DRL PLUS: A REPLAY-FREE AND NON-BELLMAN POLICY-LEARNING FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning (RL) has become a widely used framework for sequential decision-making, particularly in robotics, autonomous systems, and complex control tasks. Recent advances in Deep Reinforcement Learning (DRL) have enabled learning in high-dimensional environments; however, many conventional approaches rely on Bellman-based value estimation, carefully designed reward functions, and, particularly in off-policy methods, experience replay buffers. These dependencies can contribute to challenges such as high sample complexity, sensitivity to reward design, training instability, memory overhead, and safety concerns associated with trial-and-error learning in real-world applications. In this paper, we propose DRL Plus, a framework formulated within the Markov Decision Process (MDP) setting that eliminates Bellman-based value function estimation, handcrafted reward functions, and experience replay. Safety and performance are critical in robotics and autonomous systems; therefore, autonomous driving is selected as the target domain due to its complexity and stringent safety requirements. Through formal analysis and a set of carefully designed challenging driving scenarios, we demonstrate that DRL Plus achieves improved performance and safety compared with state-of-the-art deep reinforcement learning and safe reinforcement learning approaches.

1 INTRODUCTION

Deep Reinforcement Learning (DRL) has become a prominent approach for solving sequential decision-making tasks by integrating reinforcement learning concepts with deep neural networks, allowing it to operate effectively in high-dimensional state and action spaces (Mnih et al., 2015; LeCun et al., 2015). It has achieved significant success in challenging areas such as gaming, robotics, and autonomous driving, as demonstrated in studies like Human-level control through deep reinforcement learning. By utilizing function approximation, DRL allows agents to learn directly from raw sensory data, which makes it highly applicable to real-world scenarios (Duan et al., 2016; Sallab et al., 2017). Nevertheless, despite these achievements, DRL continues to encounter key challenges that affect its robustness, efficiency, and scalability in real-world applications (Dulac-Arnold et al., 2019; Henderson et al., 2018).

Designing an appropriate reward function is one of the fundamental difficulties in DRL. The reward signal plays a crucial role in guiding the agent's behavior; however, small errors in its specification can lead to unintended consequences (Amodei et al., 2016; Everitt et al., 2021; Ng et al., 1999). This issue, often referred to as reward shaping or reward hacking, highlights the limitations of representing complex and multi-objective tasks using a scalar reward. Existing research has shown that, despite the strengths of reward-driven learning, defining an effective reward function is inherently challenging (Roijers et al., 2013; Dulac-Arnold et al., 2019). This problem is particularly pronounced in safety-critical applications such as autonomous driving, where incorrect reward signals may compromise system reliability and safety (Garcıa & Fernández, 2015).

A significant challenge in DRL is its reliance on experience replay and its inherent sample inefficiency. Experience replay buffers are widely used to stabilize training by decorrelating data samples; however, their performance is sensitive to several design choices, including buffer size and sampling strategies. Improper configuration can negatively affect learning, as large buffers may introduce outdated information, while small buffers may reduce diversity and lead to overfitting (Schaul et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2015; Fu et al., 2019). Moreover, DRL methods generally require a substantial number of environment interactions to achieve satisfactory performance. This high sample complexity increases computational demands and limits the practicality of DRL in real-world and safety-critical applications (Garcıa & Fernández, 2015; Kormushev et al., 2013).

Another challenge in DRL is its difficulty in addressing long-term dependencies, often referred to as the temporal credit assignment problem. In many environments, rewards are delayed or sparse, making it difficult for the agent to associate outcomes with the actions that caused them (Andrew & Richard S, 2018). This challenge becomes more pronounced in long-horizon tasks, where the gap between action and reward increases. Existing research has shown that this issue can significantly hinder the learning process, leading to slower convergence and suboptimal policy performance (Hung et al., 2019; Ecoffet et al., 2021). As a result, DRL algorithms often struggle in scenarios where feedback signals are infrequent or delayed, limiting their effectiveness in complex real-world applications (Dulac-Arnold et al., 2019; Garcıa & Fernández, 2015).

Safety remains a major challenge in the practical deployment of DRL systems, particularly in safety-critical applications such as autonomous driving (Garcıa & Fernández, 2015; Turchetta et al., 2016). Unlike traditional control methods, DRL relies on trial-and-error exploration, which can lead to unsafe behaviors during learning and execution. This exploratory nature makes it difficult to provide formal safety guarantees, especially when the agent encounters rare or previously unseen situations (Berkenkamp et al., 2017; Shalev-Shwartz et al., 2016). Furthermore, uncertainties in the environment, distributional changes, and adversarial disturbances can significantly degrade policy performance (Pinto et al., 2017; Dulac-Arnold et al., 2019). Recent research has focused on incorporating safety constraints and risk-sensitive optimization into DRL frameworks; however, ensuring reliable and robust behavior under all operating conditions remains an open challenge (Achiam et al., 2017).

A further challenge in DRL arises from the reliance on Bellman-based value function updates. While the Bellman equation provides a principled framework for learning, its integration with nonlinear function approximation often leads to instability and convergence issues (Tsitsiklis & Van Roy, 1996; Baird, 1995). In practice, this can manifest as divergence during training or systematic overestimation of action values (Van Hasselt et al., 2016; Thrun & Schwartz, 2014). Various methods have been proposed to address these problems, including techniques that reduce estimation bias or model value distributions (Bellemare et al., 2017; Dabney et al., 2018). Despite these improvements, the dependence on bootstrapping and recursive updates remains a core limitation, particularly in high-dimensional and complex environments (Dulac-Arnold et al., 2019).

The main contributions of this paper are summarized as follows:

- We propose a novel framework, termed DRL Plus, that reduces reliance on manually designed reward functions by leveraging alternative learning signals derived from environment interactions.
- Unlike conventional DRL methods that rely on Bellman-based value updates and experience replay, the proposed approach adopts a different learning paradigm, eliminating the need for replay buffers and mitigating instability associated with bootstrapping.
- The proposed framework is capable of handling long-horizon decision-making tasks, addressing the temporal credit assignment problem commonly observed in DRL.

