps)	11678	390.5
Avg. Deviation (Center) (m)	0.195	0.515
Total Distance (m)	1305.08	1018.16
Total Reward	2484.72	1397.27
Avg. Collision Speed (m/s)	0.385	4.385
Avg. Speed (km/h)	1.3	14.55

Table 1: Performance Comparison of VLM-RL vs. RL-only Framework

The most striking difference between the two frameworks was in their approach to the safety-efficiency trade-off. The VLM-RL framework, guided by the contrastive language goals emphasizing collision avoidance, developed a highly cautious driving strategy. In RL-only framework, the agent focuses on maximizing reward through speed and destination achievement, often at the expense of safety, resulting in higher collision rates and speeds during collisions. This was evidenced by its significantly lower average speed (1.3 km/h compared to 14.55 km/h for the RL-only framework). While this extreme caution resulted in substantially improved safety metrics, a 74% reduction in collision rate and a 91% reduction in collision speed, it also led to slower progress toward destinations.

The VLM-RL framework also achieved a task completion rate of 50%, significantly higher than the 20% achieved by the RL-only approach. This improvement suggests that the semantic guidance provided by the CLG-based reward helps the agent maintain focus on the navigation task, even while prioritizing safety. The 62% reduction in average deviation from the lane center further indicates that the VLM-RL framework developed more precise driving behavior, staying closer to the intended path.

4.2 Control Parameter Analysis

Analysis of control parameters across episodes revealed distinct behavioral patterns between the two frameworks. The VLM-RL framework exhibited control signals with higher variability and less smoothness, indicating more frequent adjustments in response to perceived safety considerations. In contrast, the RL-only framework produced significantly smoother control signals, suggesting an optimization strategy focused primarily on efficient progress rather than safety margins as illustrated in Figures 5, 6. These behavioral differences were further validated by central deviation measurements, with the VLM-RL framework maintaining a much closer alignment to the lane center (0.195m average deviation) compared to the RL-only approach (0.515m average deviation). This demonstrates how the semantic understanding provided by the VLM component encourages more cautious driving behavior that prioritizes precise lane positioning over speed optimization.

