



Autonomous Vehicle Decision-Making and Control in Complex and Unconventional Scenarios—A Review

Faizan Sana ¹, Nasser L. Azad ^{1,*} and Kaamran Raahemifar ^{2,3,4}

¹ Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; faizan@uwaterloo.ca

² Data Science and Artificial Intelligence Program, College of Information Sciences and Technology (IST), Penn State University, State College, PA 16801, USA; kraahemi@psu.edu

³ School of Optometry and Vision Science, Faculty of Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada

⁴ Department of Chemical Engineering, Faculty of Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

* Correspondence: nlashgarianazad@uwaterloo.ca

Abstract: The development of autonomous vehicles (AVs) is becoming increasingly important as the need for reliable and safe transportation grows. However, in order to achieve level 5 autonomy, it is crucial that such AVs can navigate through complex and unconventional scenarios. It has been observed that currently deployed AVs, like human drivers, struggle the most in cases of adverse weather conditions, unsignalized intersections, crosswalks, roundabouts, and near-accident scenarios. This review paper provides a comprehensive overview of the various navigation methodologies used in handling these situations. The paper discusses both traditional planning methods such as graph-based approaches and emerging solutions including machine-learning based approaches and other advanced decision-making and control techniques. The benefits and drawbacks of previous studies in this area are discussed in detail and it is identified that the biggest shortcomings and challenges are benchmarking, ensuring interpretability, incorporating safety as well as road user interactions, and unrealistic simplifications such as the availability of accurate and perfect perception information. Some suggestions to tackle these challenges are also presented.

Keywords: autonomous vehicles; self-driving cars; unconventional scenarios; decision-making; path planning; motion control; intelligent transportation systems



Citation: Sana, F.; Azad, N.L.; Raahemifar, K. Autonomous Vehicle Decision-Making and Control in Complex and Unconventional Scenarios—A Review. *Machines* **2023**, *11*, 676. <https://doi.org/10.3390/machines11070676>

Academic Editor: Yahui Liu

Received: 22 May 2023

Revised: 8 June 2023

Accepted: 16 June 2023

Published: 23 June 2023



Copyright: © 2023 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

Several organizations, both in academia and industry, are rigorously working towards developing autonomous vehicles (AVs) which have the potential to save thousands of lives every year and create significant societal benefits [1,2]. According to a report by the World Health Organization (WHO), approximately 1.35 million road traffic deaths occur every year with an additional 20–50 million people suffering nonfatal injuries [1]. A study carried out by the National Highway Traffic Safety Administration (NHTSA) in the United States concluded that 94% of accidents occurred due to human errors while only 2% were caused by vehicle failures [3]. Hence, encouragingly, the increased adoption of AVs will likely reduce vehicle accidents and hence decrease the fatalities due to road traffic. If their widespread deployment is successful, the projected annual social benefits of AVs—which include reducing traffic congestion and the number of accidents on the road, consuming less energy, and boosting productivity as a result of reallocating driving time—will reach nearly \$800 billion by 2050 [4].

Although the technology for vehicles moving in static environments is well developed [5], the dynamic nature of the real-world environment has made it extremely challenging for widespread AV adoption [6]. To experiment with AV technologies in the real world, the

Department of Motor Vehicles (DMV) in California started issuing permits to manufacturers in 2014 under the Autonomous Vehicle Tester program [7]. California's DMV requires manufacturers to test AVs and report any collision that resulted in bodily injury, property damage, or death. As of 24 June 2022, 483 collision reports have been filed since the start of the program in 2014 [8].

Since these reports are individually filed by each manufacturer, an analysis needs to be conducted to understand the trends. Based on the analysis conducted by [9] on the data between 2014 and 2019, it was realized that the most challenging situation for AVs occurred when there were changes in the road surface conditions. For instance, when the road is wet, AVs are more vulnerable to accidents. The study used the Pearson chi-square test for calculating the relation between various elements and considered both manual override and autonomous modes. Although the authors did discuss hit-and-run cases, there was no mention of how AVs respond to such situations. The study also demonstrated that none of the AVs struck any stationary vehicle or object, regardless of the weather and road conditions. Although this is remarkable, it is also expected due to the extensive research being conducted on AV perception tasks as well as improvements in perception data acquisition technologies (camera, LiDAR, etc.). Moreover, 38% of the accidents occurred when AVs were in manual mode since the driver often disengaged the autonomous mode during erratic situations. It should also be noted that [9] found that 62% of the collisions were rear-ended, implying that the following vehicle was at fault. A chart showing the types of collisions and their percentages for the years 2014–2019 is indicated in Figure 1. A more recent study [10] suggests that these rear-ended crashes are likely due to the conservative behavior of AVs. An analysis combining the collision and disengagement statistics will provide further insight into the performance of AVs and their shortcomings.

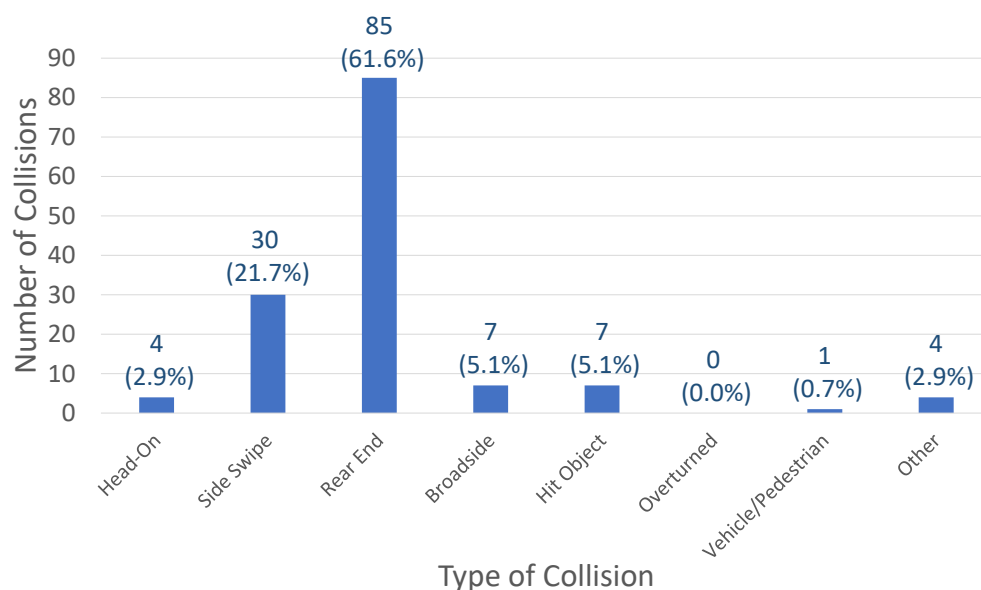


Figure 1. California AV collision statistics for 2014–2019 based on the defined 8 types of collisions [8,9].

1.1. Levels of Automation

In 2014, the Society of Automotive Engineers (SAE) released a seminal document that outlined a comprehensive framework for categorizing the levels of automation in AVs. Over the years, this document has undergone several revisions, with the most recent update occurring in April 2021 [11].

The SAE's classification system divides automation into six distinct levels, each delineating the extent of human involvement in the driving process. Levels 0–2 place the primary responsibility on the human driver, with automation providing supplementary support. In contrast, levels 3–5 introduce advanced automated driving features, wherein

the vehicle assumes a more dominant role. Figure 2 provides a visual representation of these levels, while a brief description of each level is presented below:

- **Level 0: No driving automation.** The vehicle relies entirely on the human driver, with no automated assistance available.
- **Level 1: Driver assistance.** The vehicle offers limited assistance, such as steering or brake/acceleration support, to the driver.
- **Level 2: Partial automation.** The vehicle provides both steering and brake/acceleration support simultaneously, but the driver remains responsible for monitoring the driving environment.
- **Level 3: Conditional automation.** The vehicle can manage most aspects of driving under specific conditions, but the driver must be ready to intervene when necessary.
- **Level 4: High driving automation.** The vehicle is capable of operating without driver input or intervention but is limited to predefined conditions and environments.
- **Level 5: Full automation.** The vehicle operates autonomously without requiring any driver input, functioning in all conditions and environments.

