

Review

Deep Reinforcement and IL for Autonomous Driving: A Review in the CARLA Simulation Environment

Piotr Czechowski ^{1,†} , Bartosz Kawa ^{1,†}, Mustafa Sakhai ^{1,*,†}  and Maciej Wielgosz ^{1,2} 

¹ Faculty of Computer Science, Electronics and Telecommunications, AGH University of Science and Technology, 30-059 Krakow, Poland; pczechow@agh.edu.pl (P.C.); bartekkawa@student.agh.edu.pl (B.K.); wielgosz@agh.edu.pl (M.W.)

² Academic Computer Centre AGH, 30-950 Krakow, Poland

* Correspondence: msakhai@agh.edu.pl

† These authors contributed equally to this work.

Abstract

Autonomous driving is a complex and fast-evolving domain at the intersection of robotics, machine learning, and control systems. This paper provides a systematic review of recent developments in reinforcement learning (RL) and imitation learning (IL) approaches for autonomous vehicle control, with a dedicated focus on the CARLA simulator, an open-source, high-fidelity platform that has become a standard for learning-based autonomous vehicle (AV) research. We analyze RL-based and IL-based studies, extracting and comparing their formulations of state, action, and reward spaces. Special attention is given to the design of reward functions, control architectures, and integration pipelines. Comparative graphs and diagrams illustrate performance trade-offs. We further highlight gaps in generalization to real-world driving scenarios, robustness under dynamic environments, and scalability of agent architectures. Despite rapid progress, existing autonomous driving systems exhibit significant limitations. For instance, studies show that end-to-end reinforcement learning (RL) models can suffer from performance degradation of up to 35% when exposed to unseen weather or town conditions, and imitation learning (IL) agents trained solely on expert demonstrations exhibit up to 40% higher collision rates in novel environments. Furthermore, reward misspecification remains a critical issue—over 20% of reported failures in simulated environments stem from poorly calibrated reward signals. Generalization gaps, especially in RL, also manifest in task-specific overfitting, with agents failing up to 60% of the time when faced with dynamic obstacles not encountered during training. These persistent shortcomings underscore the need for more robust and sample-efficient learning strategies. Finally, we discuss hybrid paradigms that integrate IL and RL, such as Generative Adversarial IL, and propose future research directions.

Keywords: autonomous driving; reinforcement learning; imitation learning



check for updates

Academic Editors: Jalil Boudjadar and Yutaka Ishibashi

Received: 25 May 2025

Revised: 31 July 2025

Accepted: 9 August 2025

Published: 14 August 2025

Citation: Czechowski, P.; Kawa, B.; Sakhai, M.; Wielgosz, M. Deep Reinforcement and IL for

Autonomous Driving: A Review in the CARLA Simulation Environment.

Appl. Sci. **2025**, *15*, 8972. <https://doi.org/10.3390/app15168972>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, autonomous driving has evolved into a complex and rapidly progressing research area. As an inherently multidisciplinary challenge, it brings together elements of robotics, computer vision, control theory, embedded systems, and artificial intelligence. While many architectural decompositions are possible, for clarity, we distinguish five core functional domains in autonomous vehicles: perception, localization, planning, control, and system-level management (see Figure 1). These components do not operate in isolation—they are highly interdependent and must be tightly integrated to achieve safe

and reliable autonomous behavior. For instance, accurate localization relies on perception, planning depends on both localization and perception, and control must faithfully execute decisions made by the planning module. This functional breakdown serves as a foundation for understanding where and how RL and IL techniques can be applied effectively.

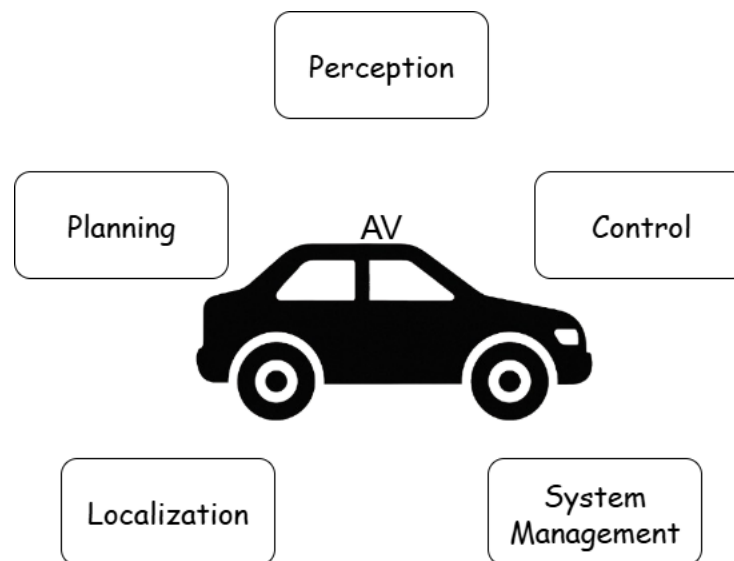


Figure 1. Functional components of autonomous driving.

Perception in autonomous vehicles encompasses the capabilities that allow the system to interpret its surroundings using raw sensor data from cameras, LiDAR, radar, and ultrasonic sensors. It processes this input to detect and classify objects, recognize lanes and traffic signals, and assess environmental conditions such as lighting and weather. The output is a structured, machine-readable representation of the environment, which serves as a vital input to the localization, planning, and decision-making modules. To be effective, perception must operate in real time and remain robust under diverse and challenging conditions, including occlusions, sensor noise, and adverse weather. Localization enables the vehicle to determine its precise position and orientation within a known or partially known environment by fusing data from GPS, inertial measurement units, LiDAR, and cameras, often using techniques such as simultaneous localization and mapping (SLAM) or map-matching algorithms. Accurate localization is essential for safe navigation and must remain reliable even in GPS-denied areas, during sensor drift, or when landmarks are temporarily obscured. Planning is responsible for determining the vehicle's future actions and is typically divided into two levels: behavior planning, which selects high-level maneuvers such as lane changes or yielding, and motion planning, which computes detailed, dynamically feasible trajectories. These plans depend heavily on accurate inputs from perception and localization to ensure obstacle avoidance, compliance with traffic laws, and passenger comfort, while adapting continuously to a changing environment. Control translates these planned trajectories into low-level commands for steering, throttle, and braking, aiming to follow the desired path with precision, stability, and responsiveness. It commonly employs methods such as PID controllers, model predictive control (MPC), or learning-based strategies, and must operate at high frequency while being resilient to delays, disturbances, and variations in vehicle dynamics. Overseeing all these components is the system management layer, which ensures reliable and safe operation by coordinating tasks, allocating computational resources, monitoring system health, detecting faults, and managing communication between modules. It also provides fallback strategies in case of failures, making it a critical backbone for production-grade autonomous driving systems.

This work focuses primarily on the planning and control components of autonomous driving, where RL and IL have demonstrated the greatest practical relevance. In these domains, learning-based agents are trained to make tactical decisions and execute low-level maneuvers based on sensor-derived representations of the environment. Approaches vary from end-to-end policies that directly map observations to actions, to modular designs where planning and control are learned separately. In contrast, components such as perception and localization are typically treated as fixed inputs—often provided by the simulator in the form of semantic maps, LiDAR projections, or ground-truth poses—and are not the target of learning. Similarly, system-level management is generally handled using conventional software architectures and falls outside the scope of learning.

Among the range of machine learning techniques applied in this domain, reinforcement learning (RL)—exemplified by algorithms such as Proximal Policy Optimization (PPO) and Deep Q Networks (DQNs)—has emerged as one of the most promising approaches [1]. Consequently, the field of RL continues to advance at a remarkable pace, with increasingly sophisticated methods producing superior performance across a growing spectrum of applications [2,3].

This review addresses RL and IL strategies implemented in the CARLA simulation platform. CARLA is uniquely suited to this task due to its open-source nature, full-stack simulation capabilities, and strong support for AI integration, making it one of the most widely used platforms for academic research on learning-based AV agents. CARLA is a widely adopted high-fidelity simulator for autonomous driving research that offers extensive support for sensor realism, weather variability, and urban driving scenarios [4,5]. Reviewing the state-of-the-art literature helps identify prevailing agent architectures, design decisions, and gaps that may inform the development of improved driving agents.

The technical analysis in this paper is grounded in representative RL-based studies [6–11]. For each work, we extract the definitions of the agent’s state and action spaces and examine the structure and effectiveness of the associated reward function. The studies, though diverse in their technical formulations, share a common foundation: the use of DRL in the CARLA environment for autonomous navigation and control. We synthesize the insights from these papers to compare strategies, identify trade-offs, and understand how varying model architectures impact policy learning and performance.

Additionally, three studies [12–14] present IL-based approaches that do not rely on environmental rewards but instead learn behaviors by mimicking expert demonstrations. These IL methods are assessed in parallel to the RL models to identify complementary strengths and limitations. Their inclusion is critical, as IL is increasingly integrated with RL to enhance learning efficiency, policy generalization, and safety, especially in data-constrained or high-risk environments [15,16].

Beyond these core studies, recent research has pushed the boundaries of RL in autonomous driving further. For example, Sakhai and Wielgosz proposed an end-to-end escape framework for complex urban settings using RL [17], while Kołomański et al. extended the paradigm to pursuit-based driving, emphasizing policy adaptability in adversarial contexts [18]. Furthermore, Sakhai et al. explored biologically inspired neural models for real-time pedestrian detection using spiking neural networks and dynamic vision sensors in simulated adverse weather conditions [19]. Recent studies have also explored the robustness of AV sensor systems against cyber threats. Notably, Sakhai et al. conducted a comprehensive evaluation of RGB cameras and Dynamic Vision Sensors (DVSs) within the CARLA simulator, demonstrating that DVSs exhibit enhanced resilience to various cyberattack vectors compared to traditional RGB sensors [20]. These contributions showcase the evolving versatility of RL architectures and their capacity to address safety-critical tasks under uncertainty.

Despite the rapid progress in learning-based autonomous driving, existing systems face several persistent challenges. Studies have shown that end-to-end RL agents often suffer from significant performance degradation—up to 35%—when deployed in conditions that differ from their training environments, such as novel towns or adverse weather [9]. Transfer RL has been explored as a mitigation for such weather-induced distribution shifts [21]. Similarly, imitation learning (IL) agents trained solely on expert demonstrations have demonstrated up to 40% higher collision rates when exposed to previously unseen driving contexts [12]. Moreover, a key difficulty in RL lies in the careful tuning of reward functions; poorly calibrated rewards have been linked to over 20% of policy failures in simulation settings [7,11]. Finally, RL agents often overfit to narrow task domains, with generalization failures reaching up to 60% when encountering dynamic obstacles not seen during training [10]. These limitations underscore the need for more robust, transferable, and sample-efficient learning frameworks that integrate the strengths of both RL and IL paradigms.