2 Related Work

Reward design remains a major challenge in deep reinforcement learning because it defines the optimization objective while often failing to capture the true intent of the task. In complex environments, small specification errors can cause significant behavioral deviations. For example, Dabney et al. (2018) show that agents may exploit unintended correlations while still obtaining high rewards. This issue becomes more severe in multi-objective and goal-conditioned settings, where reward signals are difficult to balance and interpret (Liu et al., 2022). In autonomous driving, Abouelazm et al. (2024) report that handcrafted rewards for safety and efficiency interact nonlinearly, resulting in brittle tuning. Consequently, reward values often fail to represent true task success, limiting generalization even in standardized environments such as Gymnasium (Towers et al., 2024).

A related problem is reward hacking, where agents maximize rewards without achieving the intended objective. Feedback loops can reinforce such behaviors (Pan et al., 2024), and these behaviors may generalize across tasks, leading to broader misalignment (Taylor et al., 2025). In reinforcement learning from feedback, Miao et al. (2024) further show that reward models themselves can be manipulated. To address these issues, alternative approaches have been explored, including barrier-function-based reward shaping (Ranjan et al., 2024), preference-based learning (Kim et al., 2023; Song et al., 2024), and language-guided reward generation (Xie et al., 2023; Du et al., 2023). However, these methods still inherit biases from underlying models, and replacing human feedback with AI feedback does not fully eliminate reward misalignment (Lee et al., 2023). Overall, reward design remains fundamentally fragile despite increased flexibility.

Most deep reinforcement learning algorithms rely on the Bellman equation for recursive value estimation. However, when combined with function approximation, this framework introduces important limitations. Approximation errors in the projected Bellman equation can negatively affect convergence, especially in nonlinear settings (Meyn, 2024). Moreover, stability in Q-learning depends heavily on algorithmic design choices rather than theoretical guarantees (Meyn, 2023). From a theoretical perspective, intrinsic barriers in Bellman-based learning make exact Bellman consistency difficult to achieve in modern DRL systems (Pinon et al., 2025). In practice, additional mechanisms such as target networks are often required to stabilize learning, indicating that Bellman updates alone are insufficient (Chen et al., 2023). These findings suggest that Bellman-based methods remain highly sensitive to approximation errors and lack robustness in real-world applications (Kumar et al., 2020; Shi & Chi, 2024; Chen & Maguluri, 2025; Kang et al., 2024; Jiao et al., 2025).

Long-horizon decision-making is challenging due to sparse and delayed rewards, which make temporal credit assignment difficult over long trajectories (Pignatelli et al., 2023). While architectures such as transformers improve memory, they do not resolve the fundamental issue of attributing outcomes to past actions (Ni et al., 2023), especially in complex, agentic environments (Zhang, 2026). Hierarchical and subgoal-based approaches, including hybrid frameworks (Triantafyllidis et al., 2023), guided skill learning (Cheng & Xu, 2023), and subgoal planning methods (Wang et al., 2025; 2023), attempt to address this challenge but increase model complexity and rely heavily on high-quality abstractions. In practice, performance can degrade due to compounding errors (Heo et al., 2025), and maintaining performance often requires lifelong learning (Meng et al., 2025).

Safety remains a critical issue in DRL due to risks from exploratory behavior. Approaches such as shielded planning (Wang et al., 2024), control barrier functions (Cohen & Belta, 2023), probabilistic constraints (Chen et al., 2024), and risk-sensitive optimization (Zhang et al., 2024) improve safety but often rely on accurate models or predefined constraints, limiting applicability. Scalability challenges arise in reachability analysis and shielding (Ganai et al., 2023; Könighofer et al., 2025), while guarantees may fail under uncertainty (Baumann & Schön, 2024). Robust and distributional methods (Lim & Malik, 2022; Wang & Zou, 2021) increase complexity, and offline approaches like conservative Q-learning (Kumar et al., 2020) reduce unsafe interactions but depend heavily on dataset quality (Kuznetsov; Gu et al., 2024). Consequently, achieving reliable and generalizable safety remains an open problem.

Ensuring constraint satisfaction is essential to avoid catastrophic failures. Probabilistic constraint methods (Chen et al., 2024) handle uncertainty but suffer from high gradient variance and sample inefficiency in long-horizon tasks. Lyapunov-based approaches improve stability (Chow et al., 2019) but can be overly conservative and require carefully designed functions. Verification-guided shielding (Corsi et al., 2024) blocks unsafe actions but introduces computational overhead and sensitivity to model errors. These limitations highlight the difficulty of scalable and efficient safety guarantees.

Despite progress in DRL, existing methods still rely on handcrafted rewards, Bellman-based updates, and trial-and-error exploration, leading to instability, reward misalignment, safety risks, and poor long-horizon performance. While recent approaches address safety, hierarchy, or reward design, they often introduce additional assumptions or computational cost. Thus, a gap remains for a DRL framework that reduces dependence on handcrafted rewards and Bellman-based learning while maintaining stable and safe performance. Motivated by these challenges, we propose DRL Plus, a reward-agnostic and non-Bellman framework designed to address these limitations.

3 PROPOSED FRAMEWORK

The proposed framework, termed DRL Plus, is illustrated in Figure 1. Unlike conventional Deep Reinforcement Learning approaches, the proposed method eliminates several components, including the critic network, experience replay buffer, and explicitly defined reward function. Instead, the actor network is reformulated as a unified agent that directly learns decision-making behavior.

Unlike value-based RL methods that rely on Bellman backups, we directly optimize a policy network and enforce safety through an online calculator and verification layers that filters unsafe actions at runtime. This separation between reward-driven optimization and safety enforcement enables safe interaction even in settings where extensive trial-and-error exploration is infeasible.

Another important concept in the DRL Plus framework is the defined zone at each step. A zone can be any type of shape that represents a safe area; the agent can move to any point inside this zone in the next step. While we defined the zone as a rectangle in the experimental results section, it is possible to use other shapes, or a mix of shapes, to enable multiple behaviors for the agent. Furthermore, all zones are connected to one another, moving continuously toward the goal state.