More detailed information about each level can be found in the official SAE J3016 document [11]. In the context of this research study, our focus lies on evaluating the capabilities of AVs at the highest level of automation, namely, SAE level 5.



SAE J3016™ LEVELS OF DRIVING AUTOMATION™

Learn more here: [sae.org/standards/content/j3016_202104](https://www.sae.org/standards/content/j3016_202104)

Copyright © 2021 SAE International. The summary table may be freely copied and distributed AS-IS provided that SAE International is acknowledged as the source of the content.

	SAE LEVEL 0™	SAE LEVEL 1™	SAE LEVEL 2™	SAE LEVEL 3™	SAE LEVEL 4™	SAE LEVEL 5™
What does the human in the driver's seat have to do?	You <u>are</u> driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You <u>are not</u> driving when these automated driving features are engaged – even if you are seated in “the driver’s seat”		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
Copyright © 2021 SAE International.						
	These are driver support features			These are automated driving features		
What do these features do?	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	<ul style="list-style-type: none">• automatic emergency braking• blind spot warning• lane departure warning	<ul style="list-style-type: none">• lane centering OR• adaptive cruise control	<ul style="list-style-type: none">• lane centering AND• adaptive cruise control at the same time	<ul style="list-style-type: none">• traffic jam chauffeur	<ul style="list-style-type: none">• local driverless taxi• pedals/steering wheel may or may not be installed	<ul style="list-style-type: none">• same as level 4, but feature can drive everywhere in all conditions

Figure 2. Levels of driving automation as defined by SAE J3016 (2021). Figure courtesy of [11]. Copyright ©2021, SAE International.

1.2. Selecting Edge Cases

One of the major concerns regarding the safety of AVs is that although they can perform relatively well in common day-to-day scenarios, they face lots of issues when it comes to edge cases [12]. This is mainly due to the lack of data on these edge cases in the real world and less exposure to these specific scenarios. Since there is a large magnitude of edge cases to account for, for the purpose of this review paper, it was decided to select only the five most pertinent ones.

Based on an analysis of currently deployed AVs in California [9,10] and the 2020 traffic safety fact report by the NHTSA [13], a list of the five most common edge cases with respect to their occurrences and fatalities was compiled. Based on [9,10], it was found that current AVs struggled the most with changes in road conditions, which was closely associated with *adverse weather conditions* (Section 2). Additionally, it was found that 38% of the accidents occurred during emergencies when the driver had to enable manual override which was labeled as *near-accident scenarios* (Section 6). Using the traffic accident statistics for 2020 [13], it was found that *intersections* (Section 3), *pedestrian crosswalks* (Section 4), and *roundabouts* (Section 5) were the ones where human drivers faced the most difficulties. A chart summarizing the accidents occurring at intersections can be seen in Figure 3.

Vehicle Car Crashes by Relation to Intersection in the US in 2020

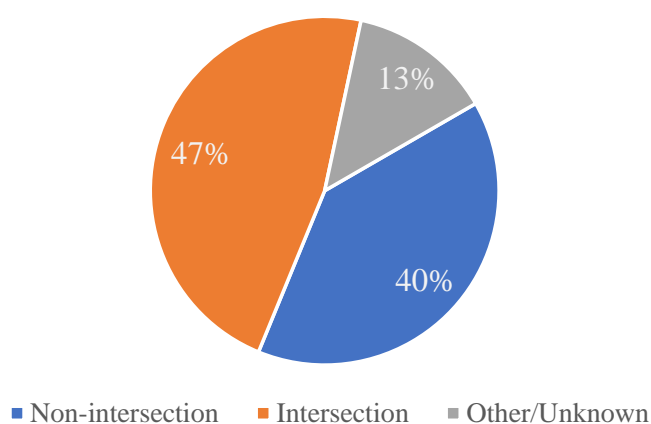


Figure 3. Vehicle car crashes in the US in 2020 by relation to intersections [13].

This review paper lists these scenarios and discusses the various planning and control techniques used to tackle the issues presented, along with their shortcomings. Hence, as compared to other surveys which are mostly divided based on the navigation algorithm used (e.g., reinforcement learning, model predictive control, etc.) [14–17], this paper is more concerned with the situations that these algorithms can handle. Rather than focusing on the methodology or the type of algorithm used, this article provides a more comprehensive approach to deal with the considered scenarios and identifies overall shortcomings that should be addressed for each scenario. In addition, the scenarios are selected based on an analysis of the current AV shortcomings based on those deployed in California [9,10], as well as the latest traffic accident statistics obtained from the NHTSA [13]. Additionally, this survey paper is organized in a way such that it is readable by researchers from other fields or those relatively new to the AV area. Moreover, this paper mostly uses studies conducted after 2018 and is therefore representative of the latest trends in each scenario.

The paper is divided into multiple sections based on the specified scenarios. Each section contains an introduction and concludes with a summary of the surveyed papers in a table identifying the advantages and disadvantages of each one. The paper ends by discussing the challenges and future directions (see Section 7). A schematic of this arrangement is shown in Figure 4.



Figure 4. Organization of the paper sections.

2. Adverse Weather Conditions

Continuously changing weather phenomena have several negative impacts on traffic and transportation. On average, global precipitation occurs 11% of the time although only 8% occurs on land [18]. Based on the study conducted in [19] which used information from fatal crashes spanning the US over a 6-year period, the authors concluded that the risk of fatal crashes increased by 34% during active precipitation while the risk was the highest during the winter months. In terms of their classification of heavy precipitation, they found that the risk increased by over 140%. Such adverse weather phenomena include rain, snow, fog, wind, and extreme heat and cold. Since such weather conditions affect the perception stack of autonomous vehicles, most studies such as [20–22] are concerned with improving the sensor fusion algorithms in which data from multiple sensors, for instance, radar, LiDAR, RGB, event-based cameras, etc., are fused to get the most accurate localization and identification of the surrounding objects. Therefore, most surveys on adverse weather conditions [23–26] focus on improving the shortcomings in the perception stack and the relevant hardware, discussing the various sensors' weaknesses. Although identifying the weather condition is crucial and hence there is an abundance of research in the perception domain, it is also important that AV controllers take into consideration these changes since they affect the friction coefficient of the road. These changes negatively afflict the ego vehicle directly because they create abnormalities in the vehicle states. These changes also affect the surrounding vehicle's behavior and hence the environment dynamics itself. According to [27], for braking distances of vehicles traveling at 80 km/h, the friction coefficient for dry roads is 0.7 while that of rainy, snowy, and icy roads is 0.4, 0.28, and 0.18, respectively. These data along with the collision data demonstrate that the behavior of AVs need to adapt to these changes in road conditions. Although identifying the effects of changes in weather on the road friction coefficient is essential, this paper focuses on the navigation, planning, and control, and thus the reader is encouraged to go over research reported in [28–31] for further details on road coefficient detection methods.

A motion planning algorithm was developed in [32] that took into consideration vehicles with a limited receptive field, which usually occurs during adverse weather conditions. The algorithm used a probabilistic model to estimate the likelihood of obstacles being present in the unobserved areas of the environment, incorporating that information into the motion planning algorithm for generating safe and efficient trajectories. The planner itself was based on finite state machines (FSM). The planner was tested in a closed-loop simulation environment developed by the authors and deployed on their automated vehicle, BERTHAONE [33]. It was shown that the planner imitated human behavior and drove with reduced speeds in shorter sensor ranges, preparing to yield to approaching not yet observable vehicles, while being aware of any erratic behaviors in other vehicles. However, they only tested their algorithm in an environment consisting of other vehicles and made several assumptions, such as all other vehicles obeyed the speed limit and the false negative object detection rate was zero for a specific range amongst others. Additionally, due to the nature of the FSM algorithm, complex maneuvers such as lane changes were not allowed since that would have caused the required FSM logic to exponentially grow. In [34], the authors developed an adaptive path planning model for collision avoidance and lane change maneuvers for curved sections of highways. The lane change model took into account the curvature of the road, the road friction coefficient, and the presence of other vehicles. They utilized a Gaussian distribution to evaluate the impact of rain on vehicle lateral dynamics. To track and execute the control signal, they used a model predictive control (MPC) approach which incorporated the effects of rain and ensured the generated trajectory was followed. The proposed algorithm was tested in a self-developed simulation environment and shown to successfully change lanes in rain. However, this algorithm was specifically tested on highways, only at curved sections, and it only outputted the lateral control of the vehicle.

