LORD: Large Models based Opposite Reward Design for Autonomous Driving

Xin Ye* Feng Tao* Abhirup Mallik Burhaneddin Yaman† Liu Ren

Bosch Research North America &
Bosch Center for Artificial Intelligence (BCAI)
{xin.ye3, feng.tao2, abhirup.mallik, burhaneddin.yaman,
liu.ren}@us.bosch.com

Abstract. Reinforcement learning (RL) based autonomous driving has emerged as a promising alternative to data-driven imitation learning approaches. However, crafting effective reward functions for RL poses challenges due to the complexity of defining and quantifying good driving behaviors across diverse scenarios. Recently, large pretrained models have gained significant attention as zero-shot reward models for tasks specified with desired linguistic goals. However, the desired linguistic goals for autonomous driving such as "*drive safely*" are ambiguous and incomprehensible by pretrained models. On the other hand, undesired linguistic goals like "*collision*" are more concrete and tractable. In this work, we introduce LORD, a novel large models based opposite reward design through undesired linguistic goals to enable the efficient use of large pretrained models as zero-shot reward models. Through extensive experiments, our proposed framework shows its efficiency in leveraging the power of large pretrained models for achieving safe and enhanced autonomous driving. Moreover, the proposed approach shows improved generalization capabilities as it outperforms counterpart methods across diverse and challenging driving scenarios.

1 Introduction

Autonomous driving is a challenging task that demands both deep comprehension of the environment and ability to swiftly reacting to changes. Rapid advancements in deep learning have triggered significant progress in this domain, mainly through imitation learning (IL) approaches [3,14,15,26]. Despite showing impressive results, the performance of these IL methodologies heavily relies on the size of data [5,27]. Thus, IL approaches are inherently subject to dataset bias and lack rational in decision making. To address these challenges, reinforcement learning (RL) based approaches that optimize driving policies by interacting with the environment and maximizing the rewards have gathered growing interest as alternatives for autonomous driving tasks [16,17,31].

Reinforcement learning thrives when paired with effective reward functions, which serve as the guiding principles for learning optimal behaviors [4,9–12,44].

* Equal contributions.† Corresponding author

However, crafting these reward functions often proves costly, particularly when relying on human feedback for their formulation [4, 43]. Additionally, manually specifying such reward functions presents a formidable challenge in avoiding reward hacking [25]. This challenge is further compounded in the autonomous driving tasks due to the difficulty of defining and quantifying good driving behavior, as well as generalizing them across diverse driving scenarios. To address these challenges, leveraging large pretrained models emerges as a promising solution for crafting efficient and generalizable reward functions for autonomous driving systems.

Large pretrained models exhibit human-like reasoning abilities and have demonstrated remarkable performance across a spectrum of tasks [2, 6, 23, 28, 41, 42, 45]. In the realm of robotics, the integration of these models in reward functions have shown good performance and promising generalization capabilities [7,30,35]. In these works, experiments evolve around tasks where the desired goal state is either known or can be easily defined. Thus, describing the desired goal state in form of a linguistic goal which is comprehensible by pretrained models enables exploiting the pretrained models as zero-shot reward models. While these approaches show good performance in variety of robotic tasks, they encounter significant challenges in autonomous driving. In such intricate scenarios, direct linguistic goals become particularly arduous for large models to grasp, highlighting the need for more nuanced strategies to ensure effective comprehension and decision-making.

In this work, we present a novel approach to reward design for safe and enhanced autonomous driving: the concept of opposite reward design through undesired linguistic goals in order to leverage large pretrained models as zero-shot reward models. In autonomous driving scenarios, linguistically defining desired goal state such as "*drive safely*" can be ambiguous and challenging. However, undesired linguistic goals, such as "*collision*", offer a more tangible and understandable objective for both humans and large pretrained models. By introducing opposite reward design, we aim to enhance the interpretability, generalizability and effectiveness of autonomous driving systems, making them more capable of navigating complex environments while prioritizing safety. To harness the full potential of our approach, we construct a closed-loop driving environment. We conduct extensive experiments on large pretrained image, video, and language models to evaluate the efficacy of our proposed framework for closed-loop autonomous driving tasks. Notably, our framework achieves significantly improved performance over counterpart methods across various driving scenarios.

The main contributions of this work are summarized as follows:

- We propose LORD, a Large models based Opposite Reward Design, which addresses the ambiguity of linguistic goals in autonomous driving with comprehendable undesired linguistic goals. To the best of our knowledge, this is the first work that leverages large pretrained models with undesired goals in embodied AI domain.

– LORD leverages large pretrained image, video and language models with a cosine distance objective for an efficient reward function design for RL based autonomous driving.
– Through extensive experiments, we show LORD consistently achieves significantly improved generalization performance over counterpart methods across various challenging driving scenarios.

2 Related Work

2.1 Reward Design for Reinforcement Learning

Reward function plays a pivotal role in reinforcement learning, dictating the behavior of autonomous agents. Unlike games where rewards occur naturally, creating a reward function for real-world tasks needs an intentional design that requires extensive expert supervision. The difficulty motivates many researchers to directly learn a reward function by observing a human expert performing the task [1, 8, 24, 44]. However, these approaches become overly complex when applied to tasks with high dimensional state and action space. More recently, some work leverage discriminator networks with demonstration sets to assign rewards based on the likelihood of a state belonging to the demonstration set [9, 10, 12]. Training these discriminator networks still requires a substantial number of expert demonstrations which is not always feasible due to the limited availability of such demonstrations. Conversely, another line of work involves using human pairwise preferences over data samples to learn the reward function [4, 11]. While these methods offer good results in some tasks, they often rely on either a large number of valid goal states or significant human effort, making them impractical for many applications, particularly in the context of autonomous driving, where efficiency and scalability are paramount.

2.2 Reward Design with Large Pretrained Models

Large pretrained models have recently gained interest as an alternative way for reward design. Describing the goal state through language has been the centerpiece for designing powerful zero-shot reward models. A line of work in this direction has focused on using large language models (LLMs) [7, 13, 18]. For example, ELLM [7] utilizes LLMs to reward agent for achieving goals suggested by LLMs. In another work, LLMs have been used as a proxy reward function to capture human preferences by prompting desired behaviors [18]. More recent works use large pretrained multi-modal models. Among these works, VLM-RMs [30] and RoboCLIP [35] leverages vision language models and video language models, respectively, with desired linguistic goals for the reward design. However, these works have focused on specific robotic applications where the desired goal state exists and is comprehensible by large pretrained models [30, 35]. In contrast, our work focuses on autonomous driving where desired goal states either does not exist or not comprehensible by large pretrained models due to the innate ambiguity of desired linguistic goals.