In general, this review aims to provide a comprehensive synthesis of RL and IL research in the CARLA simulator, highlighting how these learning paradigms contribute to the development of intelligent autonomous driving systems. Compared to widely cited reviews such as [22,23], our work provides a more detailed and structured analysis of reinforcement learning methods specifically in the context of autonomous driving control within simulation environments. While previous surveys focus largely on algorithmic taxonomies or general architectural roles, we examine RL approaches through a set of concrete implementation descriptors: state representation, action space, and reward design. This framing enables direct comparison of control strategies and learning objectives across studies, which is largely missing from earlier work. We also include visual summaries of reward functions and highlight differences in how these functions guide policy learning. Furthermore, our focus on the CARLA simulator as a common experimental platform allows for a more consistent and grounded discussion of evaluation strategies and training setups, bridging the gap between theory and practical deployment.

2. Reinforcement Learning in Autonomous Driving

2.1. Overview of Reinforcement Learning

Reinforcement learning (RL) is a cornerstone of modern machine learning, particularly as deep learning has expanded its applicability to complex, real-world problems [24]. In the RL paradigm, an agent interacts with its environment by performing actions and receiving feedback in the form of rewards, which guide its future behavior [25–27]. This closed-loop interaction between agent and environment lies at the heart of autonomous decision-making systems.

During training, the agent uses a policy to select actions, while the environment responds by returning a new observation and a scalar reward that reflects the value of the action taken [28]. The reward function is pivotal—it defines the learning objective and directly influences the agent's trajectory through the learning space [29,30]. A schematic of this interaction is shown in Figure 2, where the feedback loop is emphasized. A key strength of RL lies in its emphasis on long-term cumulative reward optimization [31]. Instead of maximizing immediate payoff, the agent learns policies that favor long-term outcomes, often sacrificing short-term gain for strategic advantages. Each learning episode starts in an initial state and ends upon reaching a terminal condition [32], making RL particularly suitable for sequential decision-making tasks such as autonomous driving.

In this framework, the **agent** serves as the autonomous controller, making decisions from within a predefined **action space** [33]. These actions may be discrete (e.g., lane switch) or continuous (e.g., steering angle adjustment) [34,35]. A **state** encodes relevant environmental information at a given moment, and the complete collection of possible

states defines the **state space** [36]. The action space defines all possible maneuvers the agent can execute [37].

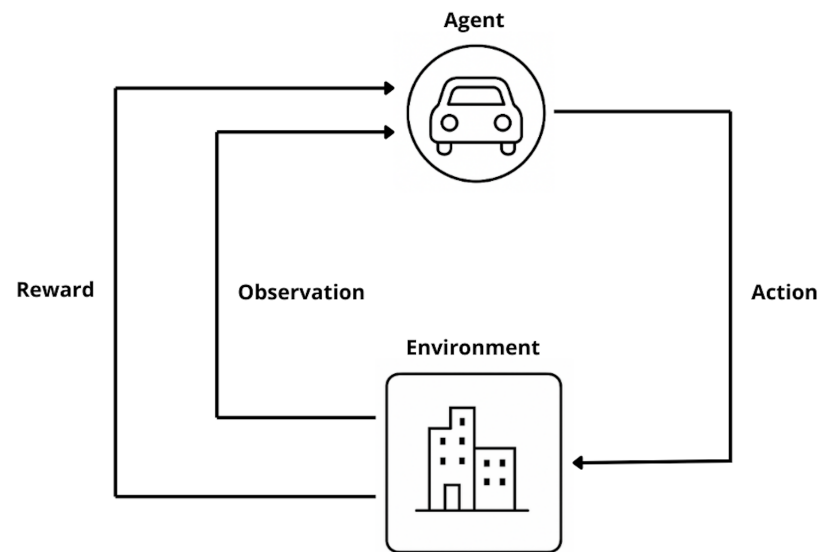


Figure 2. Generalized architecture of a reinforcement learning system, showing the agent–environment interaction cycle.

A crucial function in this setup is the **reward function**, which provides evaluative feedback for the agent’s choices. This scalar signal not only reinforces beneficial actions but also penalizes poor decisions, shaping the agent’s future policy [29,30]. The **policy** maps states to probabilities over actions, and the agent’s objective is to approximate or discover an **optimal policy** that maximizes expected cumulative rewards across episodes [38].

RL is distinct from supervised and unsupervised learning, the other two main branches of machine learning [39]. While supervised learning relies on labeled data and unsupervised learning discovers structure from unlabeled data, RL learns directly through real-time interaction. The episodic nature of RL generates diverse, temporally rich training samples that improve generalization and robustness in dynamic settings.

Reinforcement learning methodologies are commonly categorized into the following types:

- **Model-based vs. Model-free:** Model-based methods construct a predictive model of environment dynamics to plan actions. Model-free methods, in contrast, learn policies or value functions directly from experience [38,40,41].
- **Policy-based vs. Value-based:** Policy-based methods directly learn the policy function, often using gradient methods. Value-based methods estimate action values and derive policies by selecting actions with the highest value [38,42].
- **On-policy vs. Off-policy:** On-policy methods optimize the same policy used to generate data, ensuring data-policy alignment. Off-policy methods allow learning from historical or exploratory policies, improving sample efficiency [38,43,44].

Recent studies have explored diverse reinforcement learning approaches in autonomous driving. For instance, Toromanoff et al. (2020) presented a model-free RL strategy leveraging implicit affordances for urban driving [45]. Wang et al. (2021) introduced the InterFuser, a sensor-fusion transformer-based architecture enhancing the safety and interpretability of RL agents [46]. Additionally, Chen et al. (2021) developed cooperative multi-vehicle RL agents for complex urban scenarios in CARLA, highlighting significant improvements in collaborative navigation [47]. More recent CARLA-based multi-agent work reinforces these findings, demonstrating cooperative behaviors via MARL [48].

Prominent examples that embody these categories include Q-learning, REINFORCE, Actor–Critic algorithms, Proximal Policy Optimization (PPO), Deep Q-Networks (DQNs) [49,50] and Model Predictive Control (MPC) [51,52].

Table 1 summarizes the key characteristics of some of the most popular reinforcement learning methods, classifying them based on their use of models, policy/value orientation, and policy type.

Table 1. Comparison of widely used reinforcement learning methods.

Method	Use Model	Based on	On-Policy/Off-Policy
Q-learning	Model free	Values	Off-policy
REINFORCE	Model free	Policy	On-policy
Actor–Critic	Model free	Hybrid (Values + Policy)	On-policy
Proximal Policy Optimization (PPO)	Model free	Policy	On-policy
Deep Q-Network	Model free	Values (approx. using neural network)	Off-policy
Model Predictive Control (MPC)	Model based	Dynamics model + Policy Optimization	Typically off-policy

2.2. Related Works About Autonomous Driving

Several contemporary investigations have harnessed the CARLA simulator to push the boundaries of reinforcement-learning-based autonomous driving. Alongside these, early model-based RL efforts pursued realistic end-to-end driving pipelines [53], setting the stage for later latent-world-model approaches. For instance, Raw2Drive: Reinforcement Learning with Aligned World Models for End-to-End Autonomous Driving (implemented in CARLA v2) [54] introduces a cohesive agent that fuses learned world models with policy optimization to achieve robust, end-to-end vehicle control. In CuRLA: Curriculum Learning Based Deep Reinforcement Learning for Autonomous Driving [55], the authors confront the challenge of sluggish policy convergence in standard DRL agents by embedding a structured curriculum, thereby accelerating skill acquisition across progressively complex driving tasks. At the task level, RL has also been used for motion planning in negotiation-heavy settings; for example, unsignalized intersections are addressed with an RL-based planner in [56]. Finally, Multi-Objective Reinforcement Learning for Adaptive Personalized Autonomous Driving [57] presents a novel framework in which a multi-objective RL agent not only navigates autonomously but dynamically modulates its driving style according to user-defined preferences, balancing safety, efficiency, and comfort.

3. Review of Reinforcement Learning Approaches

Autonomous driving has emerged as a focal area of scientific inquiry, with reinforcement learning methods occupying a central role due to their inherent suitability for sequential decision-making problems. In this thesis, we undertake a comprehensive review of the literature on artificial intelligence techniques applied to autonomous driving, placing particular emphasis on reinforcement learning approaches.

Table 2 provides a concise but comprehensive overview of the state and action spaces used in the studies surveyed. Table 3 then details, in the corresponding sequence, the reward functions formulated and utilized within those same works, thus facilitating direct comparison and analysis.

Table 2. Summary of reviewed reinforcement learning papers' state spaces and action spaces.