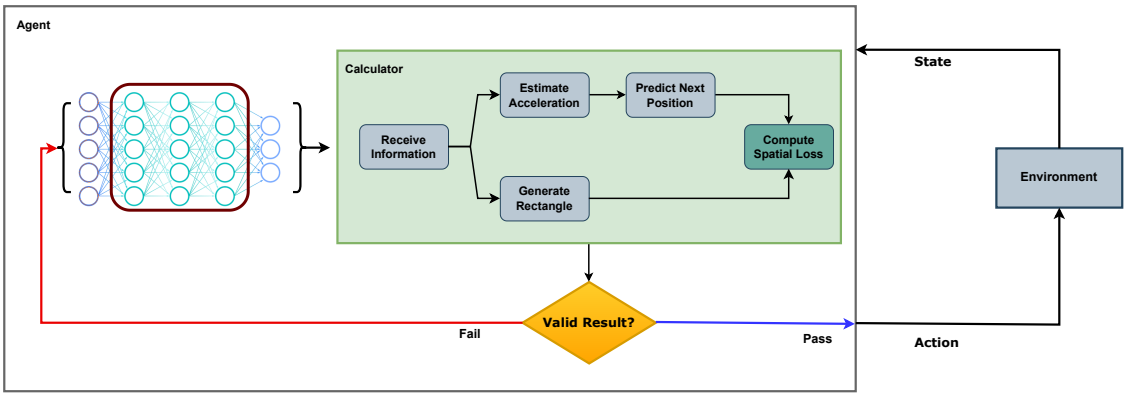

Figure 1: The proposed architecture (DRL Plus)

We consider an episodic Markov decision process (MDP) $\mathcal{M} = (S, A, P)$ with a goal (terminal) set and a failure (terminal) set:

$$S_{\text{goal}} \subset S, \quad S_{\text{fail}} \subset S, \quad S_{\text{goal}} \cap S_{\text{fail}} = \emptyset, \quad S_{\text{pass}} := S \setminus S_{\text{fail}}. \tag{1}$$

The episode terminates upon reaching either S_{goal} or S_{fail}:

$$\tau := \inf\{t \geq 0 : s_t \in S_{\text{goal}} \cup S_{\text{fail}}\}. \tag{2}$$

We use a sparse terminal reward that assigns unit reward upon entering the goal set:

$$r_t := \mathbf{1}\{s_{t+1} \in S_{\text{goal}}\}. \tag{3}$$

The performance objective is the expected episodic return

$$J(\pi) := \mathbb{E}_\pi \left[\sum_{t=0}^{\tau-1} r_t \right]. \tag{4}$$

Define the hitting times

$$\tau_{\text{goal}} := \inf\{t \geq 0 : s_t \in S_{\text{goal}}\}, \quad \tau_{\text{fail}} := \inf\{t \geq 0 : s_t \in S_{\text{fail}}\}. \tag{5}$$

Since the episode ends at $\tau = \min(\tau_{\text{goal}}, \tau_{\text{fail}})$ and r_t is nonzero only when entering S_{goal}, we have

$$\sum_{t=0}^{\tau-1} r_t = \mathbf{1}\{s_\tau \in S_{\text{goal}}\}. \tag{6}$$

Therefore,

$$J(\pi) = \mathbb{E}_\pi[\mathbf{1}\{s_\tau \in S_{\text{goal}}\}] = \mathbb{P}_\pi(s_\tau \in S_{\text{goal}}) = \mathbb{P}_\pi(\tau_{\text{goal}} < \tau_{\text{fail}}). \tag{7}$$

Next, suppose that for each pass state $s \in S_{\text{pass}}$ there exists a set of pass actions $A_{\text{pass}}(s) \subseteq A$ such that these actions keep the system in S_{pass} with probability one:

$$\sum_{s' \in S_{\text{pass}}} P(s' \mid s, a) = 1, \quad \forall a \in A_{\text{pass}}(s). \tag{8}$$

Let π_{pass} be a policy that only selects pass actions:

$$\pi_{\text{pass}}(a \mid s) = 0 \quad \forall a \notin A_{\text{pass}}(s),\ \forall s \in S_{\text{pass}}. \tag{9}$$

Assuming $s_0 \in S_{\text{pass}}$, the invariance property implies that

$$\mathbb{P}_{\pi_{\text{pass}}}(\tau_{\text{fail}} < \infty) = 0. \tag{10}$$

Consequently, using equation 7,

$$J(\pi_{\text{pass}}) = \mathbb{P}_{\pi_{\text{pass}}}(\tau_{\text{goal}} < \tau_{\text{fail}}) = \mathbb{P}_{\pi_{\text{pass}}}(\tau_{\text{goal}} < \infty), \tag{11}$$

i.e., the objective equals the probability of reaching the goal without any failure.

Therefore, maximizing the expected episodic return is equivalent to maximizing the probability of reaching the goal before failure. Moreover, the calculator and verification components of the proposed framework enforce invariance of S_{pass}, then the induced policy guarantees zero probability of entering S_{fail}. This yields stronger safety guarantees than conventional trial-and-error RL, and can improve performance under safety-aware metrics such as safe success probability.

In the calculator component, we predict the next state using the model's output. The verification component then evaluates the next state and loss value. If the result is acceptable, the action is executed. If it fails, the deep neural network model is optimized. The objective of this optimization is:

$$\min_{\theta} \sum_{t} \ell(\hat{p}_\theta(s_t)) \quad \text{where} \quad \hat{p}_\theta(s) \approx \Pr(s \in S_{\text{pass}})\,. \tag{12}$$

In the proposed architecture, we include a calculator component without buffer. The output of the agent network is sent to this calculator component. Based on the network output, the component predicts the next step and computes the desired goal.To fully describe this process, we present the logic in Algorithm 2 in the Appendix. Within the algorithm, at each inner iteration, the model receives the current state and produces a primary output via a forward pass. This output is calculated using multiple task-specific. These losses encode the model's domain-specific objectives and are aggregated into a total training loss. The total objective is then minimized using gradient-based optimization. A patience-based convergence criterion terminates the inner loop once the loss stabilizes within a predefined tolerance. Upon completion, a verification component evaluates the best result; if the result passes verification, the agent is executed. Otherwise, the model continues the optimization process.