2.3 Language Models in Autonomous Driving

The success of language models in robotics have sparked the interest for incorporation of language models in autonomous driving [21,26,34,40]. Several works have leveraged LLMs for explainable autonomous driving. LINGO-1 presents a commentator model for reasoning and explainability by training a model combining language with vision and action [36]. Similarly, DriveLM introduces a visual question answering approach to interpret driving actions [34]. Other works have focused on enhancing the planning performance for autonomous driving through language. VLP introduces a plug-in approach by incorporating LLMs with contrastive learning objective into vision-only end-to-end autonomous driving systems [26]. Several GPT-based driver agents have also been introduced as an alternative to existing IL and RL based driver agents [21, 37, 40]. Among these works, DiLu presents a systematic framework which combines reasoning and reflection mechanism to improve the decision making capability of the driver agent [37]. Nevertheless, the substantial reliance on GPT models presents several limitations, such as the occurrence of hallucinations due to the lack of grounding, as well as latency issues that are crucial for real-world deployment. Unlike these approaches, our work utilizes an RL agent with a reward mechanism based on large pretrained models to circumvent these challenges.

3 Methodology

LORD utilizes large pretrained models to generate step-wise rewards for autonomous agents (i.e. ego vehicles) aiming to encourage desired driving behavior and outcomes. This is done by evaluating how different the state of the autonomous agent at each time step is from the undesired goal state described by our opposite linguistic goal for the first time. Moreover, since the state of the autonomous agent can be observed as an image, a video and a linguistic description, we investigate vision-and-language, video-and-language and language models for embedding the agent's state and undesired goal state, respectively. Cosine similarity between the agent's and goal state's embeddings is calculated. Followingly, the agent receives cosine distance (i.e., 1 - cosine similarity) as the reward for each step. We integrate LORD with reinforcement learning algorithm for closed-loop autonomous driving task. Fig. 1 shows an overview of our LORD powered RL framework. Details of our method are presented in the following subsections. We first provide a problem formulation of the closed-loop autonomous driving task in Sec. 3.1. Subsequently, we describe our proposed LORD and its integration with RL in Sec. 3.2 and Sec. 3.3, respectively.

3.1 Problem Formulation

We formulate the closed-loop autonomous driving task as a Partially Observable Markov Decision Process (POMDP) problem defined by a 7-tuple $< S, O, \theta, A, T, R, \gamma >$. Specifically, S is the state space of the ego vehicle. O is the observation space that is determined by emission function $\theta : S \times O \rightarrow [0, 1]$. A is a set of

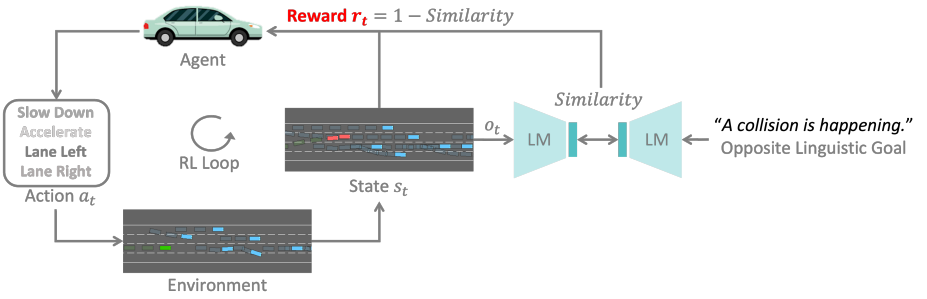


Fig. 1: An overview of our LORD powered reinforcement learning framework for closed-loop autonomous driving task. LORD firstly measures cosine similarity between agent's state and undesired goal state using large pretrained models. Followingly, it returns cosine distance as the reward to the agent.

actions used to drive the ego vehicle. $T : S \times A \times S \rightarrow [0, 1]$ is a state transition probability function. $R : \Omega \times A \rightarrow \mathbb{R}$ is a reward function to reward desired driving behavior or outcomes, and $\gamma \in (0, 1]$ is a discount factor. To learn a good driving policy $\pi(a_t|s_t)$ informing the ego vehicle which action a_t to take at the state s_t, we maximize the expected discounted cumulative rewards $\mathbb{E}[\sum_t^{\infty} \gamma^t r_{t+1}(a_t, o_{t+1})|s_t]$. The focus of this paper is to efficiently and effectively define the reward function R based on the ego vehicle's observation O and our opposite linguistic goal noted as *goal*. Note that the driving policy π is still learned from the ego vehicle's state S.

3.2 Large Models based Opposite Reward Design

Opposite Linguistic Goal. Recent work in the field of robotics has shown a great success in taking large pretrained models as a zero-shot reward model for robotic tasks [30, 35]. An essential element to the success is that they are able to describe the desired task and expected goal state with linguistic descriptions accurately. For example, a task of "*a humanoid robot kneeling*" is actually an accurate description to the expected goal state [35]. In this case, when large pretrained models project the agent's current state and the desired goal state into an embedding space, the distance between the two embeddings forms a natural reward measuring how close the agent is to the desired goal state. However, for autonomous driving task, while the ultimate

Fig. 2: The insight of using an opposite goal. In autonomous driving tasks, desired goal states such as "*drive safely*" are ambiguous to grasp, whereas undesired goal states such as "*collision*" are tractable more comprehensible to humans and large pretrained models.

target is to let the ego vehicle drive safely, it is difficult to describe concretely what the goal state is since there are infinite ways to keep safe in driving. As a result, the embedding of the abstract target goal "*ego is driving safely*" is not semantically comparable to the ego vehicle's states and observations. On the contrary, describing unexpected states that the ego vehicle should avoid is more tractable. For example, we can easily imagine what a "*collision*" looks like and ground it into an observation. Fig. 2 better illustrates our insight. To this end, we propose to use an opposite linguistic goal "*a collision is happening*" as opposed to the target goal "*ego is driving safely*".

Observation Representations The ego vehicle's observation o_t at time step t can be represented in multiple ways: (1) an image capturing the spatial information around the ego vehicle; (2) a video containing both spatial and temporal information with respect to the ego vehicle; (3) a linguistic description to the ego vehicle's current situation.