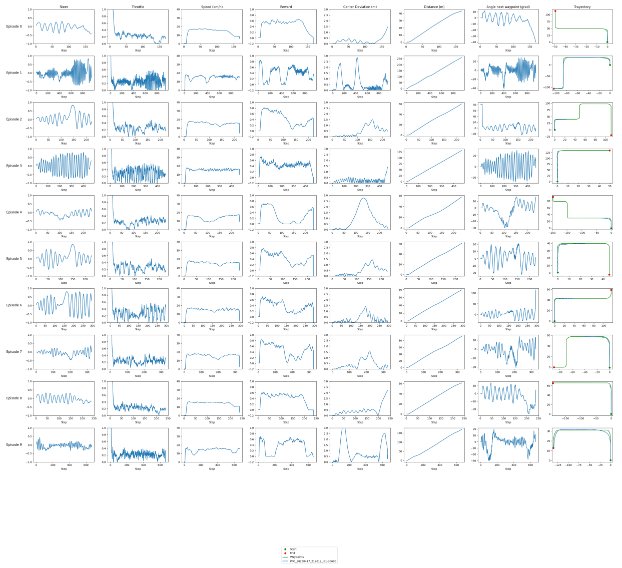

Figure 5: RL-only Framework

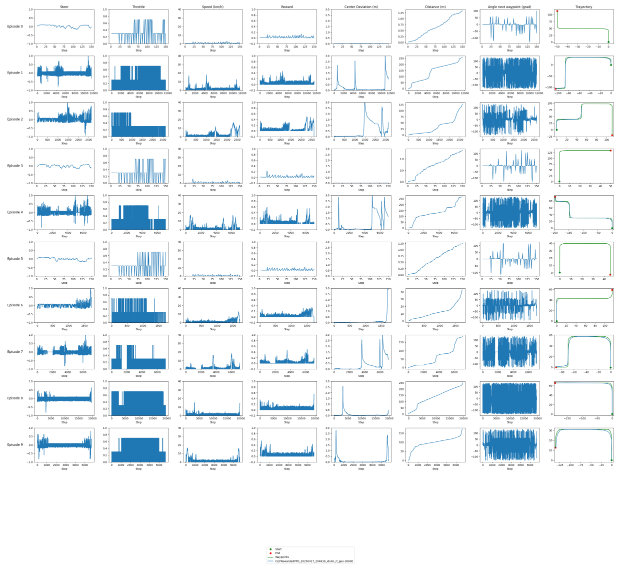

Figure 6: VLM-RL Framework

The results demonstrate that the VLM-RL framework exhibits notably better cross-environment generalization, maintaining higher success rates across all test environments. This suggests that the semantic understanding provided by the CLIP model enables the agent to better adapt to unfamiliar road layouts and driving conditions.

5 Conclusion

Our implementation and evaluation of the VLM-RL framework for autonomous driving yield several important insights. The incorporation of semantic rewards through Vision-Language Models significantly improves safety metrics, reducing collision rates by 74% and collision speeds by 91% compared to traditional RL approaches. However, the VLM-RL framework demonstrates a clear trade-off, prioritizing safety at the expense of speed (91% reduction in average speed), which may impact practical deployment considerations. The use of contrastive language goals successfully guides the agent toward safer driving behaviors, demonstrating the potential of incorporating human-interpretable language understanding into autonomous driving systems. The higher variability in control signals for the VLM-RL framework suggests that further refinement of the semantic reward function could potentially improve driving smoothness while maintaining safety advantages.

While the VLM-RL framework shows promising results in enhancing safety, future work should focus on finding a better balance between safety and efficiency. Potential improvements include refining the language goals to better capture the nuances of safe yet efficient driving, incorporating a broader range of driving scenarios in the training process, and exploring more sophisticated VLM architectures to enhance semantic understanding. This research demonstrates that integrating language understanding into autonomous driving systems offers a promising path toward creating safer, more human-aligned autonomous vehicles.

6 Appendix

The code can be accessed at: https://github.com/Veda0718/VLM-RL

References

[1] Z. Huang, Z. Sheng, Y. Qu, J. You, and S. Chen, "Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving," 2024.

[2] M. Elhenawy, H. I. Ashqar, A. Rakotonirainy, T. I. Alhadidi, A. Jaber, and M. A. Tami, "Vision-language models for autonomous driving: Clip-based dynamic scene understanding," 2025.

[3] X. Ye, F. Tao, A. Mallik, B. Yaman, and L. Ren, "Lord: Large models based opposite reward design for autonomous driving," 2024.

[4] X. Zhou, M. Liu, E. Yurtsever, B. L. Zagar, W. Zimmer, H. Cao, and A. C. Knoll, "Vision language models in autonomous driving: A survey and outlook," 2024.

[5] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner, "Vision-language models are zero-shot reward models for reinforcement learning," 2024.

[6] S. A. Sontakke, J. Zhang, S. M. R. Arnold, K. Pertsch, E. Bıyık, D. Sadigh, C. Finn, and L. Itti, "Roboclip: One demonstration is enough to learn robot policies," 2023.

[7] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson, "Rl-vlm-f: Reinforcement learning from vision language foundation model feedback," 2024.

[8] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," 2024.

[9] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," 2023.

[10] K. Baumli, S. Baveja, F. Behbahani, H. Chan, G. Comanici, S. Flennerhag, M. Gazeau, K. Holsheimer, D. Horgan, M. Laskin, C. Lyle, H. Masoom, K. McKinney, V. Mnih, A. Neitz, D. Nikulin, F. Pardo, J. Parker-Holder, J. Quan, T. Rocktäschel, H. Sahni, T. Schaul, Y. Schroecker, S. Spencer, R. Steigerwald, L. Wang, and L. Zhang, "Vision-language models as a source of rewards," 2024.