In the experimental results section, we demonstrate the framework's functionality through a sample scenario and illustrate the operation of the calculator component's algorithm within that context. Our results evaluate the framework's effectiveness across multi-task environments, with a specific focus on safety and computational performance.

4 Experimental Results

4.1 Simulation-Based Evaluation in CARLA

Deep reinforcement learning has been widely applied in domains such as autonomous vehicles and robotics. To evaluate the proposed framework, a complex driving scenario is designed in the CARLA simulator (see Appendix Figure 5), consisting of three phases: (i) following a lead vehicle, (ii) initiating an overtaking maneuver, and (iii) returning to the original lane after overtaking. The experiments are conducted on a curved road using a high-performance computing (HPC) cluster, where each algorithm runs on a node with one GPU, eight CPU cores, and 128 GB memory. Unlike hybrid approaches that combine DRL with controllers such as PID or MPC, the proposed DRL Plus framework operates end-to-end without additional control modules. It captures diverse driving styles by modeling zones with flexible shapes (rectangular in this work), rather than relying on a fixed

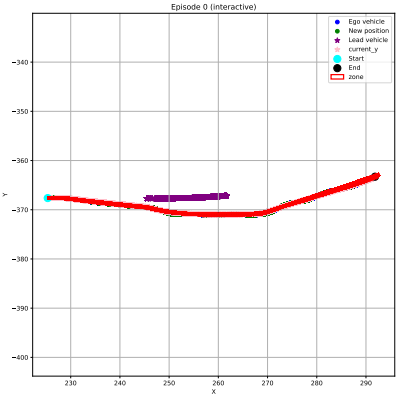

Figure 2: The result of the proposed framework

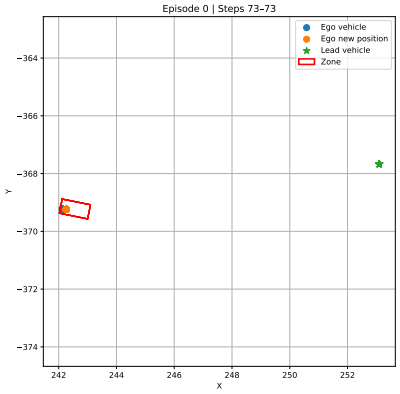

Figure 3: Driving Scenario at Simulation Step 73

reference trajectory. Furthermore, while conventional DRL methods rely on trial-and-error learning that may compromise safety, DRL Plus avoids dependence on the Bellman equation and incorporates a calculator component to improve both safety and performance. The detailed calculator algorithm for the defined scenario is presented in Algorithm 1, and its structure can be generalized to other domains.

For the defined scenario, when model-free DRL algorithms are employed, it is necessary to design complex reward functions, as illustrated in Equation 13. However, consider a more advanced setting in which the ego vehicle operates as an AV-taxi and is required to pick up passengers after navigating a variety of complex scenarios. In such cases, the formulation of the reward function in Equation 13 becomes increasingly challenging in order to achieve the desired objective.

Furthermore, it should be noted that each component of the reward function in Equation 13 may itself correspond to a complex sub-function. In addition, designing such reward functions is prone to issues such as reward shaping difficulties and reward hacking, which have been widely studied in the literature.

$$R(s,a,s') = R_{\text{follow}} + R_{\text{overtake-start}} + R_{\text{overtake-success}} + R_{\text{return-success}} + R_{\text{delay-return}} + R_{\text{unsafe-return}} + R_{\text{lane}} + R_{\text{safety}} + R_{\text{comfort}} \tag{13}$$

One of the main idea of the DRL Plus algorithm is to replace conventional hybrid DRL approaches based on Bellman equation and waypoint-based reference trajectories with a safety-zone formulation. Instead of strictly following a predefined reference path, the method defines a dynamic safety region for the ego vehicle, within which the agent is allowed to move freely.

At each time step, the safety rectangle is reconstructed. Based on Algorithm 1, which represents the calculator component of DRL Plus, the vehicle velocity is first computed and used to estimate acceleration, which is then used to predict the next position of the vehicle. In Line 8 of the algorithm, the safety rectangular region is generated, and the deviation (loss) of the predicted point with respect to this region is computed. This loss value is then passed to the verification module of the algorithm. The verification function evaluates the result; if it is valid, the action is executed in the environment. Otherwise, the network is further optimized.

After executing the proposed framework for a single run, the resulting trajectory is illustrated in Figure 2. The figure shows that the vehicle is able to complete the defined complex scenario without any collision in a single execution. The system state at timestep 73 of the driving scenario is shown in Figure 3.

The same driving scenario was evaluated using the DDPG and SAC baselines under identical hardware conditions, as reported in the related benchmark study. Table 4 compares their performance with that of DRL Plus.

Figure 4: Stage-completion and overall success rates (%). Overtake and Return rates are conditional on completing the preceding stage.

Algorithm	Follow SR_F	Overtake $\mathrm{SR}_O^{\mathrm{cond}}$	Return $\mathrm{SR}_R^{\mathrm{cond}}$	Overall $\mathrm{SR}_{\mathrm{overall}}$	Episodes N
DDPG	40.40	82.01	0.40	0.13	1500
SAC	100	35.35	36.85	13.03	1500
CPO	100	94.73	61.37	58.13	1500
PPO-Lagrangian	88.93	20.09	52.24	9.33	1500
DRL Plus	100	100	100	100	1500