- **Image based Observation.** Using an image to represent an observation is straightforward and efficient. A raw image can be obtained directly from camera-like sensors and a more informative image, like Bird's Eye View (BEV) can be built through advanced computer vision method [20]. The high dimensional image based observation contains rich information including surrounding objects and road elements and thus has been wildly adopted for autonomous driving tasks [14,26]. In this work, without loss of generality, we use the image rendered by the deployment environment as the observation. To project the image based observation and our opposite linguistic goal into the same embedding space, we adopt the vision-and-language model CLIP [28] to encode the observation and the goal with its image and text encoders respectively. In particular, we select the CLIP model pretrained on the large-scale dataset LAION-2B [32] as it has been shown to have superior performance in [30].

- **Video based Observation.** A single static image may not be able to capture the kinematic information, such as the speed and the acceleration of both the ego and npc vehicles. These information is critical for autonomous driving. For example, a vehicle is more likely to collide in a congested traffic scenario if it is driving at a higher speed and with a greater acceleration. To get an observation containing the kinematic information, we generate a video by stacking the latest 30 consecutive frames of the images at each time step t. To compare the video based observation and our opposite linguistic goal, we utilize the video-and-language model S3D [39]. S3D is pretrained on HowTo100M dataset [22] which consists of diverse short clips of human demonstrators performing daily tasks. Therefore, the video encoder and text encoder of S3D can encode our video based observation and opposite linguistic goal into a semantically meaningful latent space.

– **Text based Observation.** Inspired by the recent success of large language models (LLMs) and their applications, recent work starts to use textual scenario descriptions as the observations [7, 37]. In this way, they can leverage LLMs' exceptional human-level abilities to perform the driving task by asking LLMs to make decisions upon the text based observations. In our work, we describe the ego vehicle's current situation in terms of potential collisions. Specifically, we calculate the time to collision (ttc) with each of the surrounding vehicles based on their distance and speed difference. If the ttc is smaller than a predefined threshold, we describe it in our text based observation by "*A collision will be happening in {ttc} seconds.*". We also describe conditional collisions by "*A collision would happen in {ttc} seconds if ego makes a left/right lane change.*". To determine how similar the text based observation and our opposite linguistic goal are, we adopt a language model to convert them into embeddings that capture their semantic information. In practice, we use the pretrained SentenceBERT model [29] which is designed to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity.

Opposite Reward Generation. Given our opposite linguistic goal *goal* and the observation o_t at time step t, we define our reward r_t as follows,

$$similarity(o_t, goal) = \frac{LM^o(o_t) \cdot LM^g(goal)}{||LM^o(o_t)|| \cdot ||LM^g(goal)||} \tag{1}$$

$$r_t = 1 - similarity(o_t, goal) \tag{2}$$

where LM^o and LM^g denote the large pretrained models used to encode the observation and the opposite linguistic goal, respectively. Our model choices are specified in Sec. 3.2. Here, we adopt the cosine distance between the observation embedding $LM^o(o_t)$ and the opposite linguistic goal embedding $LM^g(goal)$ to quantify the reward value for the ego vehicle. In this way, when the ego vehicle is further away from the undesired situation described by our opposite linguistic goal, the ego vehicle can get a higher reward.

3.3 RL Training with LORD

LORD can be integrated with any standard reinforcement learning algorithms. In this work, we follow [38], the latest state-of-the-art reinforcement learning work in autonomous driving domain, by adopting Proximal Policy Optimization (PPO) [33] algorithm to learn an optimal driving policy $\pi(a_t|s_t)$ for the ego vehicle. LORD is only used in training. During testing, we evaluate the ego vehicle by performing the action $a_t \sim \pi(a_t|s_t)$ at time step t that only depends on the state s_t.

4 Experiments

We conduct experiments on Highway-env [19] to validate our LORD framework for closed-loop autonomous driving task. In particular, we aim to seek the answers to the following questions:

- Is our LORD framework effective in addressing the closed-loop autonomous driving task?
- How does each variant of our method work for the closed-loop autonomous driving task?
- Does our opposite reward design contribute to the success of our method?

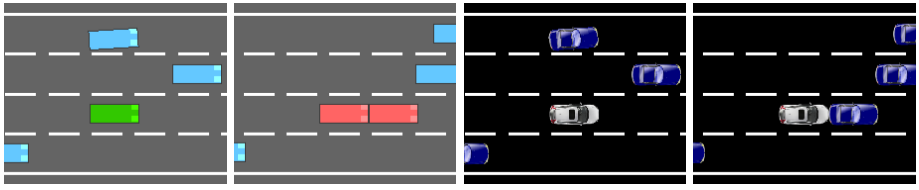
We first introduce our experiment setting in Sec. 4.1, and we rearrange the remaining of this section to answer each of these questions. In Sec. 4.2, we compare LORD with baseline methods to answer the first question. We answer the second question in Sec. 4.3 by conducting an in-depth analysis of the reward values generated by LORD. Ablation studies are provided in Sec. 4.4 to answer the third question. Sec. 4.5 presents qualitative results .

4.1 Experiment Setting

Simulation Environment. We adopt Highway-env [19] to conduct all experiments. Highway-env is a well-established simulation platform for closed-loop autonomous driving task in which npc vehicles can react to ego's behavior. It is wildly used in the research work of autonomous driving [37, 38]. We follow [38] to set up the environment. More particularly, we define the state space S of the ego vehicle as its kinematic observation which is a $V \times F$ array provided by the environment that describes a list of V nearby vehicles by a set of features of size F, including the vehicles' positions, speeds and orientations. We adopt the discrete meta-actions as the ego vehicle's action space A that consists of lane and speed change. In addition, we also create various traffic situation by setting the density of vehicles and the number of lanes as [37] does to test all methods. Detailed configurations can be found in the supplementary materials.

Domain Adaptation. Highway-env provides a simplified visualization. All vehicles are depicted as rectangles where the ego vehicle is colored in green and npc vehicles are colored in blue (see Fig. 3a). When a collision happens, the victim vehicles are colored in red as Fig. 3b illustrates. These rendered images are likely out of the training distribution of the large pretrained models. In consequence, our large model based rewards may not work well for the settings with image or video based observations. To remedy this issue, we modify the graphics of the Highway-env by replacing the rectangle textures with more photorealistic car images. Besides, we also remove the useless background of the images. Fig. 3c and Fig. 3d show the snapshots of our modified Highway-env in which the white car denotes the ego vehicle and blue cars are npc vehicles. Note that such a modification is only used for reward generation when using image and video

based observations. Meanwhile, we also customize our opposite linguistic goals to adapt to the image and video based observations. To be specific, we define the opposite linguistic goal for image and video based observation as "*White car collides with a blue car.*". In this work, we don't fine-tune any large models to adapt to the Highway-env.