In [35], a linear MPC-based stability control system was proposed for low-friction roads. The authors developed a controller for both the longitudinal and lateral controls

and demonstrated that their controller worked well even *without* knowing the value of the friction coefficient. They also developed an instability detection algorithm which was used to determine the vehicle stability threshold while estimating the friction coefficient. The algorithm was tested in a Carsim/Matlab cosimulation environment and compared with a baseline MPC for only tracking the desired path and speeds. Similarly, in [36], the authors proposed a steering control tracking strategy for roads with different tire–road friction coefficients. To estimate the friction coefficient, they developed a long short-term memory (LSTM) network which consisted of four LSTM layers and two fully connected (FC) layers. They evaluated their algorithm in a Carsim-Simulink simulation environment and demonstrated that the tracking accuracy improved by 37.7% with an adaptive tire cornering stiffness estimation strategy. However, since they were only focusing on the tracking, the algorithm only outputted the steering angle. In [37], the effects of sudden snowstorm conditions on AV dynamics were discussed, along with the controller performance on the same map. An MPC controller was used for trajectory tracking with the maximum speed set to 35 km/h. The authors showed that the average positioning error using the mean squared error (MSE) formulation in snow was 22% higher than in nonsnow conditions, particularly due to wheel slippage. As a result, the vehicle was still able to navigate autonomously. They concluded that the robustness of the positioning strategy was key to the performance of the motion controller, indicating the importance of a robust positioning strategy. However, that study was only limited to smaller-sized vehicles since they were using a heavy quadricycle, the Renault Twizy, and further analysis of vehicle dynamics was needed to be performed for larger-size vehicle applications, such as autonomous buses and trucks.

In [38], an adaptive online tuning strategy for lateral controllers was introduced, particularly the Stanley controller, to improve the performance of lateral trajectory tracking systems. This is shown in Figure 5. The authors used a modified version of the Stanley control law, originally used in [39], which essentially incorporated the vehicle kinematics as well as partial inclusions of the vehicle dynamics effects (tires and steering wheel). They used a fuzzy inference algorithm to update the controller gains online to keep the cross-tracking errors (CTEs) as low as possible. The updated gains were the CTE and heading gains of the Stanley controller.

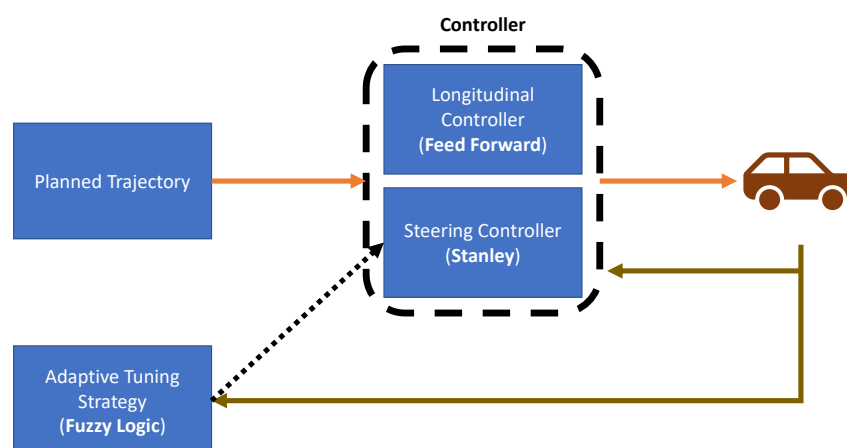


Figure 5. Motion control framework used in [38].

Furthermore, to replicate driverlike behavior, they proposed a set of rules for adjusting the gains based on a heuristic understanding of how the controller gains affected the steering angle. Experiments were conducted by randomly initializing the controller gains and monitoring the performance of an adaptation strategy by using CTEs as the performance metric. Experiments were carried out in a confined environment and the desired trajectory of the vehicle was recorded and then used for testing. However, since there were no obstacles, no local path planning was performed. Additionally, the average

speed of the vehicle was 15 km/h which did not truly represent urban driving. There were also no experiments performed on slippery roads which would change the coefficient of friction. Compared to traditional lane-keeping systems (LKS), which are typically based on a camera, in [40], an LKS was presented using a global navigation satellite system (GNSS) along with high-definition (HD) maps to overcome the effects of the surrounding environment, including weather and lighting. They also developed an MPC-based controller to calculate the steering angle based on cubic approximations of the position of the vehicle using the GNSS and HD map outputs. In a previous study [41], they had used a PID controller but found that there was significant oscillation in the steering signal. To assess the proposed technique, they used a simulation environment developed in Matlab/Simulink with conditions suitable for highway driving. They also verified the controller performance using a Mitsubishi Outlander with a real-time kinematic OxtS GNSS RT3003. Although the controller model was able to track the steering angle, they did not evaluate it in various weather conditions. Additionally, it was seen that the time delay of the GNSS severely affected the performance and was a crucial factor for the MPC controller. A summary of the papers discussed in this section is shown in Table 1.

Table 1. Summary of the papers discussed in the adverse weather conditions section.

Paper	Pros	Cons
[32]	Used stochastic model to predict the presence of obstacles in unobservable areas. Tested in the real world on the BERTHAONE vehicle [33].	Complex maneuvers such as lane changes were not allowed due to FSM usage. Several simplistic assumptions such as a zero false negative detection rate for a specific range.
[34]	Incorporated the effect of rain and road friction coefficient into the optimization.	Only outputted the lateral control of the vehicle. Tested only on curved sections of highways.
[35]	Developed controller for adapting changes in road friction even without knowing the value. Able to detect the instability of the vehicle.	Possibly computationally expensive, due to separate MPCs for lateral and longitudinal control. Tests limited to a simple simulation environment.
[36]	Used hierarchical control: upper-level MPC and low-level pinion gear target position tracking. Hardware-in-the-loop testing.	Only outputted lateral control.
[37]	Controller was still able to track with error in positioning.	Tests were limited to smaller-size vehicles, particularly a heavy quadricycle. Maximum speed set to 35 km/h, which is not realistic.
[38]	Developed an online tuning strategy. Included both vehicle kinematics and dynamics in the controller.	Only outputted lateral control. Vehicle speed was set at 15 km/h, which does not truly represent urban driving.
[40]	Tested algorithm in the real world using a Mitsubishi Outlander. Reduced oscillation as compared to a PID controller.	Did not test in various weather conditions. Time delay of GNSS severely deteriorated the performance of the controller.

3. Unsignalized Intersections

In metropolitan locations, driving across unsignalized intersections is always a challenging problem for AVs. The vehicle should decide when and how to cross the junction safely instead of just being cautious in these circumstances as typically, there is no traffic light to regulate priorities. Furthermore, these crossroads are frequently obstructed, making the vehicle's decision-making task more complex. This is also evident in the 2019 Traffic Safety report by the NHTSA [42], in which it was observed that 25% of the accidents occurred at intersections. To tackle intersection scenarios, research on the proposed solutions can be roughly classified into two categories: (a) research on traffic elements and (b) research on AV navigation. Research about part (a) focuses on signal control at intersections and topological characteristics of the road infrastructure. Some researchers have proposed using different intersection topologies through demonstrating their effectiveness to improve traffic flow while ensuring safety [43], while others focus on optimizing the control of traffic lights at intersections [44–46]. These control strategies improve traffic flow if all approaches to an intersection are not equally congested. However, regardless of the level of traffic, these methods cannot completely eliminate the stop delay of vehicles [47]. Hence, to increase traffic efficiency and reduce pollution to a greater extent, several studies focus on the autonomous driving strategy at intersections. Since AVs and human-driven vehicles are expected to be on the roads in mixed traffic conditions for the foreseeable future, only research within mixed traffic conditions was considered. Moreover, those studies focusing

on connected and autonomous vehicles (CAVs) were ignored as part of this review. The reader is referred to [48] for further discussions on the navigation of CAVs.