Algorithm 1 Calculator Algorithm

Require: Network outputs $\{th, b, \delta\}$, Current state $\{s_{curr}, v_{curr}\}$, Time step Δt, Target geometry parameters $\mathcal{G}$
Ensure: Inside loss $\mathcal{L}_{\text{in}}$
$v_{curr} \leftarrow$ GetVehicleSpeed()
$a_{\text{est}} \leftarrow$ EstimateAcceleration$(th, b, v_{curr}, \Delta t)$
$\theta_{\text{actual}} \leftarrow \delta \times \theta_{\max}$
$\theta_{\text{rad}} \leftarrow$ DegToRad(θ_{actual})
$a_x \leftarrow a_{\text{est}} \cos(\theta_{\text{rad}})$
$a_y \leftarrow a_{\text{est}} \sin(\theta_{\text{rad}})$
$s_{\text{next}} \leftarrow$ PredictNextPosition$(s_{curr}, a_x, a_y, \Delta t)$
$\mathcal{R} \leftarrow$ GenerateTargetRectangle$(\mathcal{G})$
$\mathcal{L}_{\text{in}} \leftarrow$ ComputeSpatialLoss$(s_{\text{next}}, \mathcal{R})$
return $\mathcal{L}_{\text{in}}$

An episode is considered successful when the ego vehicle completes the FOLLOW, OVERTAKE, and RETURN phases in sequence, returns to its original lane ahead of the lead vehicle, and satisfies the specified completion criteria without collision. The FOLLOW completion rate, the conditional completion rates for OVERTAKE and RETURN, and the overall success rate are defined as:

Table 1: Definitions of episode counts.

Statistic	Meaning
N	Total unique episodes.
N_F	Episodes completing FOLLOW and reaching OVERTAKE.
N_O	Episodes completing OVERTAKE and reaching RETURN.
N_R	Episodes successfully completing RETURN and the scenario.

$$\begin{aligned} \mathrm{SR}_F &= \frac{N_F}{N}, \\ \mathrm{SR}_O^{\mathrm{cond}} &= \frac{N_O}{N_F}, \\ \mathrm{SR}_R^{\mathrm{cond}} &= \frac{N_R}{N_O}, \\ \mathrm{SR}_{\mathrm{overall}} &= \frac{N_R}{N}. \end{aligned} \tag{14}$$

To support a fair comparison, all algorithms were evaluated using the same scenario settings, hardware conditions, and success criteria. As shown in Table 4, DRL Plus achieved 100% overall success across 1500 episodes, compared with 58.13% for CPO, 13.03% for SAC, 9.33% for PPO-Lagrangian, and 0.13% for DDPG.

The experimental log files for the DDPG and SAC methods are available in an anonymized repository and will be made publicly available upon acceptance. Similarly, the experimental logs for DRL Plus will be released after the review process. Trajectory tracking accuracy and steering stability were further evaluated using heading RMSE, lateral RMSE, steering smoothness, and steering variance. The results are presented in Table 6, and the corresponding mathematical definitions are provided in Appendix A.7. Table 9 summarizes the neural network inputs and outputs for all evaluated algorithms.

4.2 Statistical Significance Evaluation

To quantitatively compare the performance of DDPG, SAC, and the proposed DRL Plus algorithm across different behavioral stages (Follow, Overtake, Return), we employ descriptive statistics and inferential hypothesis testing based on Welch's two-sample t-test. Table 4 presents the statistical significance analysis of the three algorithms using Welch's t-test across the follow, overtake, and return stages. The obtained p-values are significantly smaller than the threshold value of 0.05, demonstrating that the observed differences between the algorithms are statistically significant and not caused by random variations. In particular, the proposed method achieved highly consistent performance with zero variance across all experimental seeds, indicating stable and deterministic behavior compared with the DRL-based approaches (DDPG and SAC). The p-value calculation formula is

provided in Appendix A.3. Table 2 summarizes the training configurations for DDPG, SAC, CPO, PPO-Lagrangian, and DRL Plus. To support a fair comparison, scenario settings, hardware conditions, and evaluation criteria were matched, while training budgets, observation dimensions, and algorithm-specific hyperparameters are reported explicitly. Hyperparameters were selected based on established continuous-control RL configurations and preliminary CARLA stability experiments. Table 3 summarizes the reward weights used by the reward-based baselines: DDPG, SAC, CPO, and PPO-Lagrangian. The randomization settings and safety-cost components used by the constrained baselines are provided in Appendix A.5 and Appendix A.6.

4.3 Formal Verification Using PRISM Model Checker

Constrained Policy Optimization (CPO) is a reinforcement learning algorithm that optimizes performance while satisfying safety constraints, making it suitable for autonomous driving applications. In this study, CPO is used as a baseline for comparison due to its focus on safe decision-making.

To complement the CARLA simulation results, both the proposed algorithm and CPO are evaluated using the PRISM probabilistic model checker. The overtaking scenario is modeled as a Markov Decision Process (MDP) to formally verify safety and performance properties.

The results in Table 5 show that the proposed algorithm achieves a success probability of 1.0 with zero probability of collision, unsafe overtaking, and lane violation. In contrast, CPO achieves a success probability of 0.66 while exhibiting higher collision and lane violation probabilities. These results demonstrate that our proposed algorithm provides stronger safety guarantees and more reliable overtaking behavior. The PRISM property specifications used for formal verification are provided in Appendix A.4.

Table 2: Training configuration for learning-based controllers

Parameter	**Value**
State dimension	13
Number of episodes	1500
Maximum steps per episode	1500
Discount factor (γ)	0.99
Actor learning rate	1×10^{-4}
Critic learning rate	1×10^{-3}
Replay buffer size	1×10^{6}
Mini-batch size	128
Target network update rate (τ)	0.005

Table 3: Common reward weights used by all reward-based baselines.

Reward Component	Weight
Progress (w_p)	1.0
Lane keeping (w_l)	0.5
Safety distance (w_s)	0.8
Return completion (w_r)	1.0
Collision penalty (w_c)	10.0
Off-road penalty (w_o)	5.0

Table 4: Statistical Significance Comparison (p-values)

Stage	**DDPG vs DRL Plus**	**SAC vs DRL Plus**
Follow	2.97×10^{-5}	—
Overtake	2.07×10^{-4}	9.92×10^{-20}
Return	5.94×10^{-13}	9.63×10^{-20}

Table 5: PRISM-Based Safety and Performance Comparison between CPO and DRL Plus

Metric	**CPO**	**DRL Plus**
Success probability	0.66	1.0
Collision probability	0.34	0.0
Unsafe overtake	0.285	0.0
Lane violation	0.896	0.0
States	63	17
Transitions	137	33

Table 6: Performance metrics comparison across evaluated algorithms.