(a) Original driving. **(b)** Original colliding. **(c)** Modified driving. **(d)** Modified colliding.

Fig. 3: Illustrations of the original and the modified Highway-env. In the modified environment, white car denotes the ego vehicle and blue cars depict the npc vehicles.

4.2 Comparison with Baseline Methods

We compare 3 variants of our method (i.e. observation represented by image, video and text) with following baselines.

- GRAD [38]. The latest state-of-the-art reinforcement learning method for the Highway-env [19]. It learns driving policy from a graph-based state representation using PPO [33] algorithm. The reward is a sum of 1) a constant surviving reward which is 0.2; 2) a speed reward linearly mapped from the speed of $(20, 40)$ to $(0, 0.8)$.
- Const. Similar to GRAD except the reward only consists of a constant surviving reward in order to motivate the ego vehicle to survive as long as possible.
- DiLu [37]. The latest state-of-the-art large language model-based method. It leverages large language models to perform step-wise decision-making for autonomous driving task.

We optimize all methods on `lane-4-density-2` setting in Highway-env and evaluate them on various traffic situations, namely `lane-4-density-2`, `lane-5-density-2.5` and `lane-5-density-3`. Each evaluation is repeated 17 times with different random seeds specified in the code repository[1] of DiLu [37]. We report Success Rate (SR), Traveled Distance (TD) and Rewards (RE) achieved by these methods in Table 1. Success Rate (SR) is defined in [37] where a success denotes that the ego vehicle survives over 30 time steps without any collisions. Traveled Distance (TD) means how far the ego vehicle drives along the x axis before a collision happens. We also adopt the reward function defined in [38] to calculate

[1] https://github.com/PJLab-ADG/DiLu

Table 1: Performance comparisons of all methods in different traffic situations of highway environments. All methods are only optimized on `lane-4-density-2` setting and evaluated on the `lane-4-density-2`, `lane-5-density-2.5`, and `lane-5-density-3`. The best results are highlighted in bold and the second-best results are marked with an underline. (SR: Success Rate, TD: Traveled Distance, RE: Rewards)

	lane-4-density-2			lane-5-density-2.5			lane-5-density-3		
Method	SR↑	TD↑	RE↑	SR↑	TD↑	RE↑	SR↑	TD↑	RE↑
GRAD [38]	<u>94.12</u>	**930.14**	**20.16**	<u>88.24</u>	**930.99**	**20.35**	58.82	**664.55**	**13.61**
Const.	**100.0**	610.17	6.39	<u>88.24</u>	568.03	5.80	52.94	424.07	4.18
DiLu [37]	70.00	-	-	65.00	-	-	35.00	-	-
LORD image	**100.0**	<u>694.20</u>	<u>9.75</u>	<u>88.24</u>	<u>652.02</u>	<u>9.12</u>	<u>64.71</u>	578.41	<u>8.39</u>
LORD video	**100.0**	630.37	7.24	<u>88.24</u>	612.77	7.67	**82.35**	<u>599.86</u>	7.78
LORD text	**100.0**	682.24	9.27	**94.12**	630.53	7.94	58.82	493.24	5.59

the Rewards (RE) as an additional metric to evaluate how well the ego vehicle drives.

As shown in Table 1, in the in-domain training environment `lane-4-density-2`, all variants of our method achieve 100% success rate (SR), which is 5.88% higher than the RL-only baseline GRAD and 30% higher than the LLMs-only baseline DiLu. LORD also shows strong generalization performance in unseen out-of domain scenarios. In particular, for `lane-5-density-2.5`, our method with image and video based observation achieves 88.24% SR, which is same as GRAD and Const., and 23.24% higher than DiLu. With text based observation, our method outperforms all the counterpart methods by achieving further 5.88% SR improvement. In the most complex traffic situation, i.e. `lane-5-density-3`, our method with image and video based observation outperforms the best performing counterpart method (GRAD) in terms of SR by 5.89% and 23.53%, respectively. These comparisons demonstrate that the driving policy learned by our LORD not only helps the ego vehicle to better avoid collisions in seen environments but also generalizes well to unseen environments.

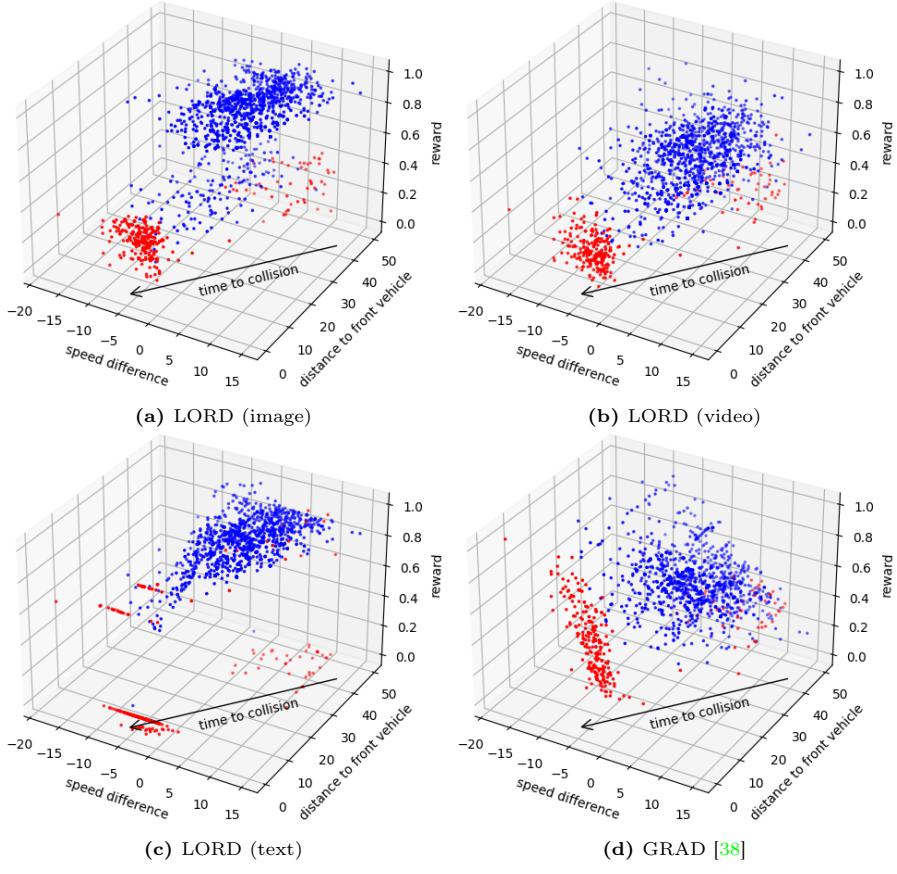
Table 2: The performance of our LORD with addition of the speed reward using text based observation.