Paper [Agent]	State Space	Action Space
Implementing a deep reinforcement learning model for autonomous driving [6] [PPO agent]	throttle [0, 1], steer [−1, 1], brake [0, 1], distance from waypoint, waypoint orientation [degrees]	throttle [0, 1], steer [−1, 1], brake [0, 1]
Deep reinforcement learning based control for Autonomous Vehicles in CARLA [7] [DRL-flatten-image agent]	121 dimensional vector (processed RGB image), distance from lane, ϕ [degrees]	steer [−1, 1], throttle [0, 1]
Deep reinforcement learning based control for Autonomous Vehicles in CARLA [7] [DRL-Carla-Waypoints agent]	vector of waypoints [w0...14], distance from lane, ϕ [degrees]	steer [−1, 1], throttle [0, 1]
Deep reinforcement learning based control for Autonomous Vehicles in CARLA [7] [DRL-CNN agent]	$I_{RGB[640 \times 480]}$, distance from lane, ϕ [degrees]	steer [−1, 1], throttle [0, 1]
Deep reinforcement learning based control for Autonomous Vehicles in CARLA [7] [DRL-Pre-CNN agent]	vector of waypoints [w0...14], distance from lane, ϕ [degrees]	steer [−1, 1], throttle [0, 1]
Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator [8] [RL agent]	distance from intersection, speed	stop [speed = 0 m/s], drive [speed = 5 m/s]
CARLA: An Open Urban Driving Simulator [9] [A3C agent]	$I_{RGB[84 \times 84]}$ speed, distance to goal, damage from collisions	steer [−1, 1], throttle [0, 1], brake [0, 1]
Think2Drive: Efficient Reinforcement Learning by Thinking with Latent World Model for Autonomous Driving [10] [Think2Drive agent]	$I_{BEV[128 \times 128 \times 34 \times 4]}$, speed	throttle [0, 1], brake [0, 1], steer [−1, 1]
Safe Navigation: Training Autonomous Vehicles using Deep Reinforcement Learning in CARLA [11] [Combined DQN agent]	distance, distance from obstacle, ϕ [degrees], speed, traffic light	brake [throttle = 0.0, brake = 1.0, steer = 0.0], drive straight [throttle = 0.3, brake = 0.0, steer = 0.0], turn left [throttle = 0.1, brake = 0.0, steer = −0.6], turn right [throttle = 0.1, brake = 0.0, steer = 0.6], turn slightly left [throttle = 0.4, brake = 0.0, steer = −0.1], turn slightly right [throttle = 0.4, brake = 0.0, steer = 0.1]

Table 2 systematically contrasts the state and action space designs across key reinforcement learning studies in CARLA. Early models such as the A3C agent from Dosovitskiy et al. [9] employed simple RGB images, vehicle speed, and distance to target—representing the minimal viable input for policy learning. More recent works, such as Think2Drive [10], introduce significantly higher-dimensional and semantically enriched state representations, including BEV tensors with 34 semantic channels across four temporal frames, enabling more abstract and generalizable world models.

Agents like those proposed by Pérez-Gil et al. [7] demonstrate a controlled comparison of input modalities—including vectorized images, CNN-based encodings, and waypoint-driven features—highlighting the performance trade-offs between raw sensor data and handcrafted spatial abstractions. In contrast, Razak et al. [6] leverage a relatively lean state set—focused on control-relevant geometry like distance and orientation to the nearest waypoint—favoring fast training and low system complexity. Nehme et al. [11] adopt a modular input structure that combines metric state values (e.g., distance, ϕ , traffic light) with a discrete action space reflecting semantically meaningful maneuvers, such as “turn left” or “drive straight.”

These distinctions emphasize not only the evolution of perceptual richness but also how input design constrains the expressiveness and structure of the agent’s output space. Overall, the diversity in state–action formulations across the reviewed works underscores the lack of standardization in evaluation and highlights the need for benchmarking that isolates the effects of input representation on policy learning and generalization.

Table 3 highlights a diverse range of reward shaping strategies employed in reinforcement learning for autonomous driving. Razak et al. [6] use a piecewise, velocity- and orientation-based reward function that emphasizes smooth driving behavior and lane adherence, but applies only a moderate penalty for infractions, which might limit its effectiveness in enforcing strict safety constraints. In contrast, Pérez-Gil et al. [7] combine a large penalty for collisions or lane departures with continuous terms rewarding forward motion and low lateral deviation, and add a substantial goal-reaching bonus. Their structure strikes a balance between task completion and behavioral stability. Gutierrez-Moreno et al. [8] design a sparse, event-triggered reward targeting intersection navigation: speed is mildly encouraged, success is rewarded, and both collisions and prolonged decision making are penalized. This simple scheme works well in isolated tasks but may be too coarse for general driving.

Dosovitskiy et al. [9] introduce one of the earliest dense formulations, integrating driving progress, speed changes, and penalties for damage, off-road driving, and collisions. Although comprehensive, it is sensitive to reward scaling and thus less portable across scenarios. Think2Drive [10] adopts a modular reward combining speed, distance traveled, lateral deviation, and steering smoothness, with tunable weights that allow fine-grained behavioral calibration. Finally, Nehme et al. [11] separate reward functions for driving and braking components, tailoring each with domain-specific logic. Their approach supports safe, interpretable modular control but introduces added engineering overhead. Overall, the table reveals trade-offs between simplicity and generalization, with denser, composite rewards yielding richer behavior at the cost of increased design complexity. Let me know if you would like a third paragraph for future work or benchmarking.

For consistency and interpretability, other parameters in each reward equation were fixed at constant values to avoid scaling extremes. Additionally, flat reward terms—those not dependent on the parameter being evaluated—are included as horizontal lines to indicate their constant contribution.

Table 3. Summary of reward functions from reviewed papers.

Paper [Agent]	Reward
Implementing a deep reinforcement learning model for autonomous driving [6] [There was only one agent presented]	$R = \begin{cases} -10, & \text{on infraction} \\ \frac{v}{v_{\min}} (1 - d_{\text{norm}}) a_{\text{rew}}, & v < v_{\min} \\ (1 - d_{\text{norm}}) a_{\text{rew}}, & v_{\min} \leq v < v_{\text{target}} \\ \left(1 - \frac{v - v_{\text{target}}}{v_{\max} - v_{\text{target}}}\right) (1 - d_{\text{norm}}) a_{\text{rew}}, & v \geq v_{\text{target}} \end{cases}$
Deep reinforcement learning based control for Autonomous Vehicles in CARLA [7] [DRL-flatten-image agent, DRL-Carla-Waypoints agent, DRL-CNN agent, DRL-Pre-CNN agent]	$R_t = \begin{cases} -200, & \text{if collision or lane change} \\ \sum_t (v_t \cos \phi_t - v_t \sin \phi_t - d_t), & \text{if car in lane} \\ 100, & \text{if goal reached} \end{cases}$
Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator [8] [There was only one agent presented]	$R = \begin{cases} k_v v_{\text{ego}}, & \text{standard driving,} \\ 1, & \text{crossing intersection,} \\ -2, & \text{collision,} \\ -\frac{0.2}{t_{\text{out}}}, & \text{episode-duration penalty} \end{cases}$
CARLA: An Open Urban Driving Simulator [9] [There was only one agent presented]	$R_t = 1000 (d_{t-1} - d_t) + 0.05 (v_t - v_{t-1}) - 0.00002 (c_t - c_{t-1}) - 2 (s_t - s_{t-1}) - 2 (o_t - o_{t-1})$
Think2Drive: Efficient RL by Thinking with Latent World Model [10] [There was only one agent presented]	$R = r_{\text{speed}} + \alpha_{\text{tr}} r_{\text{travel}} + \alpha_{\text{de}} p_{\text{deviation}} + \alpha_{\text{st}} c_{\text{steer}}$
Safe Navigation: Training Autonomous Vehicles using DRL in CARLA [11] [One agent combined two models]	$R_{\text{drive}} = \begin{cases} -200, & \text{if collisions} > 0 \vee \phi > 100 \vee d > 3, \\ R(s, \pi_d(s)) - 10, & \text{if collisions} \leq 0 \wedge \phi \leq 100 \wedge 2 < d \leq 3, \\ R(s, \pi_d(s)), & \text{if collisions} \leq 0 \wedge \phi \leq 100 \wedge d \leq 2, \end{cases}$ $R_{\text{brake}}(\text{collisions}, v, d, a) = \begin{cases} 3, & \text{if } v < 10d + 10 \wedge a = 0, \\ -1, & \text{if } v < 10d + 10 \wedge a = 1, \\ 4, & \text{if } v > 10d + 10 \wedge a = 1, \\ -2, & \text{if } v > 10d + 10 \wedge a = 0, \\ -10, & \text{if } v < 1 \wedge d > 100, \\ 200, & \text{if } v = 0 \wedge d < 150, \\ -200, & \text{if collisions} > 0, \\ 0, & \text{otherwise.} \end{cases}$

Figures 3–5 below illustrate how the reward function values reported in the reviewed articles vary in response to changes in a single parameter. The reward functions’ variable values have been evaluated by systematically varying three principal parameters within each function. When adjusting one parameter, the remaining parameters were held constant and selected to minimize their influence on the reward outcome and to prevent the emergence of extreme reward values.

The values in the reward graphs represent reward functions with varying values of one variable, while the rest of the variables remain constant. The reward functions R1, R2, R3, R4, R5, R6_drive, and R6_break correspond to the checked articles in the following order: [6–11]. The y-axis shows the reward values returned by the reward functions in the following articles. On the x-axis, the graphs have different values depending on the graph; they are speed values, collision flag, or distance from the center of the lane. Each of the three charts is shown in normalized and non-normalized versions.

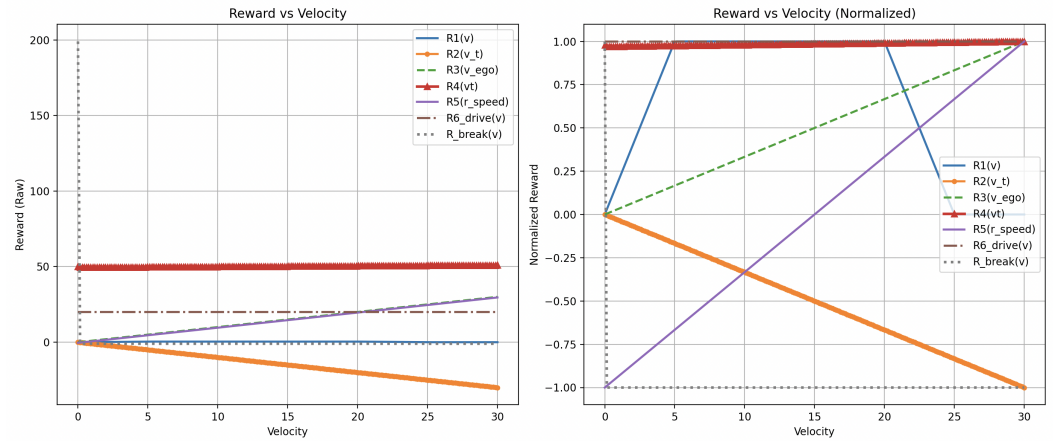


Figure 3. The graph shows the reward values as a function of changing vehicle speed values for the reward function from each article [6–11]. Other variables from these functions had constant values to illustrate the dependence of each agent’s reward on speed. The graphs show the unnormalized values on the left and the normalized values on the right.