These navigation algorithms to traverse unsignalized intersections can be broadly categorized into four groups, namely, graph-based, optimization-based approaches, machine-learning-based methods, and approaches that combine two or more of these methods. These approaches are discussed below.

3.1. Graph-Based Approach

Graph-based methods such as A* search [49], Dijkstra, and other search algorithms are commonly used in mobile robotics for path planning purposes [5]. Since dynamic path planning is required in the case of intersections, the rapidly exploring random tree (RRT) algorithm [50] is commonly used. RRT is a sampling-based method that can find a possible path within a comparatively short amount of time and react to changes in the environment. The tree expansion phase and the path construction step are two fundamental components of the RRT method. The algorithm generates an initial random configuration and attempts to link it to the current tree during the tree expansion stage. The distance measure is used to first locate the node in the tree that is closest to the new configuration which is then used to generate a new configuration. If the new arrangement avoids collisions, it is linked to the closest node and inserted as a new node to the tree. During the path construction step, the algorithm attempts to find a path from the start configuration to the goal configuration. Starting with the goal configuration, the algorithm locates the node in the tree that is closest to the goal. It then moves upwards, being terminated when the start configuration is reached. This algorithm is shown in Figure 6.

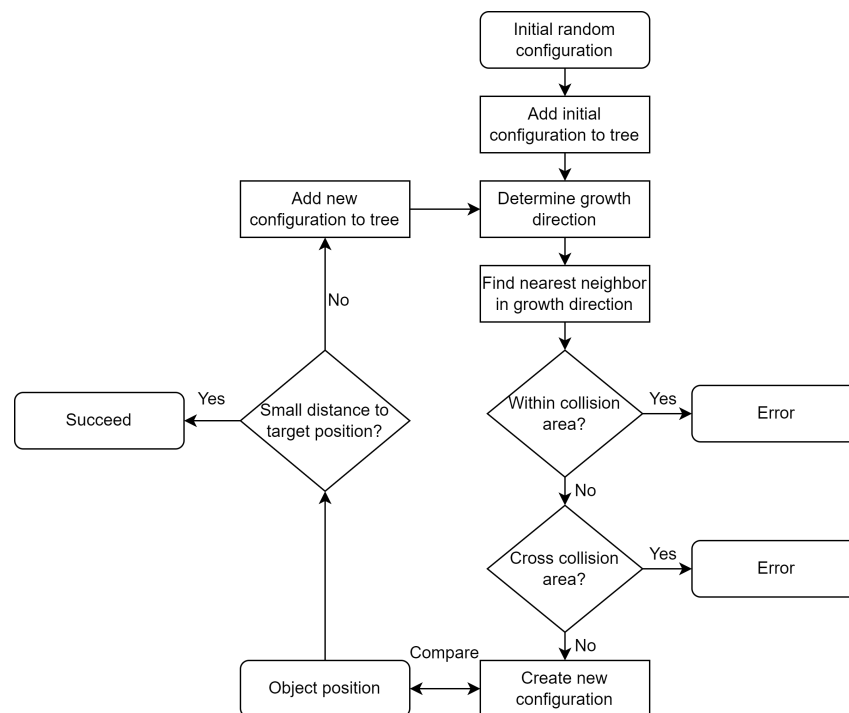


Figure 6. Flowchart of the RRT algorithm [50].

Since the proposal of RRT, several variants have been developed, including the RRT* algorithm [51] and closed-loop-RRT (CL-RRT) [52] which are often used as benchmarks. CL-RRT was developed by the MIT team for the 2007 DARPA Grand Challenge for their motion planning and control subsystem. As compared to the traditional RRT, the vehicle model is used to generate more feasible paths considering the kinematics during the path generation process. In [53], a faster version of the RRT algorithm was introduced by developing a rule-template set based on traffic scenes and an aggressive extension strategy of the search

tree itself. These rules were generated offline based on the context of the traffic scenes and saved as templates to be selected based on the short-term goal state. The search tree was repeatedly regenerated at high frequency to enable obstacle avoidance. In [54], the authors proposed a spline-based RRT* approach where the minimum turning radius (or kinematics) of vehicles was satisfied using cubic Bezier curves. They approximated the shape of the vehicle as an oriented rectangle and used it to confirm the required space of a moving vehicle. By using Bezier curves, they were able to ensure that the generated paths were continuous and satisfied the constraints of the vehicle motion [55]. In [56], a prediction algorithm based on Gaussian process regression (GPR) to predict the future locations of vehicles was combined with the RRT algorithm for motion planning. The proposed method was evaluated in simulations at a four-way intersection and the capabilities of fusing probabilistic maps with sampling-based planning methods was demonstrated.

Overall, the main issue of graph search algorithms is that, although they work well in static environments, with intricacies and swift transitions in intersection environments, they usually produce conservative decisions. The refresh rate of the planner can also be challenging, particularly in critical corner-case scenarios. Nonetheless, algorithms based on graph search methods are interpretable when failures occur.

3.2. Optimization-Based Approaches

Another common idea for navigation at unsignalized intersections is to formulate the situation as a real-time optimization problem. This involves setting up a cost function, boundary conditions, and constraints. One of the most popular methods used in this category is the MPC approach. The key components of an MPC problem formulation include obstacle models, ego vehicle models, and a proper optimization solver.

In [57], a Monte Carlo simulation was created to predict the probabilistic occupancy of other objects on the map, and then, MPC was used to optimize the reference trajectory based on the current state of the vehicle in a hierarchical fashion. The probabilistic occupancies of the road traffic users were computed offline and the results were subsequently used to reduce the real-time computational load. In [58], a bilevel controller was described, consisting of (a) a coordination level and (b) a vehicle level. At the coordination level, the occupancy time slots at an intersection were calculated while the control commands were given at the vehicle level. However, they made use of vehicle-to-vehicle (V2V) communications to coordinate that planning. A unified path planning approach using MPC was devised in [59] with the capability of automatically selecting appropriate parameters for various types of maneuvers. By modeling the surrounding vehicles as polygons and developing a lane-associated potential field, the authors could provide better driving comfort while ensuring safety.

Other optimal control techniques such as the Bezier curve optimization method have also been implemented for intersection traversing. In [60], the Bezier curve optimization method was used to cope with the constraints of obstacles at an intersection and solve an optimization problem through a combination of Lagrangian and gradient-based methods. To consider kinematic constraints, the authors used a nonlinear kinematic model with slip-free rolling conditions and formalized it into an optimization problem. They used the Bezier curve parameters to find a new path in the presence of an obstacle via minimizing quadratic errors between an initial reference path and the newly generated path.

Although these optimization-based approaches are deterministic, several unrealistic assumptions have to be made to formulate the optimization problem and solve it efficiently.

3.3. Machine-Learning-Based Approaches

Another common methodology is to use machine learning, particularly neural networks, due to their ability to approximate function for nonlinear systems. The authors of [61] presented a reinforcement learning (RL)-based approach, namely, deep Q-learning (DQN), to drive an AV through occluded intersections. Instead of using a sparse rewarding scheme in which rewards are based on collisions, they proposed the use of a risk-based

reward function for punishing risky situations. The risk was defined as follows: a safe stop condition where the ego vehicle could stop behind a conflict zone, and a safe leave condition wherein the ego vehicle could enter the conflict zone before another vehicle or if another vehicle had already left the zone. The RL agent learned a high-level policy where the action space only consisted of stop, drive fast, and drive slow actions. The actuation was handled by the low-level controllers, which the authors claimed improved the quality of learning and allowed for lower update rates by the RL policy. The risk-aware DQN approach was compared against collision-aware DQN and a rule-based policy in CARLA [62]. More information on the rule-based policy can be found in [61]. It was found that the collision-aware DQN approach was less stable during the training, which was done for 400 thousand training steps. During their experiments, it was seen that the rule-based policy was the most conservative while the collision-aware DQN approach was the most aggressive. They also assessed these algorithms on more challenging scenarios with dense traffic, severe occlusion, increased sensor noise, and a shorter sensor range (40 m). Risk-aware DQN had the highest success rate in all scenarios with the lowest being 80% in dense traffic. However, the intersections used in that work did not have crosswalks and every vehicle was assumed to be of the same length.