lane-4-density-2			lane-5-density-3		
SR↑	TD↑	RE↑	SR↑	TD↑	RE↑
94.12	925.94	19.57	64.71	716.47	14.32

For the metrics of Traveled Distance (TD) and Rewards (RE), our method also outperforms the Const. baseline in all traffic situations, showing that our large model based reward motivates the ego vehicle to drive faster than a constant survival reward. We note that GRAD achieves much higher TD and RE. We hypothesize it is because GRAD directly optimizes RE in which the speed

reward encourages the ego vehicle to speed up. To validate our hypothesis, we conduct an experiment by adding the additional speed reward into our LORD method (observation represented by text). Since the speed is a concrete information and can be easily obtained from the sensory inputs, adding the speed reward is straightforward. We report the results in Table 2 and the results show that our method achieves 51.92 higher TD and 0.71 higher RE than the GRAD baseline in the challenging `lane-5-density-3` setting.

4.3 Deep Dive into LORD



(a) LORD (image) **(b)** LORD (video)

(c) LORD (text) **(d)** GRAD [38]

Fig. 4: Illustrations of the reward values generated by our LORD under various observation representations and GRAD [38] for different states. We distinguish different states in terms of the ego vehicle's distance to its nearest front vehicle and their speed difference. In this way, time to collision can be roughly estimated. Blue points denote collision-free states while red points indicate the ego vehicle collides with other vehicles.

Table 1 shows comparison of the 3 variants of our method as detailed in Sec. 3. Overall, our method with different observation representations achieve similar performance. Notably, LORD achieves 100% success rate in the training environment `lane-4-density-2` under all image, video and text based observations, indicating the effectiveness of our LORD method. In terms of generalization ability, LORD with text based observation achieves higher success rate in `lane-5-density-2.5` while image and video based observation helps LORD perform better in `lane-5-density-3` environment.

To have an in-depth understanding of the remarkable performance achieved by LORD, we illustrate in Fig. 4 the reward values generated by LORD for different states. Since the state of the ego vehicle is a high dimensional vector that includes position, speed and orientation information of all vehicles in the scenario, it is impractical to visualize the rewards with respect to the states directly. It is also extremely difficult to evaluate how good a state is and how well the reward values align with the goodness of the states. Therefore, we instead choose 2 features to represent a state, namely the distance to the nearest front vehicle and the vehicle's relative speed comparing to the ego vehicle. In this way, we can estimate their time to collision (ttc), and use the ttc as a surrogate metric to measure how good the state is. In addition, we also distinguish the collision and non-collision states by coloring them in red and blue, respectively.

As shown in Fig. 4, the collision states denoted by the red points are centered as expected at the area having small distance to the front vehicle and negative speed difference, which means that the ego vehicle collides with the front vehicle at a higher speed. From Fig. 4a, 4b and 4c, we can see that our LORD consistently assigns smaller rewards to these collision states. For non-collision states denoted by the blue points, we also observe a decrease in reward values along the trend over the time to collision, especially when the distance is small and the speed difference is negative as such a collision is more likely to happen. We note that when the ego vehicle is far away from the front vehicle or drives slower than the front vehicle, more tailored features might be needed to evaluate the state in such cases. Collisions could also happen as the red points in the area of large distance and positive speed difference show. In comparison, GRAD [38] gives high rewards even to the collision states as Fig. 4d shows. The rewards also show a clear upward trend when the speed difference becomes negative, i.e., the ego vehicle drives faster than its front vehicle. It is expected as GRAD adopts a speed reward but such a reward strategy doesn't encourage a safe driving policy.

4.4 Effectiveness of Opposite Reward Design

To validate our opposite reward design, we conduct ablation study to compare the 3 variants of our method with the ones using the corresponding target goals. To be specific, when using image and video to represent the observation, we set the target goal as "*White car drives safely.*". For the setting using text to represent the observation, the target goal is set as "*Ego is driving safely.*". In addition, when using the target goal, we define the reward r_t at time step t as the cosine

similarity between the state o_t and the target goal, i.e. $r_t = similarity(o_t, goal)$ where the $similarity(\cdot, \cdot)$ function is defined in Eq. 1.

Table 3: An ablation study of our method using opposite goals versus target goals. Our approach witnesses the opposite goals show significantly improved generalization performance on challenging scenarios.

		lane-4-density-2			lane-5-density-2.5			lane-5-density-3		
State	Goal	SR↑	TD↑	RE↑	SR↑	TD↑	RE↑	SR↑	TD↑	RE↑
Image	oppo.	**100.0**	**694.20**	9.75	**88.24**	**652.02**	9.12	**64.71**	**578.41**	**8.39**
	target	70.59	672.92	**10.03**	47.06	639.69	**10.01**	29.41	527.48	7.78
Video	oppo.	**100.0**	630.37	7.24	**88.24**	612.77	7.67	**82.35**	**599.86**	**7.78**
	target	**100.0**	**679.10**	**9.14**	76.47	**646.01**	**8.34**	58.82	519.54	6.32
Text	oppo.	**100.0**	**682.24**	**9.27**	**94.12**	**630.53**	**7.94**	**58.82**	**493.24**	**5.59**
	target	**100.0**	672.88	8.87	76.47	583.62	7.52	23.53	305.66	3.81

The comparisons of the methods using our opposite goals and target goals are shown in Table 3. Overall, in terms of success rate (SR), all variants of our method with opposite reward design consistently outperform their counterparts using target goals in all traffic situations, indicating that our opposite reward design is more effective in learning a safe driving policy. In addition, while the method using target goals achieves competitive performance in in-domain traffic situation `lane-4-density-2`, its performance degrades a lot in more difficult out-of-domain traffic situations. In particular, with video and text based observation, the method using target goals achieve 100% SR in `lane-4-density-2`. However, its SR drops to 76.47% in `lane-5-density-2.5` and finally to 58.82% and 23.53% respectively in `lane-5-density-3`. It demonstrates that the target goal reward design is not generalizable for autonomous driving task. Comparing to our method, in `lane-5-density-2.5`, we note that the method with target goal reward design has a slightly higher rewards (RE) under image and video based observation. However, its success rates (SR) are much lower. Specifically, while the target goal reward design obtains 0.89 and 0.67 higher RE under image and video observation, the SR is 41.18% and 11.77% lower than ours. We explain this as the method using target goals learns to accelerate but ignores the importance of avoiding collisions. More importantly, in the most difficult traffic situation `lane-5-density-3`, our opposite reward design outperforms the target goal reward design in all metrics, showing the superiority of our method in generalization.