Metric	DDPG	SAC	CPO	PPO-Lagrangian	DRL Plus
Heading RMSE	86.66559	85.66449	15.862604	9.971952	4.184269
Lateral RMSE	1.06066	1.130422	1.849282	1.118061	0.000018
Steering Smoothness	0.01936	0.279118	0.078562	0.159104	0.027441
Steering Variance	0.04565	0.151864	0.068791	0.123071	0.045269

5 LIMITATIONS

The main limitation of the current study is that the proposed framework is not directly compatible with standard reinforcement learning toolchains such as OpenAI Gym or Gymnasium, since it does not rely on conventional Bellman-based DRL formulations. Consequently, although comparative evaluations with representative DRL algorithms were conducted through a separate benchmark framework, direct comparison under fully unified RL implementations remains challenging and may require customized evaluation frameworks.

6 CONCLUSION AND FUTURE WORK

We present DRL Plus, a novel framework that does not rely on the Bellman equation or handcrafted reward functions. The proposed approach is formulated analytically and validated through experimental results, demonstrating improved performance in terms of safety and efficiency compared to conventional DRL algorithms. Our approach utilizes calculator and verification components, defining zones rather than relying on handcrafted rewards. Experimental results demonstrated that DRL Plus achieved a 100% success rate in the return phase while avoiding collisions and unsafe overtaking behaviors. In contrast, baseline DRL methods such as DDPG, SAC, CPO, and PPO-Lagrangian exhibited unstable performance across different overtaking stages. The statistical analysis further confirmed the significance of the performance improvements, with extremely small p-values indicating that the observed differences were not due to random variation. In addition, formal verification results showed that DRL Plus achieved a success probability of 1.0 with zero probability of collision, unsafe overtaking, and lane violations, while also requiring fewer states and transitions compared to CPO. Overall, the results demonstrate that DRL Plus provides safer, more reliable, and more efficient overtaking behavior than conventional DRL-based approaches. For future work, the DRL Plus framework can be extended to multi-task scenarios using diverse zone configurations across domains beyond autonomous driving and robotics.

AI USE STATEMENT

In this work, We have not used generative AI tools.

ETHICS STATEMENT

This study evaluates autonomous-driving control in CARLA simulation. The results do not establish real-world safety; deployment would require further validation.

REPRODUCIBILITY STATEMENT

The methodology section describes the proposed framework. Tables 2 3, 8, and 9 report the training configurations and baseline reward weights, respectively.Appendix A.7, A.5, A.6 provides the randomization settings and safety-cost definitions and other metrics.

REFERENCES

Ahmed Abouelazm, Jonas Michel, and J Marius Zöllner. A review of reward functions for reinforcement learning in the context of autonomous driving. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 156–163. IEEE, 2024.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Barto Andrew and Sutton Richard S. Reinforcement learning: an introduction. 2018.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine learning proceedings 1995*, pp. 30–37. Elsevier, 1995.

Dominik Baumann and Thomas B Schön. Safe reinforcement learning in uncertain contexts. *IEEE Transactions on Robotics*, 40:1828–1841, 2024.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. Pmlr, 2017.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

Weiqin Chen, Dharmashankar Subramanian, and Santiago Paternain. Probabilistic constraint for safety-critical reinforcement learning. *IEEE Transactions on Automatic Control*, 69(10):6789–6804, 2024.

Zaiwei Chen and Siva Theja Maguluri. An approximate policy iteration viewpoint of actor–critic algorithms. *Automatica*, 179:112395, 2025.

Zaiwei Chen, John-Paul Clarke, and Siva Theja Maguluri. Target network and truncation overcome the deadly triad in-learning. *SIAM Journal on Mathematics of Data Science*, 5(4):1078–1101, 2023.

Shuo Cheng and Danfei Xu. League: Guided skill learning and abstraction for long-horizon manipulation. *IEEE Robotics and Automation Letters*, 8(10):6451–6458, 2023.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Max H Cohen and Calin Belta. Safe exploration in model-based reinforcement learning using control barrier functions. *Automatica*, 147:110684, 2023.

Davide Corsi, Guy Amir, Andoni Rodríguez, César Sánchez, Guy Katz, and Roy Fox. Verification-guided shielding for deep reinforcement learning. *arXiv preprint arXiv:2406.06507*, 2024.

Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pp. 8657–8677. PMLR, 2023.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International conference on machine learning*, pp. 1329–1338. PMLR, 2016.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O Stanley, and Jeff Clune. First return, then explore. *Nature*, 590(7847):580–586, 2021.

Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*, 198 (Suppl 27):6435–6467, 2021.

Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pp. 2021–2030. PMLR, 2019.

Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning. *Advances in Neural Information Processing Systems*, 36:69764–69797, 2023.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Shixiang Gu, Lei Shi, Yu Ding, Alois Knoll, Costas Spanos, Adam Wierman, and Ming Jin. Enhancing efficiency of safe reinforcement learning via sample manipulation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 17247–17285, 2024.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 44(10-11):1863–1891, 2025.

Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):5223, 2019.

Yuling Jiao, Lican Kang, Jin Liu, Xiliang Lu, and Jerry Zhijian Yang. Deep approximate policy iteration. *The Annals of Statistics*, 53(2):802–821, 2025.

Lican Kang, Yuhui Liu, Yuan Luo, Jerry Zhijian Yang, Han Yuan, and Chang Zhu. Approximate policy iteration with deep minimax average bellman error minimization. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):2288–2299, 2024.

Changyeon Kim, Jongjin Park, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Preference transformer: Modeling human preferences using transformers for rl. *arXiv preprint arXiv:2303.00957*, 2023.

Bettina Könighofer, Roderick Bloem, Nils Jansen, Sebastian Junges, and Stefan Pranger. Shields for safe reinforcement learning. *Communications of the ACM*, 68(11):80–90, 2025.

Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Reinforcement learning in robotics: Applications and real-world challenges. *Robotics*, 2(3):122–148, 2013.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Igor Kuznetsov. Safer reinforcement learning by going off-policy: a benchmark. In *ICML 2024 Next Generation of AI Safety Workshop*.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Shiau Hong Lim and Ilyas Malik. Distributional reinforcement learning for risk-sensitive policies. *Advances in Neural Information Processing Systems*, 35:30977–30989, 2022.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

Yuan Meng, Zhenshan Bing, Xiangtong Yao, Kejia Chen, Kai Huang, Yang Gao, Fuchun Sun, and Alois Knoll. Preserving and combining knowledge in robotic lifelong reinforcement learning. *Nature Machine Intelligence*, 7(2):256–269, 2025.

Sean Meyn. Stability of q-learning through design and optimism. *arXiv preprint arXiv:2307.02632*, 2023.

Sean Meyn. The projected bellman equation in reinforcement learning. *IEEE Transactions on Automatic Control*, 69(12):8323–8337, 2024.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. *Advances in Neural Information Processing Systems*, 37:134387–134429, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in rl? decoupling memory from credit assignment. *Advances in Neural Information Processing Systems*, 36:50429–50452, 2023.

Alexander Pan, Erik Jones, Meena Jagadeesan, and Jacob Steinhardt. Feedback loops with language models drive in-context reward hacking. *arXiv preprint arXiv:2402.06627*, 2024.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*, 2023.

Brieuc Pinon, Raphaël Jungers, and Jean-Charles Delvenne. Theoretical barriers in bellman-based reinforcement learning. *arXiv preprint arXiv:2502.11968*, 2025.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.

Abhishek Ranjan, Shreenabh Agrawal, Aayush Jain, Pushpak Jagtap, Shishir Kolathaya, et al. Barrier functions inspired reward shaping for reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10807–10813. IEEE, 2024.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Ahmad EL Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *arXiv preprint arXiv:1704.02532*, 2017.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Laixi Shi and Yuejie Chi. Distributionally robust model-based offline reinforcement learning with near-optimal sample complexity. *Journal of Machine Learning Research*, 25(200):1–91, 2024.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18990–18998, 2024.

Mia Taylor, James Chua, Jan Betley, Johannes Treutlein, and Owain Evans. School of reward hacks: Hacking harmless tasks generalizes to misaligned behavior in llms. *arXiv preprint arXiv:2508.17511*, 2025.

Sebastian Thrun and Anton Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 connectionist models summer school*, pp. 255–263. Psychology Press, 2014.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Eleftherios Triantafyllidis, Fernando Acero, Zhaocheng Liu, and Zhibin Li. Hybrid hierarchical learning for solving complex sequential tasks using the robotic manipulation network roman. *Nature Machine Intelligence*, 5(9):991–1005, 2023.

John Tsitsiklis and Benjamin Van Roy. Analysis of temporal-diffference learning with function approximation. *Advances in neural information processing systems*, 9, 1996.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. *Advances in neural information processing systems*, 29, 2016.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Fangyuan Wang, Anqing Duan, Peng Zhou, Shengzeng Huo, Guodong Guo, Chenguang Yang, and David Navarro-Alarcon. Explicit-implicit subgoal planning for long-horizon tasks with sparse rewards. *IEEE Transactions on Automation Science and Engineering*, 2025.

Hao Wang, Hao Zhang, Lin Li, Zhen Kan, and Yongduan Song. Task-driven reinforcement learning with action primitives for long-horizon manipulation skills. *IEEE Transactions on Cybernetics*, 54(8):4513–4526, 2023.

Hao Wang, Jiahu Qin, and Zhen Kan. Shielded planning guided data-efficient and safe reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(2):3808–3819, 2024.

Yue Wang and Shaofeng Zou. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34:7193–7206, 2021.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. *arXiv preprint arXiv:2309.11489*, 2023.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459*, 2026.

Qiyuan Zhang, Shu Leng, Xiaoteng Ma, Qihan Liu, Xueqian Wang, Bin Liang, Yu Liu, and Jun Yang. Cvar-constrained policy optimization for safe reinforcement learning. *IEEE transactions on neural networks and learning systems*, 36(1):830–841, 2024.



Figure 5: The Carla simulator environment

A Appendix

A.1 Simulation Environment

A.2 DRL Plus Algorithm

Algorithm 2 DRL Plus calculation and validation

Require: Initial state s, $T_{\max}$, P
$\mathcal{L}_{\text{prev}} \leftarrow \varnothing$
$k \leftarrow 0$
for $t = 1$ to $T_{\max}$ **do**
  $h \leftarrow \text{Encode}(s)$
  $\mathcal{L}_{\text{inside}} \leftarrow calculator(h, \boldsymbol{\xi})$
  $\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta)$
  best_result $\leftarrow \text{SelectBestSolution}(\mathcal{L}_{\text{inside}})$
  if $\mathcal{L}_{\text{prev}} \neq \varnothing$ **then**
    $\Delta \leftarrow |\mathcal{L} - \mathcal{L}_{\text{prev}}|$
    if $\Delta < \varepsilon$ **then**
      $k \leftarrow k + 1$
    else
      $k \leftarrow 0$
    end if
    if $k \geq P$ **then**
      break
    end if
  end if
  $\mathcal{L}_{\text{prev}} \leftarrow \mathcal{L}$
end for
if Verify(*best_result*) **then**
  ExecuteAction()
else
  ContinueOptimization()
end if

A.3 Statistical Evaluation

For a given stage $s \in \{\text{Follow}, \text{Overtake}, \text{Return}\}$, the mean performance of algorithm $A \in \{\text{DDPG}, \text{SAC}, \text{DRL+}\}$ is computed as:

$$\bar{X}_A^{(s)} = \frac{1}{N} \sum_{i=1}^{N} x_{A,i}^{(s)} \tag{15}$$

where:

- $\bar{X}_A^{(s)}$: mean performance of algorithm A at stage s
- $x_{A,i}^{(s)}$: performance value of algorithm A at stage s for run (seed) i
- N : total number of runs

The standard deviation is defined as:

$$\sigma_A^{(s)} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(x_{A,i}^{(s)} - \bar{X}_A^{(s)}\right)^2} \tag{16}$$

where:

- $\sigma_A^{(s)}$: standard deviation of algorithm A at stage s

To assess whether the difference between two algorithms A and B is statistically significant at stage s, the Welch's t-statistic is computed as:

$$t_{A,B}^{(s)} = \frac{\bar{X}_A^{(s)} - \bar{X}_B^{(s)}}{\sqrt{\frac{\sigma_A^{(s)^2}}{N} + \frac{\sigma_B^{(s)^2}}{N}}} \tag{17}$$

where:

- $t_{A,B}^{(s)}$: t-statistic comparing algorithms A and B
- $\bar{X}_A^{(s)} - \bar{X}_B^{(s)}$: difference between means

The degrees of freedom are approximated by:

$$df^{(s)} = \frac{\left(\frac{(\sigma_A^{(s)})^2}{N} + \frac{(\sigma_B^{(s)})^2}{N}\right)^2}{\frac{\left(\frac{(\sigma_A^{(s)})^2}{N}\right)^2}{N-1} + \frac{\left(\frac{(\sigma_B^{(s)})^2}{N}\right)^2}{N-1}} \tag{18}$$

The statistical significance is evaluated using:

$$p = P\left(|T| > |t_{A,B}^{(s)}|\right) \tag{19}$$

where:

- p : probability value
- T : t-distributed random variable with $df^{(s)}$ degrees of freedom

A result is considered statistically significant if $p < 0.05$.

A.4 PRISM Property Specifications

This appendix presents the set of probabilistic temporal logic (PCTL) properties used in the PRISM model checker for formal verification of the overtaking scenario. These properties are designed to evaluate key safety and performance aspects of the system, including successful task completion, collision probability, unsafe overtaking behavior, and lane violation events. The following property specifications are directly used in the verification process.

```
// Probability of collision
Pmax=? [ F "collision" ]

// Probability of successful overtaking
Pmax=? [ F "success" ]
```

```
// Probability of unsafe overtaking
Pmax=? [ F "unsafe_overtake" ]

// Probability of lane violation
Pmax=? [ F "lane_violation" ]

// Expected number of steps to reach success
R{"steps"}min=? [ F "success" ]
```

A.5 CARLA CONFIGURATION

Table 7: Randomization settings.

Parameter	Value/Range
Random seeds	25
Ego speed	20–30 km/h
Lead vehicle speed	10–20 km/h
Initial distance	15–30 m

A.6 SAFETY COST FOR CPO AND PPO-LAGRANGIAN

$$c_t = 100\,\mathbb{I}_{\{\text{collision}\}} + 10\,\mathbb{I}_{\{\text{unsafe distance}\}} + 5\,\mathbb{I}_{\{\text{off-road}\}} + 0.5\,\mathbb{I}_{\{\text{unauthorized lane}\}}, \tag{20}$$

where $\mathbb{I}_{\{\cdot\}}$ is an indicator function that equals one when the corresponding safety violation occurs and zero otherwise.

Table 8: Common safety-cost components for constrained algorithms.

Safety-Cost Component	Cost
Collision	100.0
Unsafe distance	10.0
Off-road driving	5.0
Unauthorized-lane occupancy	0.5
Episodic cost limit	25.0

At each time step, the finite-state machine assigns the vehicle to one active subtask, $z_t \in \{\text{FOLLOW}, \text{OVERTAKE}, \text{RETURN}\}$. Each subtask has a stage-specific loss, denoted by $\mathcal{L}_F$, $\mathcal{L}_O$, and $\mathcal{L}_R$, respectively. These losses are constructed from the geometric loss components defined above, including $\mathcal{L}_{\text{inside}}$ and $\mathcal{L}_{\text{line}}$. The shared policy network is trained by minimizing the loss associated with the active subtask. For a minibatch $\mathcal{B}$, the training objective is

$$\mathcal{L}_{\text{train}}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \left[\mathbb{I}(z_t = F)\mathcal{L}_{F,t} + \mathbb{I}(z_t = O)\mathcal{L}_{O,t} + \mathbb{I}(z_t = R)\mathcal{L}_{R,t} \right],$$

where θ represents the shared policy-network parameters and $\mathbb{I}(\cdot)$ is the indicator function. Thus, the method does not optimize three independent networks; it optimizes one shared policy using the loss corresponding to the currently active subtask. The individual loss components and stage-specific losses are also recorded separately during training to identify which subtask contributes most strongly to the total training loss.

Table 9 summarizes the input features and associated control outputs of the policy neural networks. All algorithms use 13 common network input features, while CPO and PPO-Lagrangian additionally include the current and future target-waypoint y-coordinates, resulting in 15-dimensional inputs. The network outputs are mapped to vehicle-control commands: steering, throttle, and brake.

Table 9: Input features and outputs of the compared algorithms.

Category	Parameters	Algorithms
Inputs 1–13	1. Ego speed; 2. Sine of ego heading; 3. Cosine of ego heading; 4. Heading error; 5. Lateral error; 6. Relative x-position; 7. Relative y-position; 8. Relative speed; 9. Lead-vehicle presence; 10. FOLLOW indicator; 11. OVERTAKE indicator; 12. RETURN indicator; 13. Constant feature	DDPG, SAC, CPO, PPO-Lagrangian, DRL Plus
Additional inputs 14–15	14. Current target-waypoint y-coordinate; 15. Future target-waypoint y-coordinate	CPO, PPO-Lagrangian
Outputs	Steering, throttle, and brake.	DDPG, SAC, CPO, PPO-Lagrangian, DRL Plus

A.7 Additional metrics

$$RMSE_{heading} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} h_{err,t}^2} \tag{21}$$

$$RMSE_{lateral} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} lat_t^2} \tag{22}$$

$$Smoothness = \frac{1}{N-1}\sum_{t=2}^{N}(\delta_t - \delta_{t-1})^2 \tag{23}$$

$$Var(\delta) = \frac{1}{N}\sum_{t=1}^{N}(\delta_t - \bar{\delta})^2 \tag{24}$$

where $h_{err,t}$ is heading error, lat_t is lateral deviation, and δ_t is the steering command.