In [63], curriculum learning was used to learn driving behavior at four-way urban intersections. Curriculum learning was first introduced in [64] to speed up the learning process by first training a model with a simpler task and gradually increasing the complexity of the problem. Since designing the curriculum itself is a challenging task, the authors proposed an automatic curriculum generation (ACG) algorithm. They trained a DQN and a deep deterministic policy gradient (DDPG) algorithm using the ACG and random curricula. They also used a rule-based algorithm and the time to collision (TTC) to compare the results. The algorithms were evaluated in two scenarios: (a) intersection approaching in which the ego vehicle had to stop at the stop line and (b) intersection traversing. The algorithm trained using the ACG had the best mean reward with a success rate of 98.7% for the intersection-approaching scenario and 82.1% during the intersection traversing. It is also important to note that the ACG-based curricula required the lowest number of training steps, implying a more efficient training. However, the algorithm needed to be tested in other complex scenarios to show more robustness. Moreover, the authors only considered other vehicles as road users.

In [65], an RL agent was introduced that was aware of the effects that the ego vehicle would have on other human-driven vehicles and leveraged that information to improve efficiency. The authors trained a model of a human driver using inverse reinforcement learning (IRL) which was used to signify how a human driver would react to the actions of other vehicles. That model was then used as part of the reward function of the AV's RL agent. They tested their algorithm in simulations and demonstrated that the RL agent could be taught to let human participants go first by reversing itself at an intersection. This was a human-interpretable result that was not explicitly programmed. However, they assumed that the agent had a bird's-eye view of the environment and therefore had access to all the states. They also only considered interaction with a single human driver and argued that modeling interactions with multiple road users was not immediately clear.

3.4. Fusion of Various Methods

To deal with the drawbacks of the individual methods, some researchers have focused on combining them.

For instance, in [66], a hierarchical algorithm was introduced in which a high-level decision-maker made decisions for how the vehicle had to drive through an intersection and a low-level planner optimized the motion trajectories to be safe. The high-level decision-maker was implemented using deep Q-learning, an RL framework, while the low-level planner used MPC. The MPC method was not used directly since it would significantly add to the computational complexity due to an increasing number of vehicles at an intersection. Using the traffic configuration, the MPC module optimized the trajectory while the high-

level decision-maker solved the planning problem. The authors in [66] examined their algorithm in simulations and compared it using a sliding-mode (SM) controller. They performed two experiments, an intersection with a single crossing and one with a double crossing, and benchmarked it using the success rate and the collision-to-timeout ratio (CTR). Table 2 shows the results of their experiments.

Table 2. Results of the unsignalized intersection experiments conducted in [67].

Controller	Success Rate		CTR	
	Single	Double	Single	Double
SM	96.1%	90.9%	72%	93%
MPC	97.3%	95.2%	45%	76%

As seen in Table 2, the algorithm with MPC outperformed the SM in both cases. In addition, they found that the agent involving MPC converged much faster due to the immediate reward available to the RL policy (10^4 vs. 4.10^4 training episodes). However, they limited the scope of the problem by only considering at most four vehicles at any moment. They also focused on the longitudinal control while assuming the lateral control already existed. Additionally, they only considered other vehicles although they did mention that the method could be extended to other road users. An enhancement to the model would be the incorporation of a safety layer after the decision-making algorithm to limit the acceleration values for the system to stay safe. A collision avoidance system that uses the environment status and desired accelerations to determine if the present path has a collision risk and enables much better bounds to prevent the collision would also likely enhance the outcomes. A summary of the papers discussed in this section is shown in Table 3.

Table 3. Summary of the papers discussed in the unsignalized intersections section.

Paper	Pros	Cons
[53]	Allowed for obstacle avoidance due to the high-frequency regeneration of a tree.	Computationally expensive. Required much offline computation.
[54]	Ensured that paths generated were continuous and satisfied vehicle constraints. Tested in the real world.	Assumed that vehicles followed the path exactly with minimal error.
[56]	Fused stochastic maps with a sampling-based RRT algorithm. Tested at a 4-way unsignalized intersection.	The model was created for predicting other vehicle's future locations and could not account for varying vehicle speeds.
[57]	Incorporated a safety process within the decision-making stage. The whole process of planning and safety assessment took less than 100 ms.	Assumed other vehicles kept a constant velocity along the road section. They did not account for unusual events such as jaywalking, sudden reversing, etc.
[58]	Improved stability and performance at intersections by using a bilevel controller. Demonstrated that the controller had a high performance even in cases of high positioning error.	Used V2V communication which does not always exist in current systems. Controller needs to be tested in the situation of continuously oncoming vehicles.
[59]	Allowed maneuvers such as ramp merging, lane change, etc., to be determined by the MPC generated path. Infused both safety as well as comfort within the MPC constraints.	Assumed fully observable environment.
[60]	Low computational cost of replanning.	Assumed no noise in perception.
[61]	Accounted for occlusions that occur at intersections. Defined a risk-based reward function instead of a sparse rewarding scheme.	Discretized action space (3 actions). Assumed all other vehicles were of the same length.
[63]	Used curriculum learning [64] to speed up training. High success rate for both intersection-traversing and -approaching scenarios.	Considered traffic users to only consist of vehicles. Required further testing in more complex scenarios.
[65]	Used IRL and incorporated a policy into the reward function while training the RL agent.	Assumed a fully observable environment with a bird's-eye view of the environment. Only considered interaction with a single other human driver.
[66]	Used a hierarchical controller to separate high-level decision-making from lower-level control. Faster model convergence.	No safety layers. Only outputted longitudinal control, assuming lateral control existed.

4. Crosswalks

AVs must share the road with pedestrians, both in cases such as pedestrians at crosswalks and navigating through crowds of pedestrianized streets. Unlike static objects,

pedestrians have a dynamic behavior, which is extremely complicated to model, and when compared to vehicles, pedestrians generally display more randomness in their behavior [68]. Based on the surveys on pedestrian behavior modeling for autonomous driving conducted in [69,70], studies in this area can be broken down into sensing, detection, recognition and tracking, prediction, interaction, and game-theoretic models. They also emphasize that just stopping in the presence of pedestrians is inefficient and leads to the freezing robot problem. Instead, AVs must learn to interact with pedestrians using techniques similar to those used by human drivers. These techniques include understanding the behavior of pedestrians, forecasting their future behavior, anticipating their reactions to the AV's movements, and selecting appropriate motions to effectively manage the interaction.

Since everyone is a pedestrian during some time of the day, an important aspect is how AVs interact with pedestrians. Based on the US Centers for Disease Control's (CDC) WISQARS database, over 7000 pedestrians lose their lives in crashes involving a motor vehicle each year [71]. This translates to one death every 75 min and when compared with the NHTSA report [72], it shows that one in six people who died in crashes in 2020 were pedestrians. Additionally, there were approximately 140,000 visits to emergency departments from pedestrians for nonfatal injuries in 2020. As AVs become more widely used, they will need a clear pedestrian interaction control system to handle a wide range of pedestrian actions while maintaining a decent traffic flow.