4.5 Qualitative Results

Fig. 5 depicts how the driving policy learned by our LORD in `lane-4-density-2` setting of Highway-env performs in more challenging `lane-5-density-3` envi-

Fig. 5: Illustrations of how the driving policy learned by our LORD with image, video and text based observation performs in the `lane-5-density-3` setting of Highway-env. The ego vehicle is colored in green with a line depicting its past trajectory. The ego vehicle behaves properly in the congested traffic scenarios.

ronments with image, video and text based observation respectively. As shown in the figure, the ego vehicle learns to behave properly in the congested traffic scenarios even if it hasn't encountered such situations before. For example, from the top row of Fig. 5, we can see that the ego vehicle chooses to follow the front vehicle when there is no room for a lane change. Once the ego vehicle surpasses all left-lane vehicles, it makes a left lane change to gain more space. The ego vehicle also learns to overtake its front vehicle with video based observation as illustrated in the middle row. Similarly, with text based observation, the ego vehicle also succeeds in improving its situation by changing its lane to the right. We provide more qualitative results in the supplementary materials.

5 Conclusion and Future Work

In this paper, we introduce a novel large models based opposite reward design (LORD) framework for autonomous driving. LORD presents opposite reward design through undesired linguistic goals for efficient use of large pretrained models as zero-shot reward mechanism since such undesired goals are more tractable and comprehensible for large pretrained models compared to desired ones. We leverage large pretrained image, video and language models with a cosine distance objective in our reward function. We integrate LORD with reinforcement learning algorithms to perform autonomous driving tasks. Extensive experiments on the closed-loop autonomous driving tasks show the efficacy of the opposite reward design mechanism over the desired target goal reward design. Moreover, LORD achieves improved generalization performance over the counterpart reinforcement learning and language model based methods.

Our experiments are currently confined to Highway-env simulation as baseline approaches only report their performance in this simulation. We also note that the exceptional abilities of large pretrained models have not been fully utilized due to the simulated images and videos we input and thereby limits the

performance of our approach. We will assess our approach on more environments in our future work. In this work, our study has evolved around the "*collision*" as our opposite linguistic goal. In reality, there are numerous undesired driving behaviors and situations the autonomous agents should avoid, such as running a red light, occupying an emergency lane or violating other traffic rules. With our opposite reward design, we can add these undesired behaviors as additional opposite linguistic goals into our LORD framework so that we can further optimize the driving policy. In our future work, we will explore these in more sophisticated environments as Highway-env does not contain such detailed information.

References

1. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: International Conference on Machine Learning (2004) 3
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022) 2
3. Chen, D., Zhou, B., Koltun, V., Krähenbühl, P.: Learning by cheating. In: Conference on Robot Learning. pp. 66–75. PMLR (2020) 1
4. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems **30** (2017) 1, 2, 3
5. Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9329–9338 (2019) 1
6. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: Palm-e: An embodied multimodal language model. arXiv preprint arXiv:2303.03378 (2023) 2
7. Du, Y., Watkins, O., Wang, Z., Colas, C., Darrell, T., Abbeel, P., Gupta, A., Andreas, J.: Guiding pretraining in reinforcement learning with large language models (2023) 2, 3, 7
8. Finn, C., Levine, S., Abbeel, P.: Guided cost learning: Deep inverse optimal control via policy optimization. In: International conference on machine learning. pp. 49–58. PMLR (2016) 3
9. Fu, J., Luo, K., Levine, S.: Learning robust rewards with adverserial inverse reinforcement learning. In: International Conference on Learning Representations (2018) 1, 3
10. Fu, J., Singh, A., Ghosh, D., Yang, L., Levine, S.: Variational inverse control with events: A general framework for data-driven reward definition. Advances in Neural Information Processing Systems **31** (2018) 1, 3
11. Hejna III, D.J., Sadigh, D.: Few-shot preference learning for human-in-the-loop rl. In: Conference on Robot Learning. pp. 2014–2025. PMLR (2023) 1, 3
12. Ho, J., Ermon, S.: Generative adversarial imitation learning. Advances in Neural Information Processing Systems **29** (2016) 1, 3
13. Hu, H., Sadigh, D.: Language instructed reinforcement learning for human-ai coordination. In: Proceedings of the 40th International Conference on Machine Learning. ICML'23, JMLR.org (2023) 3

14. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., Li, H.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023) 1, 6
15. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. ICCV (2023) 1
16. Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.M., Lam, V.D., Bewley, A., Shah, A.: Learning to drive in a day. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 8248–8254. IEEE (2019) 1
17. Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S., Pérez, P.: Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems **23**(6), 4909–4926 (2021) 1
18. Kwon, M., Xie, S.M., Bullard, K., Sadigh, D.: Reward design with language models. In: The Eleventh International Conference on Learning Representations (2022) 3
19. Leurent, E.: An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env (2018) 8, 9
20. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022) 6
21. Mao, J., Qian, Y., Zhao, H., Wang, Y.: Gpt-driver: Learning to drive with gpt. arXiv preprint arXiv:2310.01415 (2023) 4
22. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2630–2640 (2019) 6
23. Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., Roth, D.: Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys **56**(2), 1–40 (2023) 2
24. Ng, A.Y., Russell, S., et al.: Algorithms for inverse reinforcement learning. In: International Conference on Machine Learning (2000) 3
25. Pan, A., Bhatia, K., Steinhardt, J.: The effects of reward misspecification: Mapping and mitigating misaligned models. In: International Conference on Learning Representations (2021) 2
26. Pan, C., Yaman, B., Nesti, T., Mallik, A., Allievi, A.G., Velipasalar, S., Ren, L.: Vlp: Vision language planning for autonomous driving (2024) 1, 4, 6
27. Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset shift in machine learning. Mit Press (2008) 1
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 2, 6
29. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics (2019) 7