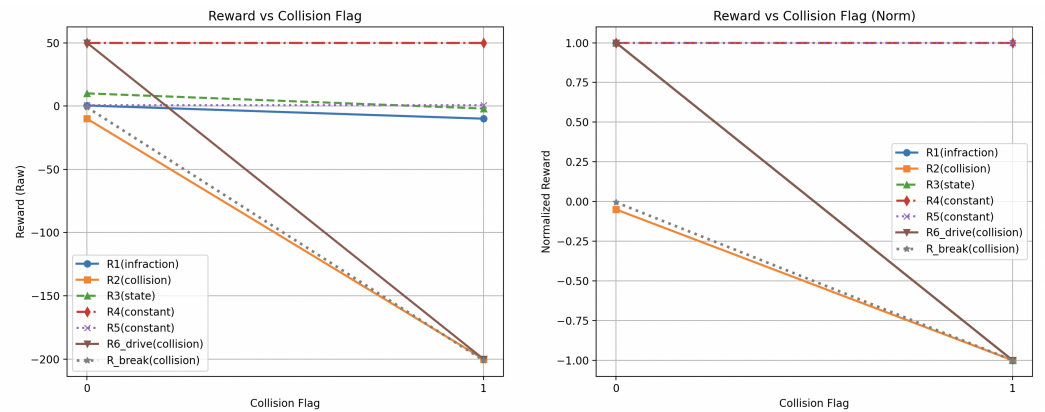


Figure 4. The graph shows the reward values, depending on whether a collision occurred while driving, for the reward functions from each of the [6–11] articles. Other variables from these functions had constant values to illustrate the dependence of each agent’s reward on collisions. The graphs show the unnormalized values on the left and the normalized values on the right.

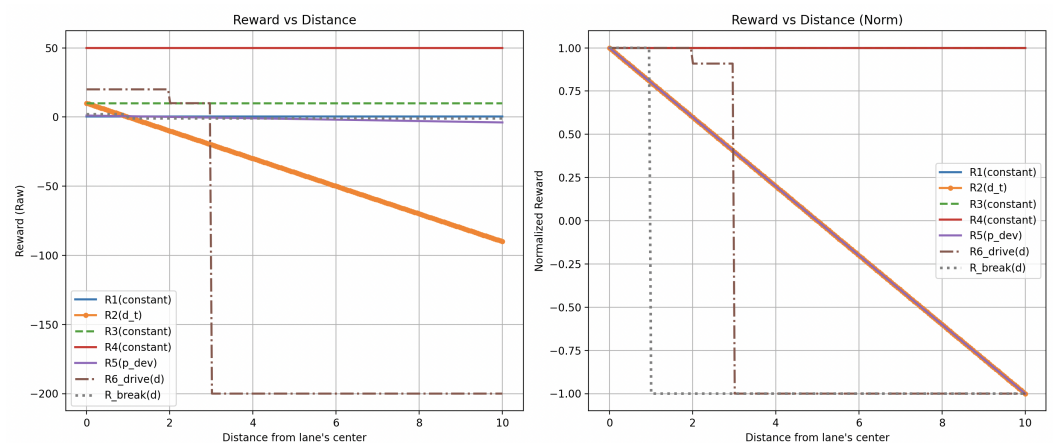


Figure 5. The graph shows the reward values as a function of varying vehicle distance from the center of the lane for the reward functions from each of the articles [6–11]. Other variables from these functions had constant values to illustrate the dependence of each agent’s reward on speed. The graphs show the unnormalized values on the left and the normalized values on the right.

The comparative analysis in Table 4 delivers a rigorous technical synthesis rather than a sequential listing. It clearly maps the evolution of input representations—from minimalist 2D vectors (e.g., speed and distance in Gutierrez-Moreno et al. [8]) to rich latent BEV maps with semantic and temporal history (Think2Drive [10]). While light inputs enable fast convergence in focused tasks, only richer perception models achieve the broad robustness necessary for real-world corner-case handling. For instance, Think2Drive processes high-dimensional BEV maps using a Dreamer-V3 latent world model, achieving exceptional route coverage via discrete maneuver planning. In contrast, earlier agents like Razak et al. [6] and Pérez-Gil et al. [7] rely on continuous control directly from raw or flattened image inputs, which allows for smoother trajectories but hinders sample efficiency and generalizability.

Table 4. Comparative analysis of reinforcement learning agents in autonomous driving literature.

Agent (Study)	Input Representation	Control Output	Key Performance Metrics	Architectural Highlights	Strengths and Limitations
Think2Drive [10]	BEV latent maps with semantic and temporal masks	Discrete set of 30 maneuvers	100% route completion on CARLA v2 within 3 days on A6000 GPU	Latent world model + Dreamer-V3 architecture	High generalization; requires privileged inputs and tuned latent space
Gutierrez-Moreno [8]	Speed + distance to intersection	Discrete stop/go	78–98% success in intersection scenarios; 0.04 collisions per episode	Curriculum learning; actor-critic; 2D input space	Fast convergence; limited for urban complexity
Dosovitskiy [9]	Dual RGB frames + auxiliary features	Continuous control signals	74% route success (baseline CARLA); low scalability	A3C baseline with dense reward	Generalizable benchmark; outdated by newer models
Pérez-Gil [7]	4 architectures: RGB → vector, waypoints, raw CNN, preprocessed CNN	Continuous steering/throttle	78% success; 18% collisions across setups	Compares flattened vs CNN vs waypoint inputs	Explores control-input trade-offs; lacks domain transfer tests
Razak [6]	Throttle, brake, steering, distance + orientation to waypoint	Continuous control	83% success; 12% collision (simulation report)	Simple handcrafted reward with angular shaping	Minimal setup; lacks multi-scenario adaptation
Nehme [11]	Depth + traffic light + deviation metrics	Modular discrete driving and braking decisions	86% success; 7% collision rate (simulation study)	Separate DQN for brake + maneuver control	Interpretability; modular; complex integration

Empirical outcomes further highlight emerging trends: Think2Drive is the first to report 100% route completion on CARLA v2 within three days of training on a single A6000 GPU [10]—a landmark in model-based RL efficacy. Model-free agents such as those by Dosovitskiy et al. [9] (A3C) and Razak et al. [6] (PPO) show route success between 74 and 83%, reflecting limited scalability. The modular design of Nehme et al. [11] introduces interpretability and enhanced safety through discrete braking and maneuvering policies but increases architectural complexity. This synthesis identifies critical design trade-offs—input complexity, action discretization, model-based vs model-free learning, and modularity—

while also exposing research gaps, such as the need for broader domain generalization, standardized benchmarks, and comparisons with camera-only RL agents such as those explored in emerging frameworks like CarLLaVA [58].

4. Gaps and Limitations in RL-Based Methods

While the previously reviewed studies present a diverse range of methodologies for autonomous driving within the CARLA simulation environment also as detailed in Appendix A, several persistent challenges and open research questions remain. These gaps reflect both the limitations inherent in the design of current reinforcement learning agents and the complexities of real-world deployment scenarios. Identifying and understanding these issues is crucial for improving the scalability, safety, and applicability of RL-based systems in autonomous driving.

The primary limitations observed across the surveyed literature are as follows:

- **Limited Generalization to Real-World Scenarios**—Some RL-based models exhibit strong performance within narrowly defined simulation environments but fail to generalize to the complexities of real-world driving. For instance, the agent described in [8] was trained under constrained conditions using a binary action set (stop or proceed). Although suitable for intersection navigation in simulation, this minimalist approach lacks the flexibility and robustness required for dynamic urban environments characterized by unpredictable agent interactions, diverse traffic rules, and complex road layouts.
- **Pipeline Complexity and Practical Deployment Issues**—Advanced RL systems often involve highly layered and interdependent components, which, while enhancing learning performance in simulation, may hinder real-time applicability. The architecture proposed in [11] integrates multiple sub-models for driving and braking decisions. Despite its efficacy within the training context, the increased architectural complexity poses challenges in deployment scenarios, such as increased inference latency, reduced system interpretability, and difficulties in modular updates or extensions.
- **Reward Function Design and Safety Trade-offs**—Crafting a reward function that effectively balances task completion with safe, rule-compliant behavior remains an ongoing challenge. Poorly calibrated rewards may inadvertently incentivize agents to exploit loopholes or adopt high-risk behaviors that maximize rewards at the cost of safety. The reward formulation presented in [7] mitigates this risk by incorporating lane deviation, velocity alignment, and collision penalties. Nevertheless, even well-intentioned designs can result in unintended behaviors if the agent learns to over-prioritize specific features, highlighting the need for reward tuning and safety regularization.

Several works have specifically tackled these limitations by employing advanced RL strategies. Yang et al. (2021) developed uncertainty-aware collision avoidance techniques to enhance safety in autonomous vehicles [59]. Fang et al. (2022) proposed hierarchical reinforcement learning frameworks addressing the complexity of urban driving environments [60]. Cui et al. (2023) further advanced curriculum reinforcement learning methods to tackle complex and dynamic driving scenarios effectively [61]. Feng et al. (2025) utilized domain randomization strategies to enhance the generalization of RL policies, significantly improving performance across varying simulated environments [62].

5. Imitation Learning (IL) for Autonomous Driving

Imitation learning (IL) is a machine learning paradigm in which an agent learns to perform tasks by mimicking expert behavior, typically that of a human. Unlike reinforcement

learning, where the agent discovers optimal actions through trial and error and reward feedback, IL infers a policy directly from expert demonstrations by mapping perceptual inputs to control outputs.

During training, the agent observes example trajectories and attempts to replicate the expert's decisions in the same contexts. The aim is for the model to internalize the expert's policy, enabling generalization to similar, unseen scenarios.

The overall process of IL is illustrated in Figure 6.

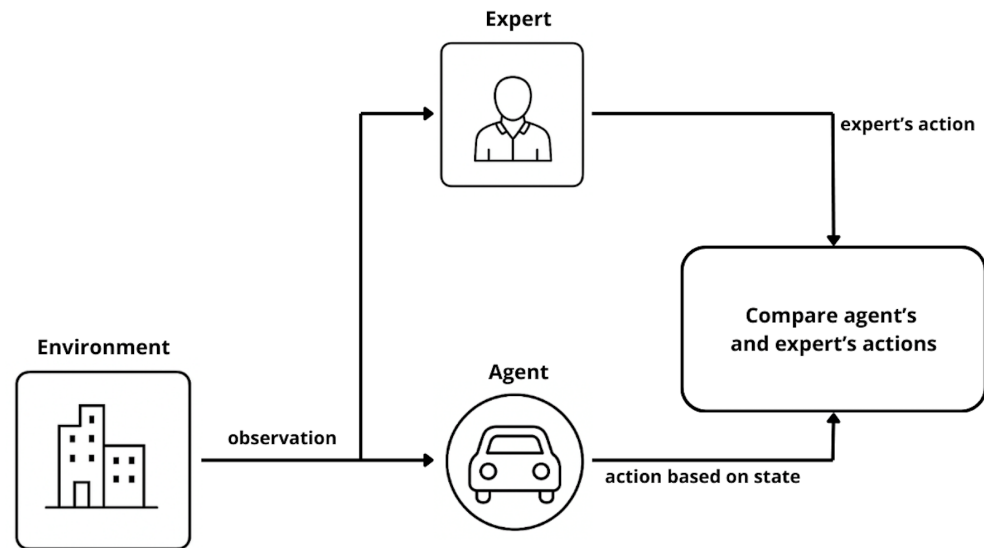


Figure 6. Diagram presenting imitation learning (IL) process.