Chao developed an algorithm in [73] to handle unsignalized pedestrian crosswalks. They used the windows-of-time approach wherein an AV only drove across crosswalks if its window of time did not overlap with any of the pedestrians' window of time. They calculated two properties to realize that approach: the time for pedestrians within the crosswalk to exit and the time for pedestrians to enter the crosswalk. They first evaluated their algorithm on the Unreal Engine simulator by hand-crafting various scenarios. The simulation results demonstrated that their driving policy was neither too assertive nor conservative. Additionally, Chao also provided some parameters to adjust the assertiveness of the policy. For instance, the radius of the circle which denoted the area of consideration of pedestrians could be adjusted. If the radius increased, the policy became more conservative since pedestrians further away from the crosswalk were also taken into account when calculating the window of time. They also tested one of the scenarios with the most assertive and conservative policies by adjusting the crosswalk dividing line, i.e., no division for the most conservative case. As such, the entire crosswalk was considered and divided into halves to only consider the crosswalk segment on the AV side for the most assertive case. They also compared the driving policy to that of real human drivers. This was done by recording the footage from a drone at one of the crosswalks at the University of Waterloo and replicating the same scenario in simulations. The comparison showed that the driving policy was similar to that of human drivers in most cases. However, the algorithm was not evaluated on a physical vehicle in real time. Moreover, the study only considered isolated unsignalized crosswalks and the algorithm should be revised to use, for example, four-way intersections with crosswalks. This work also did not consider how pedestrians would react when the vehicle was approaching, and the assumption was that they walked at a constant pace throughout. In addition, bicycles, strollers, and other dynamic objects at crosswalks were not considered.

In [74], the interactions of an AV with pedestrians at an unsignalized intersection with a crosswalk were considered. The authors proposed a hybrid controller that could account for several different behaviors and consisted of four distinct states: *nominal driving*, *yielding*, *hard braking*, and *speeding up*. They used a state-machine approach and specified the states and their algorithms based on the engineering specifications given in [75]. They tested their algorithm in a simulation environment created using Python and Robot Operating System (ROS) and compared it with the partially observable Markov decision process (POMDP) approach proposed in [75] used to compute a closed-loop policy using the QMDP solver [76]. Based on the simulation results, the hybrid controller was more efficient since passing through the crosswalk was permitted when there was sufficient distance

between the pedestrian and the crosswalk. This was not the case for the algorithm given in [75] since, to reduce the size of the discretized state space, the POMDP used a binary variable to represent the state of the pedestrian, whereas the hybrid controller considered a continuous state space of pedestrian distance and velocity. The authors also examined the performance of an experimental vehicle over a relatively simple road with a crosswalk using ROS. It was demonstrated that their algorithm worked well in the real world too. However, they simulated pedestrians instead of using a real person during their experiments. Similar to other studies, they assumed that the pedestrian was walking at a constant speed and only altered the decision-making, i.e., entering or exiting a crosswalk. All the test cases were also hand-generated and not based on real-world data, implying that the tests were not representative of real-world conditions. Interestingly, although they mentioned that they handled crosswalks at intersections, neither the simulation nor the real-world environment consisted of intersection traversing.

In [77], AVs and pedestrian interactions at unsignalized midblock crosswalks were investigated while taking into consideration the occlusion effects of oncoming traffic. The authors proposed a mathematical model to calculate effective and occluded visual fields for both AV and pedestrian observers. They also introduced an algorithm to allow the AV to safely handle these occluded pedestrians. They collected pedestrian data, walking speeds in particular, from a single crosswalk in Japan for a period of six hours which represented 46 samples. To represent the behavior of pedestrians, they considered both one-stage crossing and rolling gap crossing defined in [78,79]. To tackle the effects of occlusion, they proposed a memory aid decision module for tracking a pedestrian from multiple steps and storing them temporarily in a set M . The *no-yield* decision by the AV was only given if M was empty, signifying that there were no pedestrians currently within the conflict zone. It is also interesting to note that they only considered sedans, in particular hatchbacks, within the subject vehicle lane while considering SUVs and vans for oncoming traffic. They created a simulation environment using Simulation of Urban MObility (SUMO) [80] and Python to update the states of the different agents. To benchmark their algorithm against human beings and an AV algorithm without a memory aid module, they used the percentage short post encroachment time (%SPET) where PET is defined as: “the time difference between the last road user’s leaving the conflict zone and the next road user’s entering it”. However, they assumed that all pedestrians were middle-aged people with a similar height and did not consider children and elders, which are generally shorter, and therefore, likely to increase occlusion effects. They also did not take into account pedestrians walking in groups due to the sophisticated modeling required and since such data did not exist within their recorded samples. A major issue with the proposed algorithm was that it was essentially rule-based and therefore did not take into consideration the stochastic characteristics of pedestrians’ decision-making behaviors. They also found that their algorithm did not improve %SPET when oncoming traffic was stationary since the pedestrian was always occluded.

In [81], a pedestrian path prediction algorithm was proposed based on a fusion of an attention mechanism long short-term memory (Att-LSTM) network and a modified social force model (MSFM). The authors stated that the existing research mostly focused on pedestrian–pedestrian and pedestrian–static object interactions while ignoring the influence of pedestrian–vehicle interactions on pedestrian path predictions. They introduced heterogeneity in not only pedestrian interactions but also among pedestrians themselves by incorporating age and gender, since these factors were affecting their behavior. They recorded eight hours of video footage from an unsignalized crosswalk using a UAV and road-placed cameras. They also used data augmentation techniques as seen in [82] to generate 12,460 suitable pedestrians’ paths and 2534 conflicting vehicles’ paths. They trained and tested their stacked fusion model with these data and compared it with nine other existing methods. They found that the stacked fusion model had the lowest average displacement error and a final displacement error with at least an improvement of 26% in the prediction accuracy when compared to some other methods. In addition, they proposed

an MPC-based control algorithm that consumed the predicted pedestrian path output from the stacked fusion model for AV path planning. They evaluated the path planning algorithm in numerical simulations based on Simulink-Carsim and demonstrated that their method was effective for protecting pedestrians. However, obtaining gender and age information of pedestrians in a real-world setting, as required in this case, is not trivial and is still under research in the computer vision field. They also assumed that the vehicle was only traveling at a maximum speed of 30 km/h, which is not valid in many cases. Moreover, they only considered a single pedestrian at any moment, and it was not clear whether the MPC planning algorithm was evaluated in several different situations or only for one sample of the data.

Continuous action space DRL algorithms, DDPG, and proximal policy optimization (PPO) were used in [83] to plan AV trajectories involving interactions with pedestrians. The authors proposed the use of a continuous DRL algorithm to avoid the need for discretizing the action space, thus preventing issues such as inaccuracy and particularly the curse of dimensionality. They demonstrated the feasibility of both approaches and also used quadratic programming (QP) as a benchmark since QP could find an exact solution to adapt the vehicle to a particular trajectory. The reward function was designed to incorporate safety, speed, acceleration, and supporting good behavior at the end, such as smooth stopping and acceleration. Both algorithms were trained in simulations for two million steps and evaluated on two test samples. The results showed that the accident occurrences for all three algorithms were zero, with QP having the highest average car speed. It was seen that PPO gave a better reward and converged faster while DDPG had smoother acceleration and speed curves. However, if the pedestrian stopped right before the crosswalk, an accident would occur in all three algorithms. The authors suggested increasing the safety reward to deal with this issue and also using curriculum learning. They only evaluated the algorithm in a simple simulation environment with the ego vehicle and one pedestrian. Moreover, the algorithms only outputted longitudinal control.

The authors of [84] developed a behavior-aware MPC (B-MPC) method by incorporating long-term pedestrian trajectory predictions based on a previously developed pedestrian crossing model [85]. By incorporating pedestrian behavior within the control methodology, a less conservative policy could be achieved. They also compared their algorithm with a rule-based algorithm which was an FSM consisting of four states: *maintain speed*, *accelerate*, *yield*, and *hard stop*. They used a pedestrian horizon of 5 s and evaluated both the controllers for varying pedestrian time gaps and AV speeds. It was observed that the rule-based controller was unable to avoid collisions in 0.8% of the experiments (a total of 500 experiments were performed). Additionally, the B-MPC controller had an overall higher minimum distance to pedestrians as the time gap was changed. However, they assessed the controllers in a scenario with simple motion models. They also assumed that each discrete state had a constant velocity and that all of the pedestrians close to the crosswalk had the intention to cross.

In [86], an interaction-aware decision-making approach was proposed (see Figure 7). In particular, they first used game theory to model vehicle–pedestrian interactions, which were used to develop an interaction inference framework. By using this framework, they were able to obtain the intentions of pedestrians, which in turn were then used to generate collaborative actions. Next, the collaborative action was formulated as an optimization problem based on the task and the action, which was solved using MPC. They also developed an interactive environment to model aggression in external pedestrians, which was useful to address the randomness of pedestrians' behaviors. The pedestrian motion model was based on an FSM with a social force model.