30. Rocamonde, J., Montesinos, V., Nava, E., Perez, E., Lindner, D.: Vision-language models are zero-shot reward models for reinforcement learning. In: NeurIPS 2023 Foundation Models for Decision Making Workshop (2023) 2, 3, 5, 6
31. Sallab, A.E., Abdou, M., Perot, E., Yogamani, S.K.: Deep reinforcement learning framework for autonomous driving. In: Autonomous Vehicles and Machines (2017), https://api.semanticscholar.org/CorpusID:12064877 1
32. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022) 6
33. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017) 7, 9
34. Sima, C., Renz, K., Chitta, K., Chen, L., Zhang, H., Xie, C., Luo, P., Geiger, A., Li, H.: Drivelm: Driving with graph visual question answering. arXiv preprint arXiv:2312.14150 (2023) 4
35. Sontakke, S., Zhang, J., Arnold, S., Pertsch, K., Bıyık, E., Sadigh, D., Finn, C., Itti, L.: Roboclip: One demonstration is enough to learn robot policies. Advances in Neural Information Processing Systems **36** (2024) 2, 3, 5
36. Wayve: Lingo-1: Exploring natural language for autonomous driving. https://wayve.ai/thinking/lingo-natural-language-autonomous-driving/ (2023) 4
37. Wen, L., Fu, D., Li, X., Cai, X., Tao, M., Cai, P., Dou, M., Shi, B., He, L., Qiao, Y.: Dilu: A knowledge-driven approach to autonomous driving with large language models. In: The Twelfth International Conference on Learning Representations (2023) 4, 7, 8, 9, 10
38. Xi, Z., Sukthankar, G.: A graph representation for autonomous driving. In: The 36th Conference on Neural Information Processing Systems Workshop (2022) 7, 8, 9, 10, 11, 12
39. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018) 6
40. Xu, Z., Zhang, Y., Xie, E., Zhao, Z., Guo, Y., Wong, K.K., Li, Z., Zhao, H.: Drivegpt4: Interpretable end-to-end autonomous driving via large language model. arXiv preprint arXiv:2310.01412 (2023) 4
41. Yang, S., Nachum, O., Du, Y., Wei, J., Abbeel, P., Schuurmans, D.: Foundation models for decision making: Problems, methods, and opportunities. arXiv preprint arXiv:2303.04129 (2023) 2
42. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021) 2
43. Zhan, H., Tao, F., Cao, Y.: Human-guided robot behavior learning: A gan-assisted preference-based reinforcement learning approach. IEEE Robotics and Automation Letters **6**(2), 3545–3552 (2021) 2
44. Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K., et al.: Maximum entropy inverse reinforcement learning. In: AAAI. vol. 8, pp. 1433–1438. Chicago, IL, USA (2008) 1, 3
45. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023) 2

Appendix

A Implementation Details

A.1 Detailed Setup of Highway-env

We follow the code repository [2] of GRAD baseline to setup Highway-env. Table 4 shows our customized configurations and we use default values for other parameters. During training, we set `lane_count` as 4 and `vehicles_density` as 2 to train all methods in `lane-4-density-2` setting. In addition, we set `duration` as 60 to train the agent to address long-horizon tasks. During testing, we set `lane_count` and `vehicles_density` accordingly and change `duration` to 30 to evaluate the success rate of all methods in `lane-4-density-2`, `lane-5-density-2.5` and `lane-5-density-3` settings.

Table 4: Configurations of Highway-env for training.

Parameter	Value
observation	
–type	Kinematics
–features	[presence, x, y, vx, vy, cos_h, sin_h, heading]
–absolute	True
–normalize	True
–vehicles_count	33
–see_behind	True
action	
–type	DiscreteMetaAction
–target_speeds	[20, 25, 30, 35, 40]
duration	60
ego_spacing	4
lane_count	4
vehicles_density	2

A.2 Observation Design

To enable a more efficient use of large pretrained models as zero-shot reward models, we empirically adopt the following observation designs as inputs to the large pretrained models. (1) For image based observation, we adopt the simulated image rendered by Highway-env with the parameter `scaling` being set to 10. We then replace the rectangles used to represent the ego and npc vehicles with more photorealistic car images. We further remove the image background and we crop the image to the size of 224×224 centered on the ego vehicle. (2) For video

[2] https://github.com/zerongxi/graph-sdc

based observation, we stack the latest 30 consecutive image based observations with a 15Hz frequency. (3) For text based observation, we only pay attention to the nearby vehicles that are within $5 \times ego\_speed$ meters of the ego vehicle and drive on the same, left or right lane of the ego vehicle. We then calculate the ego vehicle's time to collision (ttc) to each of these attended vehicles. If a vehicle drives on the same lane of the ego vehicle and the ttc is smaller than 5s, we describe it in our text based observation by "*A collision will be happening in {ttc}s.*". Otherwise, we give a description of "*No foreseeable collision in 5s.*". We also describe conditional collisions by "*A collision would happen in {ttc}s if ego makes a left/right lane change.*". Examples of the three types of observations can be found in Fig. 6 and Fig. 7.

B Case Study

B.1 Rewards from Different Observations

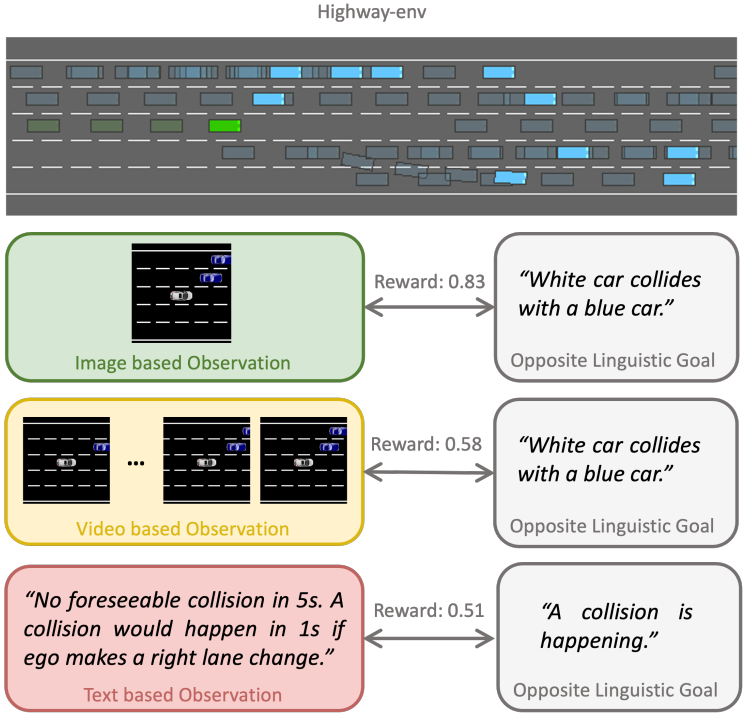


Fig. 6: An example of our image, video and text based observations and the corresponding rewards for a non-collision state.