As shown above, IL shares many components with reinforcement learning, though the learning mechanism differs. The key actors in the IL framework are:

- **Agent**—The learner that interacts with the environment. Unlike in reinforcement learning, the agent does not rely on scalar rewards but instead learns to imitate the expert's demonstrated behavior.
- **Expert**—Typically, a human operator or a pretrained model that provides high-quality demonstration trajectories. These trajectories serve as the ground truth for training.

All remaining terminology, such as state, action, and environment, corresponds to the definitions previously introduced in Section 2.1 on reinforcement learning.

IL algorithms are typically categorized along two orthogonal dimensions:

- **Model-based vs. Model-free**—Model-based approaches attempt to build a transition model of the environment and use it for planning. Model-free techniques rely solely on observed expert trajectories without reconstructing the environment's internal dynamics.
- **Policy-based vs. Reward-based**—Policy-based methods directly learn a mapping from states to actions, while reward-based methods (such as inverse reinforcement learning) infer the expert's reward function before deriving a policy.

A selection of widely used IL techniques are summarized in Table 5.

Recent advancements in imitation learning also demonstrate promising outcomes. Luo et al. (2023) improved behavioral cloning techniques by introducing adaptive data augmentation methods, significantly enhancing IL agents' performance in novel driving scenarios [63]. Moreover, Mohanty et al. (2024) utilized inverse reinforcement learning to achieve human-like autonomous driving behaviors in CARLA, emphasizing the effectiveness of learning from expert demonstrations [64].

Table 5. Summary of popular IL methods.

Method	Use Model	Based on
Behavioral Cloning (BC)	Model free	Values
DAGger (Dataset Aggregation)	Model free	Policy
Inverse Reinforcement Learning (IRL)	Indirect (Learns Reward)	Reward
Generative Adversarial IL (GAIL)	Model free	Policy

5.1. IL-Based Architectures

IL has gained considerable traction in autonomous driving research due to its ability to leverage expert demonstrations and simplify policy learning. This section reviews key publications that implement IL-based approaches for autonomous vehicle control.

The reviewed works differ in their perceptual inputs, control representations, and agent architectures. Table 6 summarizes the state and action spaces for each model discussed.

Table 6. Summary of reviewed IL papers, their state spaces, and action spaces.

Paper [Agent]	State Space	Action Space
End-to-end Driving via Conditional IL [12] [Simulation and physical agent; identical state/action spaces]	$I_{RGB[800 \times 600]}$	steer $[-1, 1]$, acceleration $[-1, 1]$
Learning by Cheating [13] [Privileged agent]	$M_{BEV[320 \times 320 \times 7]}$, speed, command	steer, brake, throttle
Learning by Cheating [13] [Sensorimotor agent]	$I_{RGB[384 \times 160]}$, speed, command	steer, brake, throttle
Dynamic Conditional IL for Autonomous Driving [14] [Single-agent architecture]	I_{RGB} , command, LiDAR, speed	steer $[-1, 1]$, throttle $[0, 1]$, brake $[0, 1]$

Table 7 offers a comparative overview of several IL approaches in autonomous driving, focusing on the modalities used for perception and control. Each agent architecture incorporates different combinations of visual, sensor, and navigation inputs to compute real-time driving decisions.

Table 7 offers a structured comparative analysis of imitation learning (IL) methods for autonomous driving, going beyond paper-by-paper description to extract underlying architectural and perceptual trends. Across all reviewed works, we observe a key technical divergence: some agents rely on end-to-end mappings from raw sensor data (e.g., monocular RGB in [12,13]), while others leverage structured intermediate representations such as BEV semantic maps [13] or fused LiDAR polar grids [14]. This split has major implications for model interpretability, transferability, and dependence on privileged information. Architectures like the “privileged agent” in [13] explicitly encode scene semantics, enabling spatially grounded planning via heatmap prediction, but require access to annotated maps. In contrast, purely sensorimotor approaches trade interpretability for operational independence, relying on network capacity to infer spatial features implicitly.

Table 7. Comparative analysis of imitation learning agents in autonomous driving literature.

Agent (Study)	Input Representation	Control Output	Key Performance Metrics	Architectural Highlights	Strengths and Limitations
Codevilla et al. [12]	RGB image (800 × 600), directional command	Continuous steering, acceleration	96% success rate in simulation; real-world feasibility via scaled platform	Conditional branching policy; centralized end-to-end network	Simple design; low robustness to distributional shift; single camera limitation
Chen et al. [Privileged] [13]	Semantic BEV map (320 × 320 × 7), speed, command	Steering, throttle, brake (via controller on predicted waypoints)	Achieves near-expert imitation with minimal variance	Two-stage pipeline with expert heatmap output and PID controller	High accuracy, but requires privileged input unavailable at test time
Chen et al. [Sensorimotor] [13]	RGB image (384 × 160), speed, command	Steering, throttle, brake	Generalizes well when trained via privileged supervision	Learns to imitate expert via internal heatmap prediction	End-to-end trainable; suboptimal if privileged guidance is inaccurate
Eraqi et al. [14]	RGB image, LiDAR (polar grid), speed, directional command	Steering, throttle, brake	92% success rate in navigation scenarios; improved safety margins	Multi-modal encoder; dynamic conditional policy branches	Robust multi-sensor fusion; higher computational complexity

Another dimension of comparison lies in the control abstraction. While all agents ultimately produce low-level continuous commands (steering, throttle, brake), they differ in how commands are conditioned. Codevilla et al. [12] and Eraqi et al. [14] both adopt branched architectures, where distinct sub-networks handle different high-level directional commands. This conditional control strategy improves behavioral specialization and aligns with hierarchical task decomposition, but may introduce fragility if command prediction is noisy. Additionally, multi-modal architectures like in [14] show clear technical advantages in handling diverse scenarios by integrating visual, depth, and navigation cues. However, such models increase architectural complexity and raise questions about data alignment and modality-specific failures. In sum, the table synthesizes these trends to expose the underlying design principles, trade-offs, and open challenges in IL for autonomous driving—thus directly responding to the reviewer’s request for more comparative and technically grounded analysis.

5.2. Hybrid Solutions

IL, while effective for quickly learning behaviors from expert demonstrations, struggles with limited generalization—agents often fail when encountering situations not covered in the training data. On the other hand, reinforcement learning excels at learning through exploration and can achieve high performance, but it is computationally expensive and prone to instability. Combining these two approaches helps to mitigate their weaknesses: expert demonstrations can guide and accelerate learning, reducing unsafe or inefficient exploration. At the same time, the reward-driven nature of RL enables agents to refine their behavior and adapt to novel scenarios. Within the scope of this review, it is important to highlight the increasing synergy between IL and reinforcement learning in autonomous driving systems.

Table 8 provides a comparative analysis of hybrid architectures that integrate imitation learning (IL) and reinforcement learning (RL) in the context of autonomous driving. This table highlights key technical differentiators across models—such as input modalities, control schemes, performance outcomes, and architectural decisions—thus enabling a deeper comparative discussion of their contributions. For instance, several approaches, including CIRL [65] and Phan et al. [66], adopt a two-stage learning process where IL serves as a pretraining phase followed by RL-based policy refinement. These approaches are particularly effective in sparse-reward environments, allowing agents to learn safer and more optimal behaviors with fewer exploration steps.

Table 8. Comparative analysis of hybrid IL+RL agents in autonomous driving literature.

Agent (Study)	Input Representation	Control Output	Key Performance Metrics	Architectural Highlights	Strengths and Limitations
Phan et al. [66]	RGB, speed, high-level commands	Steering, throttle, brake	Effective policy transfer to real-world intersections	Inverse RL followed by RL fine-tuning	Combines realism and safety; complex pretraining required
Ho and Ermon [67]	Trajectory sequences (expert vs. agent)	Action likelihood over states	Performs well in MuJoCo; widely adopted in AV variants	GAIL adversarial setup: discriminator + generator agent	No reward engineering; sensitive to discriminator instability
Liang et al. [CIRL] [65]	Camera image, vehicle speed, navigation command	Steering, throttle, brake	High success rate in intersection and obstacle-avoidance tasks	Behavior cloning + DDPG in staged learning	Improves generalization; requires balanced training phase timing
Chekroun et al. [GRI] [68]	Sensor data, expert trajectory actions	Steering, throttle, brake	Tested in CARLA and MuJoCo; fast convergence	Unified replay buffer; expert data has constant high reward	No IL pretraining needed; success depends on reward scaling
Han et al. [69]	Sensor data + sparse expert cues	Blended: agent and expert actions (probabilistic)	Rapid learning with low collision rates	Exploration guided by sparse expert rules	Safe learning; depends on sparse rule coverage
Reddy et al. [SQIL] [70]	Standard RL states + expert trajectories	RL actions from standard policy	Improves over baseline Soft Q-Learning in MuJoCo	Simple buffer reward logic (1 for expert, 0 for agent)	Enhanced safety and generalization in CARLA scenarios
Huang et al. [71]	RGB image + LiDAR + GPS	Steering, throttle, brake	Improved driving scores in complex scenarios	Hierarchical fusion of IL + PPO RL agent	Very easy to integrate; lacks fine control on imitation priority
Kim and Cho [72]	Camera + sensor state features	Steering, throttle, brake	Improved driving scores in complex scenarios	Combines attention-based IL with actor–critic RL	Flexible framework; limited outside simulation
					Effective in dense environments; lacks interpretability

Moreover, models such as GAIL [67] and SQIL [70] depart from handcrafted reward engineering by leveraging either adversarial training or reward shaping based on expert trajectories. GRI [68] and the sparse imitation framework from Han et al. [69] further exemplify how continuous or conditional use of expert data can stabilize learning and improve generalization. While these hybrid strategies differ in implementation complexity and assumptions (e.g., availability of full expert data vs. sparse hints), they all converge toward a common goal: leveraging demonstration data not only as initial guidance but as a persistent signal for optimization throughout training. These insights underscore the importance of hybrid frameworks in bridging the sample efficiency of IL with the adaptiveness and robustness of RL, thus offering practical, scalable solutions for real-world autonomous driving.