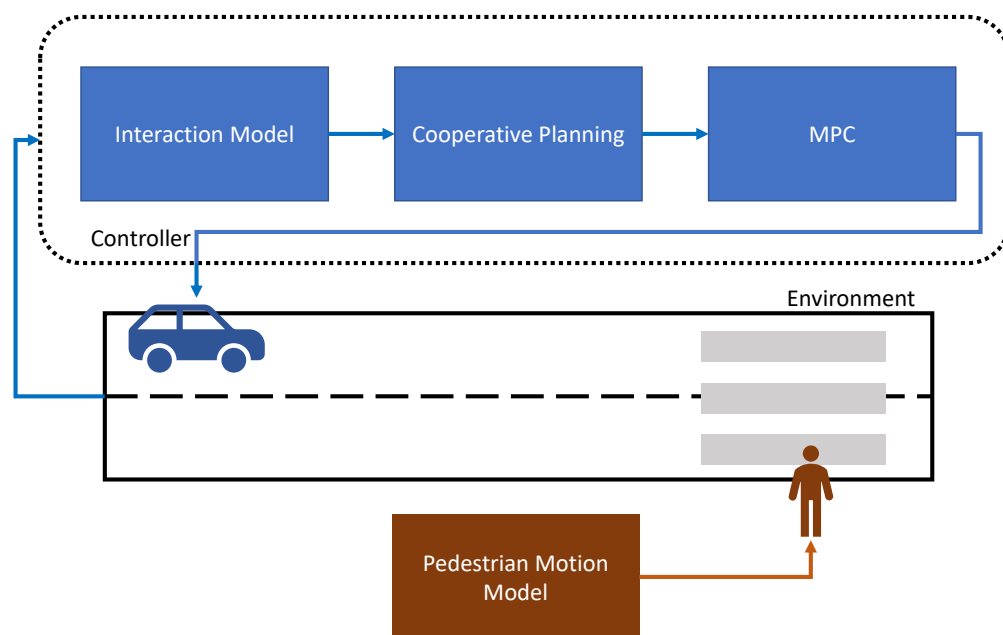


Figure 7. Decision-making framework developed in [86].

They performed experiments in simulations with only one single pedestrian and therefore more extensive tests needs to be performed. By adjusting the cooperativeness of pedestrians, they demonstrated that their algorithm worked well to successfully interact with them and follow a safe and efficient trajectory. A summary of the papers discussed in this section is shown in Table 4.

Table 4. Summary of the papers discussed in the crosswalks section.

Paper	Pros	Cons
[73]	Had parameters to adjust the conservativeness of the policy.	Assumed pedestrians walk at constant speed. Did not consider other road users such as cyclists.
[74]	Tested in the real world. Continuous state space of pedestrian distance and velocity	Assumed pedestrians walk at constant speed. All test cases were hand-generated. Even in the real world, pedestrians were simulated.
[75]	Represented as POMDP, which represents the real world closely.	Discretized the state space using a binary variable for the pedestrian.
[77]	Handled occlusions that normally occur. Considered different types of crosswalks.	Did not account for pedestrians walking in groups. Simulation environment only consisted of one type of vehicle and similar height pedestrians.
[81]	Incorporated the effect of pedestrian–vehicle interaction.	Used gender and age information of pedestrians, which is not readily available. Set constant speed of 30 km/h.
[83]	Continuous action space. Well-defined reward function incorporating safety, speed as well as end-road behavior.	Did not account for the case where a pedestrian stops at the crosswalk. Only tested with a single pedestrian.
[84]	Tested for varying pedestrian gaps and AV speeds. Higher overall minimum distance to a pedestrian as compared to the FSM approach.	Tested in a simple scenario with simple motion models. Assumed each discrete state had constant velocity and all pedestrians had the intent to cross.
[86]	Modeled aggression of external pedestrians to generate random behaviors.	Tested with a single pedestrian.

5. Roundabouts

In many regions of the world, roundabouts are becoming more prevalent and are intended to enhance traffic flow and lessen congestion. Several governments have used roundabouts as a proven solution to prevent more serious accidents that happen at or near intersections [87]. Roundabouts have various inherent characteristics that promote traffic safety compared to other types of crossings. For instance, they significantly lower speeds and also the number of potential conflict sites between road users [88]. According to the Insurance Institute for Highway Safety and the US Federal Highway Administration, roundabouts can reduce overall crashes by 37%, injury crashes by 75%, and fatal crashes by 90% compared to traditional stop-controlled intersections [89]. Yet, because of the

intricate traffic patterns and requirements to yield to other cars, navigating roundabouts is challenging, even for human drivers. Unsignalized roundabout junctions are usually thought to provide more sophisticated and challenging multivehicle interactions in urban environments than crossroad intersections [90]. Similar to other navigation algorithms, the current approaches can be classified into four broad categories: (i) rule-based (ii) collaborative scheduling-based, (iii) learning-based, and (iv) model-based. Most rule-based decision policies use TTC as a safety indication to ensure a safe gap between two vehicles because they are typically built from human knowledge and engineering experience [91]. Similar to the other cases, although most driving situations can be satisfied with such engineered systems, the designed rules lack flexibility and fail in edge cases since it is extremely challenging to account for all possible scenarios. The collaborative scheduling-based approach requires the installation of expensive infrastructure since it relies upon V2X communications and is therefore not easily scalable. Learning-based approaches obtain optimal policies by learning from expert driving demonstrations or during interactive training within a simulation environment. Model-based approaches use a model of the environment which represents the relationship between the control actions and the resulting motion in the environment and employ this to generate the appropriate control input.

In [92], three deep reinforcement learning algorithms (DRL) were developed to solve complex urban driving scenarios, a roundabout in this case. Chen mentioned that there had not been successful applications of DRL because most methods directly used a frontal camera view as the input, dramatically enlarging the sample complexity for learning. In their work, the camera images were first passed through a perception stack to identify the objects (humans, vehicles, lanes) around the AV. The outputs of that module as well as the route planner were converted to a bird's eye view, and a variable autoencoder (VAE) was used to encode it into low-dimensional latent states. These low-dimensional states were then fed to the deep neural policy to output the control commands, namely, the steering angle and the acceleration. The model-free DRL methods used were twin-delayed deep deterministic policy gradient (TD3), double deep Q-network (DDQN), and soft actor-critic (SAC). The networks were trained initially for a roundabout without vehicles and then with dense traffic. They also compared it with a baseline method in [93]. However, they did not discuss the computational resources although they did mention that the algorithms were computationally expensive. Additionally, the authors did not discuss if there was a pedestrian crosswalk at the roundabouts. SAC was the algorithm with the highest accuracy but had a success rate of just 58%.

Two of the major issues of RL are sample efficiency and the design of the reward function. Since the agent has to perform several interactions with the environment to train itself, the RL algorithms have very low sample efficiency. In [94], an RL framework was proposed to use human demonstrations to incorporate expert prior knowledge into the RL agent. They accomplished this task by first training the expert policy with behavior cloning (BC), which was then utilized to regularize the RL agent's behavior. Hence, a simplified reward function could be used even for more complicated situations. They tested their algorithm on an unprotected left turn and a roundabout without any pedestrian crosswalks. Instead of the direct vehicle control commands, i.e., acceleration and steering angle, they designed a lane-following controller involving lane changes and a target speed. To benchmark their algorithm, they also trained the following four algorithms: SAC, BC, PPO, and generative adversarial imitation learning (GAIL) for the same task. SMARTS [95] was used as the simulation platform and it was seen that the proposed algorithm had the highest success rate in both scenarios. The state-of-the-art RL algorithm, SAC, indicated jerky motions while the proposed algorithm generated comfortable accelerations and also improved the sample efficiency by 60% compared to that of SAC. Interestingly, it was noticed that the algorithm found it easier to learn aggressive left-turn behaviors as compared to passive left-turn behaviors. The proposed algorithm achieved an accuracy of 96% for the unprotected left turn and 84% for the roundabout as compared to 78% and 74% for SAC, respectively. However, the proposed algorithm used high-level human decisions

(change lane, speed up by 2 m/s, slow down by 2 m/s) from human demonstration data which would not scale to real-world scenarios since only trajectories are available rather than these high-level human decisions. In addition, the proposed algorithm introduced more hyperparameters, making tuning more time-consuming.