Fig. 6 and Fig. 7 present the rewards we get from different observations for a non-collision and a collision state respectively. While the reward values are nonidentical across different observations, they are all higher for the non-collision state compared to the collision one. In this way, the ego vehicle can distinguish the dangerous states and learn a safe driving policy.

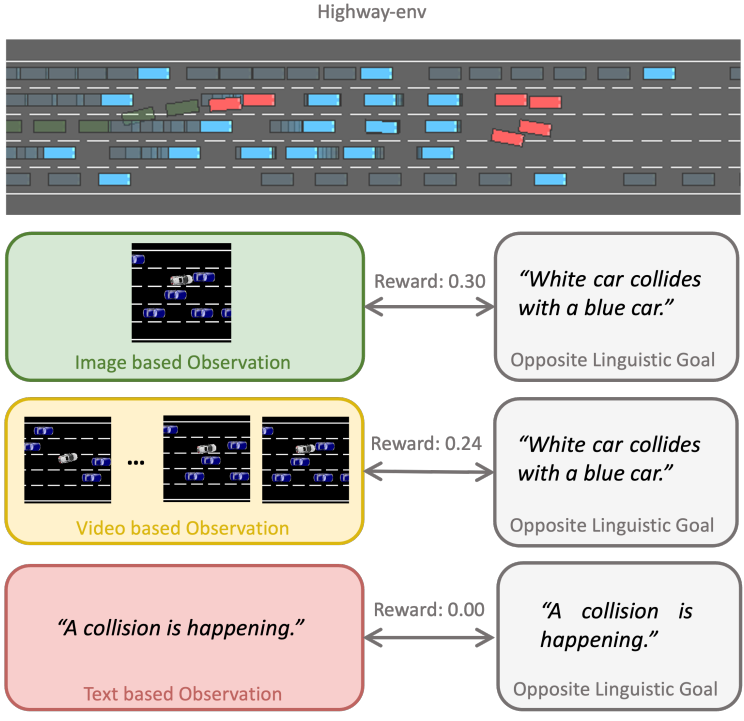
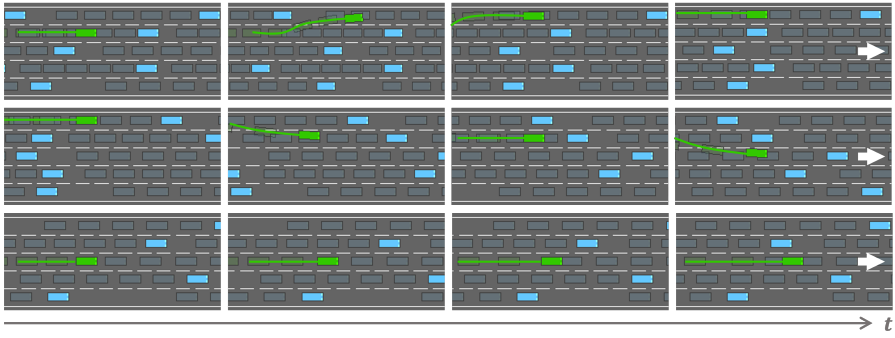


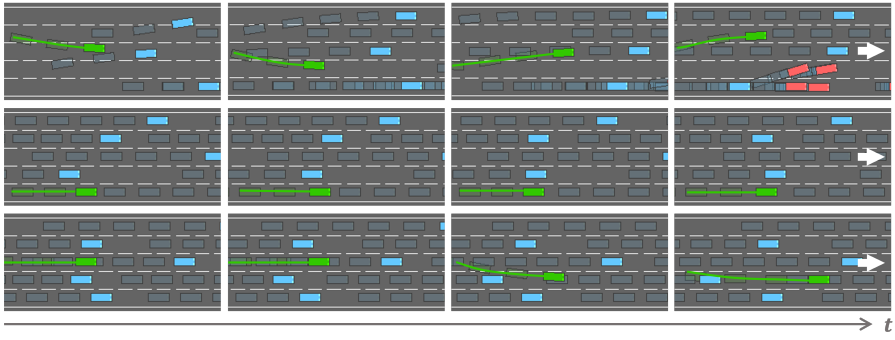
Fig. 7: An example of our image, video and text based observations and the corresponding rewards for a collision state.

B.2 Qualitative Results

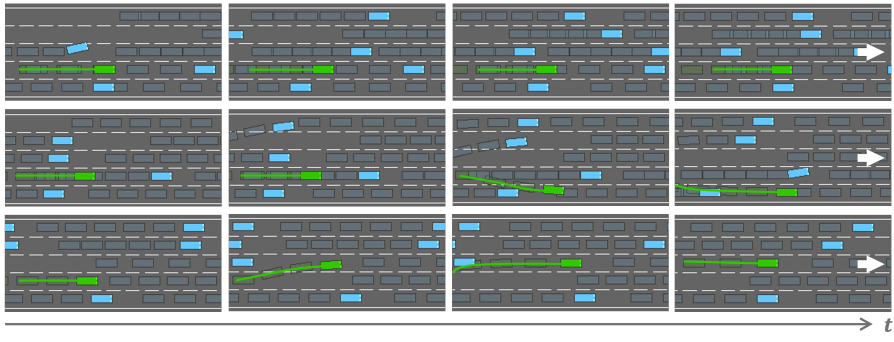
Fig. 8 shows more examples of how the driving policy learned by our LORD with image, video and text based observation performs in the `lane-5-density-3` setting of Highway-env. We can observe that the ego vehicle learns diverse ways to avoid collisions in congested traffic scenarios. The results shall be better viewed in the supplementary videos.



(a) LORD with image based observation.



(b) LORD with video based observation.



(c) LORD with text based observation.

Fig. 8: The driving policy learned by our LORD with image, video and text based observation.