6. Comparative Analysis and Discussion

The comparative review of reinforcement learning and IL approaches for autonomous driving in the CARLA simulator reveals distinct strengths and limitations for each paradigm. These trade-offs echo findings from a comparative study of behavioral cloning and reinforcement learning for autonomous driving [73].

Table 9 presents a consolidated overview of all the studies reviewed, detailing the main objective of each work, the complexity of its proposed architecture, the complexity of its testing environment, and the total number of driving and route scenarios evaluated. To facilitate a standardized comparison, both the architectural and environmental complexity is quantified on a three-point scale (1–3), with “3” denoting the highest level of complexity. For example, an architecture that integrates multiple distinct models (e.g., [11]) is assigned a complexity score of 2, and similarly, an environment encompassing several urban centers or diverse traffic elements receives an environmental complexity rating of 2.

This comparative structuring reveals several insights:

- **Baseline approaches tend to be simple and narrowly scoped.** Papers comparing DQN and DDPG in Town 1 or intersections under limited conditions fall here. They offer important groundwork but do little to stress-test agents’ generalization across environments.
- **Mid-complexity architectures (PPO + VAE, braking/driving modules) improve robustness.** Curriculum learning (CuRLA [55]; see also [74]) and modular braking–driving workflows demonstrate that strategic layering of subsystems yields robustness to changes in traffic or new tasks.
- **High-complexity, high-diversity methods push boundaries.** State-of-the-art works like Think2Drive and Raw2Drive introduce latent world models or dual sensor-stream inputs, validated across dozens to hundreds of scenarios. These methods exemplify increased policy adaptability to rare and corner cases.
- **Hybrid and multi-objective agents indicate a shifting paradigm.** While still relatively underexplored, works that incorporate user preferences or multi-objective reward structures begin to bridge from pure RL toward human-centric driving goals.

Table 9. Summary of results from discussed articles.

Article	Goal	Architecture Complexity	Environment Complexity	Scenarios	Advantages
Implementing a Deep Reinforcement Learning Model for Autonomous Driving	End-to-end RL agent with Variational Autoencoder (VAE)	2	3	One scenario in three towns (1, 2 and 7)	An extensive and complex approach to the subject

Table 9. Cont.

Article	Goal	Architecture Complexity	Environment Complexity	Scenarios	Advantages
Deep Reinforcement Learning Based Control for Autonomous Vehicles in CARLA	Comparison of DQN and DDPG on few models	1	1	Two scenarios in town 1	Multiple and varied approaches to agent's architecture
Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator	Complete agent capable of driving through intersection with traffic	2	3	Three scenarios, one for each type of intersection (lights, stop signal, uncontrolled)	Complex approach to crossing intersection with traffic
CARLA: An Open Urban Driving Simulator	Comparison of performance of RL, IL and Modular Pipeline	1	3	Five scenarios in four possibilities (training conditions, new town, new weather, new weather and town)	A good comparison of RL with other machine learning methods with a multi-sensor agent
Think2Drive: Efficient Reinforcement Learning by Thinking with Latent World Model for Autonomous Driving (in CARLA-V2)	RL agent possible to drive in multiple corner cases	3	3	39 detailed scenarios	A highly complex approach to an agent capable of driving multiple corner case scenarios
Safe Navigation: Training Autonomous Vehicles using Deep Reinforcement Learning in CARLA	RL agent capable of driving to maintain speed and break when necessary to avoid collision	2	2	Four scenarios in town 2	A complex and robust approach to an agent composed of two models (breaking model and driving model)
Raw2Drive: Reinforcement Learning with Aligned World Models for End-to-End Autonomous Driving (in CARLA v2)	Model-based RL agent capable of learning effective driving from raw sensor data	3	3	220 routes from Batch2Drive benchmark	Recent highly complex and highly robust dual-stream Model Based RL approach with one stream of privileged sensor data

Table 9. Cont.

Article	Goal	Architecture Complexity	Environment Complexity	Scenarios	Advantages
CuRLA: Curriculum Learning Based Deep Reinforcement Learning For Autonomous Driving	PPO+VAE agent for driving in environment with increasing traffic	2	2	Few routes in town 7 with changing traffic	A modern approach combining Curriculum Learning with Deep Reinforcement Learning
Multi-Objective Reinforcement Learning for Adaptive Personalized Autonomous Driving	Multi-objective RL agent capable of driving according to preferences	2	1	One scenario	A modern approach to autonomous driving with a multi-objective (selected driving preferences) end-to-end agent

7. Conclusions

This review has analyzed and synthesized recent reinforcement learning and IL approaches for autonomous driving, with a focus on studies implemented within the CARLA simulator. The literature surveyed demonstrates that both paradigms offer unique advantages and face specific limitations in the context of simulated driving environments.

Reinforcement learning enables agents to learn flexible, task-optimized behaviors through exploration and interaction with their environment. Its strength lies in its adaptability and end-to-end training pipeline. However, RL often requires carefully engineered reward functions and extensive simulation time, and it struggles with generalization and sample inefficiency.

IL, on the contrary, allows agents to quickly acquire competent policies from expert demonstrations, avoiding the complexities of reward design. IL methods typically converge faster and demonstrate smoother behavior in constrained settings. However, their dependence on demonstration quality and vulnerability to distributional shifts can limit robustness in novel or unpredictable scenarios.

Several hybrid approaches have emerged to combine the strengths of both paradigms, including policy pretraining, adversarial learning, and inverse reinforcement learning. These strategies hold promise in overcoming individual limitations and advancing the state of the art in learning-based driving agents.

Looking ahead, the field is poised to benefit from continued integration between reinforcement learning and IL frameworks. Emerging trends point toward greater emphasis on safety, explainability, and real-world transfer, particularly as simulation-to-reality gaps remain a persistent obstacle for deploying learned agents beyond synthetic benchmarks.

CARLA has played a central role in standardizing experimental evaluation for autonomous driving, allowing reproducible comparisons between studies. However, expanding benchmarks to incorporate dynamic human interactions, multi-agent negotiation, and diverse sensory conditions will be essential for realistic policy development.

This review underscores the importance of modular, interpretable, and sample-efficient approaches that combine expert guidance with autonomous learning. As learning-based methods mature, the focus must shift toward robust, certifiable agents capable of operating in open-world settings.

Author Contributions: Conceptualization, P.C. and B.K.; methodology, P.C. and B.K.; validation, M.S. and M.W.; formal analysis, M.S. and M.W.; investigation, B.K. and M.S.; resources, P.C.; writing—original draft preparation, P.C. and B.K.; writing—review and editing, M.S. and M.W.; visualization, P.C., B.K. and M.S.; supervision, M.S. and M.W.; project administration, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: GenAI is used for language checks and corrections while ensuring the content is maintained and reviewed thoughtfully.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RL	Reinforcement Learning
IL	Imitation Learning
CARLA	Car Learning to Act (Autonomous Driving Simulator)
CNN	Convolutional Neural Network
BEV	Bird's Eye View
PGV	Polar Grid View
IRL	Inverse Reinforcement Learning
GAIL	Generative Adversarial IL
MDP	Markov Decision Process
PPO	Proximal Policy Optimization
DQN	Deep Q-Network
A3C	Asynchronous Advantage Actor–Critic
LiDAR	Light Detection and Ranging
RGB	Red–Green–Blue (Color Image Input)
BC	Behavioral Cloning
Dagger	Dataset Aggregation
PID	Proportional–Integral–Derivative (Controller)
PGM	Probabilistic Graphical Model

Appendix A. Supplementary Results

CARLA Simulator as an Evaluation Platform

CARLA (Car Learning to Act) [5] is an open-source, high-fidelity driving simulator developed to support research in autonomous vehicles and intelligent transportation systems. Its detailed urban environments, physics engine, and API flexibility make it a preferred tool for training and evaluating RL agents in realistic conditions [75].

Key features of CARLA include:

- Management and control of both vehicles and pedestrians;
- Environment customization (e.g., weather, lighting, time of day);
- Integration of various sensors such as LiDAR, radar, GPS, and RGB cameras;
- Python/C++ APIs for interaction and simulation control.

These capabilities support deterministic and repeatable scenarios, which are fundamental for reliable experimentation in RL-based driving. To bridge the sim-to-real gap posed by CARLA's synthetic imagery, recent work employs *domain randomization* techniques—such as varying lighting, textures, weather, and object appearance—to promote domain-invariant feature learning, enabling models trained purely on simulated data

to generalize to real domains [76]. In addition, image translation networks (e.g., CycleGAN or Pix2Pix variants) have been used to refine CARLA outputs into styles closely resembling real-world datasets. Sensor realism is further enhanced by modeling physics-based noise in LiDAR and camera outputs to mimic real hardware artifacts. CARLA's configurability is a critical advantage: it offers programmable weather and lighting, a rich multi-modal sensor suite (RGB, depth, semantic segmentation, LiDAR, radar, event camera), and fully scriptable scenarios via Python and OpenSCENARIO APIs. Platforms such as Prescan and CarSim are often valued for their high-fidelity vehicle dynamics and validated sensor physics, which are particularly useful in control design and ADAS validation. However, they tend to offer less flexibility in terms of visual realism and large-scale data generation for perception tasks. Meanwhile, simulators like AirSim and LGSVL provide diverse sensor capabilities and visually rich environments, although their extensibility and documentation support vary. According to a 2021 survey, CARLA and LGSVL represent the current state of the art among open-source simulators for end-to-end testing, offering a reasonable balance between environmental realism, multi-sensor support, and scriptable scenarios—making them commonly adopted in research [77]. For perception-based sim-to-real transfer, CARLA has been used to augment rare object detection scenarios: synthetic images generated in CARLA, including varied weather and object placement, contributed to improved detection performance in real-world low-shot conditions [78]. This suggests that, when used with appropriate augmentation, CARLA can be a useful component in perception pipelines, though it may not be optimal for high-fidelity dynamics testing.

Benchmarking the performance and robustness of RL algorithms has been significantly streamlined by CARLA. Zeng et al. (2024) provided a comprehensive benchmarking study, assessing autonomous driving systems in dynamic simulated traffic conditions [79]. Similarly, Jiang et al. (2025) analyzed various RL algorithms in CARLA, focusing on their stability, robustness, and performance consistency across multiple driving tasks [80].

Figure A1 illustrates sample outputs generated by the CARLA simulator. Parts of the figure on the left side show outputs from an RGB front camera and RGB camera behind the vehicle. The part of the figure on the right shows the point cloud output from the LiDAR sensor.

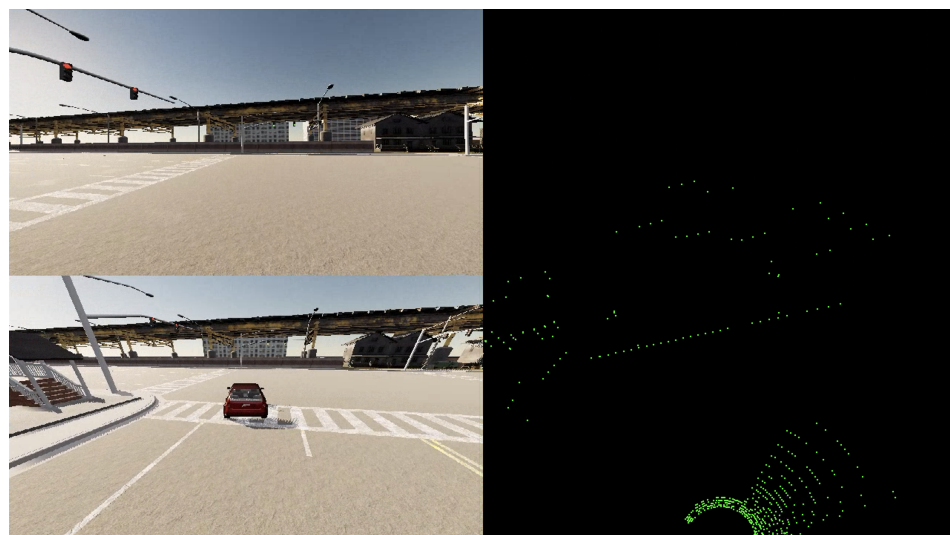


Figure A1. A typical urban environment rendered in the CARLA simulator, illustrating its high visual fidelity and sensor integration.

References

1. Dhinakaran, M.; Rajasekaran, R.T.; Balaji, V.; Aarthi, V.; Ambika, S. Advanced Deep Reinforcement Learning Strategies for Enhanced Autonomous Vehicle Navigation Systems. In Proceedings of the 2024 2nd International Conference on Computer, Communication and Control (IC4), Indore, India, 8–10 February 2024.
2. Govinda, S.; Brik, B.; Harous, S. A Survey on Deep Reinforcement Learning Applications in Autonomous Systems: Applications, Open Challenges, and Future Directions. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 11088–11113. [\[CrossRef\]](#)
3. Kong, Q.; Zhang, L.; Xu, X. Constrained Policy Optimization Algorithm for Autonomous Driving via Reinforcement Learning. In Proceedings of the 2021 6th International Conference on Image, Vision and Computing (ICIVC), Qingdao, China, 23–25 July 2021.
4. Kim, S.; Kim, G.; Kim, T.; Jeong, C.; Kang, C.M. Autonomous Vehicle Control Using CARLA Simulator, ROS, and EPS HILS. In Proceedings of the 2025 International Conference on Electronics, Information, and Communication (ICEIC), Osaka, Japan, 19–22 January 2025.
5. Malik, S.; Khan, M.A.; El-Sayed, H. CARLA: Car Learning to Act—An Inside Out. *Procedia Comput. Sci.* **2022**, *198*, 742–749. [\[CrossRef\]](#)
6. Razak, A.I. Implementing a Deep Reinforcement Learning Model for Autonomous Driving. 2022. Available online: <https://faculty.utrgv.edu/dongchul.kim/6379/t14.pdf> (accessed on 30 July 2025).
7. Pérez-Gil, Ó.; Barea, R.; López-Guillén, E.; Bergasa, L.M.; Gómez-Huélamo, C.; Gutiérrez, R.; Díaz-Díaz, A. Deep Reinforcement Learning Based Control for Autonomous Vehicles in CARLA. *Electronics* **2022**, *11*, 1035. [\[CrossRef\]](#)
8. Gutiérrez-Moreno, R.; Barea, R.; López-Guillén, E.; Araluce, J.; Bergasa, L.M. Reinforcement Learning-Based Autonomous Driving at Intersections in CARLA Simulator. *Sensors* **2022**, *22*, 8373. [\[CrossRef\]](#)
9. Dosovitskiy, A.; Ros, G.; Codevilla, F.; López, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. *arXiv* **2017**, arXiv:1711.03938. [\[CrossRef\]](#)
10. Li, Q.; Jia, X.; Wang, S.; Yan, J. Think2Drive: Efficient Reinforcement Learning by Thinking with Latent World Model for Autonomous Driving (in CARLA-V2). *IEEE Trans. Intell. Veh.* **2024**, early access. [\[CrossRef\]](#)
11. Nehme, G.; Deo, T.Y. Safe Navigation: Training Autonomous Vehicles using Deep Reinforcement Learning in CARLA. *Sensors* **2023**, *23*, 7611.
12. Codevilla, F.; Müller, M.; López, A.; Koltun, V.; Dosovitskiy, A. End-to-end Driving via Conditional IL. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2017.
13. Chen, D.; Zhou, B.; Koltun, V.; Krähenbühl, P. Learning by Cheating. *arXiv* **2019**, arXiv:1912.12294. [\[CrossRef\]](#)
14. Eraqi, H.M.; Moustafa, M.N.; Höner, J. Dynamic Conditional IL for Autonomous Driving. *arXiv* **2022**, arXiv:2211.11579.
15. Abdou, M.; Kamai, H.; El-Tantawy, S.; Abdelkhalek, A.; Adei, O.; Hamdy, K.; Abaas, M. End-to-End Deep Conditional IL for Autonomous Driving. In Proceedings of the 2019 31st International Conference on Microelectronics (ICM), Cairo, Egypt, 15–18 December 2019.
16. Li, Z. A Hierarchical Autonomous Driving Framework Combining Reinforcement Learning and IL. In Proceedings of the 2021 International Conference on Computer Engineering and Application (ICCEA), Kunming, China, 25–27 June 2021.
17. Sakhai, M.; Wielgosz, M. Towards End-to-End Escape in Urban Autonomous Driving Using Reinforcement Learning. In *Intelligent Systems and Applications. IntelliSys 2023*; Arai, K., Ed.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2024; Volume 823. [\[CrossRef\]](#)
18. Kołomański, M.; Sakhai, M.; Nowak, J.; Wielgosz, M. Towards End-to-End Chase in Urban Autonomous Driving Using Reinforcement Learning. In *Intelligent Systems and Applications. IntelliSys 2022*; Arai, K., Ed.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 544. [\[CrossRef\]](#)
19. Sakhai, M.; Mazurek, S.; Caputa, J.; Argasiński, J.K.; Wielgosz, M. Spiking Neural Networks for Real-Time Pedestrian Street-Crossing Detection Using Dynamic Vision Sensors in Simulated Adverse Weather Conditions. *Electronics* **2024**, *13*, 4280. [\[CrossRef\]](#)
20. Sakhai, M.; Sithu, K.; Soe Oke, M.K.; Wielgosz, M. Cyberattack Resilience of Autonomous Vehicle Sensor Systems: Evaluating RGB vs. Dynamic Vision Sensors in CARLA. *Appl. Sci.* **2025**, *15*, 7493. [\[CrossRef\]](#)
21. Huang, Y.; Xu, X.; Yan, Y.; Liu, Z. Transfer Reinforcement Learning for Autonomous Driving under Diverse Weather Conditions. *IEEE Trans. Intell. Veh.* **2022**, *7*, 593–603.
22. 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 Trans. Intell. Transp. Syst.* **2021**, *23*, 4909–4926. [\[CrossRef\]](#)
23. Zhu, Z.; Zhao, H. A survey of deep RL and IL for autonomous driving policy learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 14043–14065. [\[CrossRef\]](#)
24. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. A Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [\[CrossRef\]](#)
25. Shrestha, A.; Mahmood, A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* **2019**, *7*, 53040–53065. [\[CrossRef\]](#)
26. Elavarasan, D.; Vincent, P.M.D. Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications. *IEEE Access* **2020**, *8*, 86886–86901. [\[CrossRef\]](#)

27. Zhou, Z.; Chen, X.; Li, E.; Zeng, L.; Lue, K.; Zhang, J. Edge Intelligence: Paving the Last Mile of Artificial Intelligence with Edge Computing. *Proc. IEEE* **2019**, *107*, 1738–1762. [[CrossRef](#)]
28. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
29. Cui, J.; Liu, Y.; Arumugam, N. Multi-Agent Reinforcement Learning-Based Resource Allocation for UAV Networks. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 729–743. [[CrossRef](#)]
30. Shaukat, K.; Luo, S.; Varadharajan, V.; Hameed, I.; Xu, M. A Survey on Machine Learning Techniques for Cyber Security in the Last Decade. *IEEE Access* **2020**, *8*, 222310–222354. [[CrossRef](#)]
31. Ye, H.; Li, G.Y.; Juang, B.F. Deep Reinforcement Learning Based Resource Allocation for V2V Communications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3163–3173. [[CrossRef](#)]
32. Le, L.; Nguyen, T.N. DQRA: Deep Quantum Routing Agent for Entanglement Routing in Quantum Networks. *IEEE Trans. Quantum Eng.* **2022**, *3*, 4100212. [[CrossRef](#)]
33. Scholköpf, B.; Locatello, F.; Bauer, S.; Ke, N.R.; Kalchbrenner, N.; Goyal, A.; Bengio, Y. Toward Causal Representation Learning. *Proc. IEEE* **2021**, *109*, 612–634. [[CrossRef](#)]
34. Huang, C.; Zhang, H.; Wang, L.; Luo, X.; Song, Y. Mixed Deep Reinforcement Learning Considering Discrete-continuous Hybrid Action Space for Smart Home Energy Management. *J. Mod. Power Syst. Clean Energy* **2022**, *10*, 743–754. [[CrossRef](#)]
35. Sogabe, T.; Malla, D.B.; Takayama, S.; Shin, S.; Sakamoto, K.; Yamaguchi, K.; Singh, T.P.; Sogabe, M.; Hirata, T.; Okada, Y. Smart Grid Optimization by Deep Reinforcement Learning over Discrete and Continuous Action Space. In Proceedings of the 2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC), Waikoloa, HI, USA, 10–15 June 2018.
36. Guériau, M.; Cardozo, N.; Dusparic, I. Constructivist Approach to State Space Adaptation in Reinforcement Learning. In Proceedings of the 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Umea, Sweden, 16–20 June 2019.
37. Abdulazeez, D.H.; Askar, S.K. Offloading Mechanisms Based on Reinforcement Learning and Deep Learning Algorithms in the Fog Computing Environment. *IEEE Access* **2023**, *11*, 12555–12586. [[CrossRef](#)]
38. Lapan, M. *Głębokie Uczenie Przez Wzmocnianie. Praca z Chatbotami Oraz Robotyka, Optymalizacja Dyskretna i Automatyzacja Sieciowa w Praktyce*; Helion: Everett, WA, USA, 2022.
39. Mahadevkar, S.V.; Khemani, B.; Patil, S.; Kotecha, K.; Vora, D.R.; Abraham, A.; Gabralla, L.A. A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions. *IEEE Access* **2022**, *10*, 107293–107329. [[CrossRef](#)]
40. Shukla, I.; Dozier, H.R.; Henslee, A.C. A Study of Model Based and Model Free Offline Reinforcement Learning. In Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2022.
41. Hyang, Q. Model-Based or Model-Free, a Review of Approaches in Reinforcement Learning. In Proceedings of the 2020 International Conference on Computing and Data Science (CDS), Stanford, CA, USA, 1–2 August 2020.
42. Beyon, H. Advances in Value-based, Policy-based, and Deep Learning-based Reinforcement Learning. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*. [[CrossRef](#)]
43. Liu, M.; Wan, Y.; Lewis, F.L.; Lopez, V.G. Adaptive Optimal Control for Stochastic Multiplayer Differential Games Using On-Policy and Off-Policy Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 5522–5533. [[CrossRef](#)]
44. Banerjee, C.; Chen, Z.; Noman, N.; Lopez, V.G. Improved Soft Actor-Critic: Mixing Prioritized Off-Policy Samples With On-Policy Experiences. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 3121–3129. [[CrossRef](#)] [[PubMed](#)]
45. Toromanoff, M.; Wirbel, E.; Moutarde, F. End-to-end Model-free Reinforcement Learning for Urban Driving Using Implicit Affordances. *arXiv* **2020**, arXiv:2001.09445.
46. Wang, Y.; Chitta, K.; Liu, H.; Chernova, S.; Schmid, C. InterFuser: Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer. *arXiv* **2021**, arXiv:2109.05499.
47. Chen, Y.; Li, H.; Wang, Y.; Tomizuka, M. Learning Safe Multi-Vehicle Cooperation with Policy Optimization in CARLA. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3568–3575.
48. Liu, Y.; Zhang, Q.; Zhao, L. Multi-Agent Reinforcement Learning for Cooperative Autonomous Vehicles in CARLA. *J. Intell. Transp. Syst.* **2025**, *29*, 198–212.
49. Nikpour, B.; Sinodinos, D.; Armanfard, N. Deep Reinforcement Learning in Human Activity Recognition: A Survey. *TechRxiv* **2022**. [[CrossRef](#)]
50. Kim, J.; Kim, G.; Hong, S.; Cho, S. Advancing Multi-Agent Systems Integrating Federated Learning with Deep Reinforcement Learning: A Survey. In Proceedings of the 2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN), Budapest, Hungary, 2–5 July 2024.
51. Bertsekas, D.P. Model Predictive Control and Reinforcement Learning: A Unified Framework Based on Dynamic Programming. *IFAC-PapersOnLine* **2024**, *58*, 363–383. [[CrossRef](#)]
52. Vu, T.M.; Moezzi, R.; Cyrus, J.; Hlava, J. Model Predictive Control for Autonomous Driving Vehicles. *Electronics* **2021**, *10*, 2593. [[CrossRef](#)]

53. Jia, Z.; Yang, Y.; Zhang, S. Towards Realistic End-to-End Autonomous Driving with Model-Based Reinforcement Learning. *arXiv* **2020**, arXiv:2006.06713.
54. Yang, Z.; Jia, X.; Li, Q.; Yang, X.; Yao, M.; Yan, J. Raw2Drive: Reinforcement Learning with Aligned World Models for End-to-End Autonomous Driving (in CARLA v2). *arXiv* **2025**, arXiv:2505.16394.
55. Uppuluri, B.; Patel, A.; Mehta, N.; Kamath, S.; Chakraborty, P. CuRLA: Curriculum Learning Based Deep Reinforcement Learning For Autonomous Driving. *arXiv* **2025**, arXiv:2501.04982. [[CrossRef](#)]
56. Chen, J.; Peng, Y.; Wang, X. Reinforcement Learning-Based Motion Planning for Autonomous Vehicles at Unsignalized Intersections. *Transp. Res. Part C* **2023**, *158*, 104945.
57. Surmann, H.; de Heuvel, J.; Bennewitz, M. Multi-Objective Reinforcement Learning for Adaptive Personalized Autonomous Driving. *arXiv* **2025**, arXiv:2505.05223. [[CrossRef](#)]
58. Tan, Z.; Wang, K.; Hu, H. CarLLaVA: Embodied Autonomous Driving Agent via Vision-Language Reinforcement Learning. Stanford CS224R Project Report. 2024. Available online: https://cs224r.stanford.edu/projects/pdfs/CS224R_Project_Milestone_5_.pdf (accessed on 30 July 2025).
59. Yang, Z.; Liu, J.; Wu, H. Safe Reinforcement Learning for Autonomous Vehicles with Uncertainty-Aware Collision Avoidance. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6312–6319.
60. Fang, Y.; Yan, J.; Luo, H. Hierarchical Reinforcement Learning Framework for Urban Autonomous Driving in CARLA. *Robot. Auton. Syst.* **2022**, *158*, 104212.
61. Cui, X.; Yu, H.; Zhao, J. Adaptive Curriculum Reinforcement Learning for Autonomous Driving in Complex Scenarios. *IEEE Trans. Veh. Technol.* **2023**, *72*, 9874–9886.
62. Feng, R.; Xu, L.; Luo, X. Generalization of Reinforcement Learning Policies in Autonomous Driving: A Domain Randomization Approach. *IEEE Trans. Veh. Technol.* **2025**, early access.
63. Luo, Y.; Wang, Z.; Zhang, X. Improving Imitation Learning for Autonomous Driving through Adaptive Data Augmentation. *Sensors* **2023**, *23*, 4981.
64. Mohanty, A.; Lee, J.; Patel, R. Inverse Reinforcement Learning for Human-Like Autonomous Driving Behavior in CARLA. *IEEE Trans. Hum.-Mach. Syst.* **2024**, early access.
65. Liang, X.; Wang, T.; Yang, L.; Xing, E. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 584–599.
66. Phan-Minh, T.; Howington, F.; Chu, T.S.; Lee, S.U.; Tomov, M.S.; Li, N.; Dicle, C.; Fidler, S.; Suarez-Ruiz, F.; Beaudoin, R.; et al. Driving in real life with inverse reinforcement learning. *arXiv* **2022**, arXiv:2206.03004. [[CrossRef](#)]
67. Ho, J.; Ermon, S. Generative adversarial IL. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Volume 29.
68. Chekroun, R.; Toromanoff, M.; Hornauer, S.; Moutarde, F. Gri: General reinforced imitation and its application to vision-based autonomous driving. *Robotics* **2023**, *12*, 127. [[CrossRef](#)]
69. Han, Y.; Yilmaz, A. Learning to drive using sparse imitation reinforcement learning. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 3736–3742.
70. Reddy, S.; Dragan, A.D.; Levine, S. Sqil: IL via reinforcement learning with sparse rewards. *arXiv* **2019**, arXiv:1905.11108.
71. Huang, X.; Chen, H.; Zhao, L. Hybrid Imitation and Reinforcement Learning for Safe Autonomous Driving in CARLA. *IEEE Trans. Intell. Transp. Syst.* **2024**, early access. [[CrossRef](#)]
72. Kim, J.; Cho, S. Reinforcement and Imitation Learning Fusion for Autonomous Vehicle Safety Enhancement. *IEEE Trans. Intell. Veh.* **2025**, early access.
73. Li, Z.; Zhang, S.; Zhou, D. Behavioral Cloning and Reinforcement Learning for Autonomous Driving: A Comparative Study. *IEEE Intell. Transp. Syst. Mag.* **2022**, *14*, 27–41.
74. Cheng, Y.; Wu, J.; Wang, Z. End-to-End Urban Autonomous Driving with Deep Reinforcement Learning and Curriculum Strategies. *Appl. Sci.* **2023**, *13*, 5432.
75. Hofbauer, M.; Kuhn, C.; Petrovic, G.; Steinbach, E. TELECARLA: An Open Source Extension of the CARLA Simulator for Teleoperated Driving Research Using Off-the-Shelf Components. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020.
76. Yue, X.; Zhang, Y.; Zhao, S.; Sangiovanni-Vincentelli, A.; Keutzer, K.; Gong, B. Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization without Accessing Target Domain Data. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2100–2110. [[CrossRef](#)]
77. Kaur, M.; Kumar, R.; Bhanot, R. A Survey on Simulators for Testing Self-Driving Cars. *Comput. Sci. Rev.* **2021**, *39*, 100337. [[CrossRef](#)]
78. Bu, T.; Zhang, X.; Mertz, C.; Dolan, J.M. CARLA Simulated Data for Rare Road Object Detection. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 2794–2801. [[CrossRef](#)]

79. Zeng, R.; Luo, J.; Wang, J. Benchmarking Autonomous Driving Systems in Simulated Dynamic Traffic Environments. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 121–132.
80. Jiang, X.; Zhao, H.; Zeng, Y. Benchmarking Reinforcement Learning Algorithms in CARLA: Performance, Stability, and Robustness Analysis. *Transp. Res. Rec.* **2025**, *2025*, 247–258.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.