In [96], GAIL was applied, but rather than learning low-level controls, the authors proposed a hierarchical model with a safety layer wherein high-level decision-making logic was learned by the model, and then low-level controllers were used to realize the logic. The algorithm, safety-aware hierarchical adversarial imitation learning (SHAIL), in this case, selected from a set of possibilities that targeted a specific velocity at a future time. The low-level controller then used fixed accelerations to bring the AV to the desired velocity while the safety layer ensured that the ego vehicle did not collide with other vehicles if this velocity was maintained. The authors created an “Interaction Simulator”, which was an OpenAI Gym simulator [97], which used the Interaction dataset [98], consisting of complex urban driving scenes, mostly situations involving roundabouts and intersections. They performed two experiments: in-distribution where the model was trained and tested in the same environment, and out-of-distribution wherein the training and test environments were different but based on the same sequential scene. Their experiments focused on the performance in roundabouts. They also performed the same experiments on an expert model which was the default movement from the dataset, Intelligent Driver Model (IDM), which is based on a vehicle-following algorithm, BC, GAIL, and HAIL wherein the safety aspect was removed. Although SHAIL was the best one in terms of success rate in both experiments, it only had a success rate of 70.5% and 60.5% for experiments 1 and 2, respectively. It is important to note that although the testing and training environments in experiment 2 were slightly different, the environments were not completely unseen since the roundabout was identical to the one used in the training but with newly generated vehicle data. Another concern was that the nonego vehicles in the simulation environment did not adjust themselves if the ego vehicle was in front since they were driven using predetermined data. Hence, the ego vehicle, as part of SHAIL, had to ensure that it was not rear-ended. The low-level controllers used in this case were jerky and provided very simple options. Implementing more advanced low-level controllers and better termination criteria and safety predictors would improve the results. Additionally, the current experiments needed to be scaled wherein the model would be trained on several different scenarios and then tested in other situations not seen in the training.

In [99], a modified version of asynchronous advantage actor–critic (A3C) [100], a DRL model, was proposed, which they named delayed A3C (D-A3C) since the neural network update of the asynchronous agents’ policies were done at the end of the episode rather than in short time intervals as in A3C. They also had an *aggressiveness* input to the system which could be adjusted by the user. The algorithm outputted the following: *permitted*, which allowed the ego vehicle to enter with maximum comfortable acceleration α_{max} , *not permitted*, in which the ego vehicle decelerated to the stop line, and *caution* wherein the ego vehicle had to approach the roundabout with prudence. The reward function was designed to consist of penalization for dangerous maneuvers, ensuring smooth behavior, maintaining a high speed, and reaching the destination. To test the algorithm, they created a simulation using MARL as demonstrated in [101] to ensure that the passive vehicles behaved like real-world drivers. They compared the results of D-A3C with A3C, A2C (a synchronous variant of A3C) as well as a rule-based approach. They found that D-A3C converged the fastest while A2C converged on a suboptimal solution, remaining in the *permitted* state. When compared to the rule-based approach, it was seen that D-A3C had the highest percentage of episodes ending successfully (98.9% vs. 83.1%) while the percentage of crashes were slightly higher when the rule-based approach was set to a higher distance threshold between the ego and passive vehicles. They also tested the performance on unknown roundabouts and demonstrated that the algorithm needed to include more diverse scenarios during the training to improve its generalization. To simulate the perception errors, they added noise into the observed vehicle’s states and demonstrated that the algorithm only decreased by

2%. However, future efforts should enforce these characteristics to make the system viable in both real-world and unknown contexts.

In [102], DRL was combined with imitation learning (IL) from expert demonstrations for determining longitudinal vehicle control. By adding the IL to the policy update in addition to RL, the authors enabled the model to learn from both human demonstration and the agent's self-exploration. They used the SAC algorithm and employed dynamic experience replay to adjust the sampling ratio between the agent's self-exploration and expert demonstrations to speed up the learning phase. Although the authors proposed it for general urban autonomous driving, their tests were exclusively performed in urban roundabout scenarios. They used five baseline algorithms: (i) DQN, (ii) PPO, (iii) TD3, (iv) A3C, and (v) SAC and three IL baseline methods: (i) BC, (ii) soft-Q imitation learning (SQIL) [103], and (iii) GAIL. They tested these algorithms in CARLA [62] and compared the results with a rule-based algorithm that would follow the default rules defined in CARLA. They found that their algorithm had the highest success rate (90%) as compared to the other algorithms, with BC having the lowest success rate at 16%. To further increase the robustness, they also developed a rule-based safety controller fused with their algorithm. This safety controller would take over the vehicle control during near-collision situations and increased the success rate further by 3%, to 93%. However, they only developed the algorithm to manipulate the longitudinal control and assumed that there was no noise in the perception information. They also disregarded occlusions and assumed the environment information was available in a bird's-eye view map.

In [104], the use of HD maps to predict future events at the lane level was proposed to foresee how other drivers would react within roundabouts. They provided an approach that leveraged road occupancy data to anticipate lane-level events. To account for inaccuracies in the localization, they widened those occupancy data. They determined an appropriate gap for the roundabout entrance maneuver based on the occupancy intervals to make sure the priorities were met while balancing the safety and efficiency within the policy. They evaluated their algorithm first within SUMO [80] and used the INTERACTION dataset [98] to generate dynamic traffic flows while using ROS for the navigation algorithms and AV dynamics. They also discussed the drawbacks and oversimplifications typically present in simulators and hence performed experiments on a vehicle in the real world. Since the algorithm only outputted the longitudinal control, they employed a simple lane-keeping lateral controller for adjusting the steering angle. However, they assumed that vehicles within the roundabout drove at a constant speed, and to simplify the navigation algorithm, they did not allow for lane changing (and overtaking) once the ego vehicle had entered the roundabout.

In [105], a driving policy based on the SAC algorithm combined with interval prediction and self-attention mechanism (IP-SAC) was presented to traverse unsignalized roundabouts. To maximize data usage and efficiency during the learning process, they used experience replay and randomly selected a batch of 256 items from the replay buffer. The interval prediction model determined accessible regions for obstacle vehicles and forecasted where they would be in subsequent time steps. Using this information, the ego vehicle could learn to avoid these locations of possible collisions. They performed training within a low dimensionality simulation platform, Highway-Env [106], and subsequently on CARLA, a high-dimensionality platform. The IP-SAC algorithm was compared to the SAC fused with a self-attention network [107] and they found that by fusing both the interval prediction and the self-attention network, their proposed approach had a 15% improvement in collision rate. However, they assumed that perfect perception information was available (other vehicles' positions) and only trained and tested in a two-lane roundabout. They also suggested combining their algorithm with the approach given in [108] to potentially incorporate active calculations of the road safety index within the training framework.

In [109], a nonlinear MPC (NMPC) controller was proposed to speed up the training of the DDPG algorithm. A major concern for training DRL algorithms is low sample efficiency, which means that such algorithms have to be trained for a large number of

episodes, significantly increasing training time. The proposed NMPC controller ran in parallel with the DDPG algorithm and determined the appropriate control action during the training phase. The authors used two main strategies to switch between the two algorithms, namely, ϵ -annealing and Q-learning switching. They further extended the Q-learning switching with Nash Q-value updates and double replay memory which stored desirable and undesirable experiences separately. They then sampled a larger batch from the desirable transitions during the training of the actor and critic networks, which allowed the algorithm to find a more appropriate policy and avoided being trapped in local minima. The reward function was designed for multiple objectives: to lower jerk and collision rate, and to ensure a smooth traffic flow by using a step penalty. They evaluated their algorithm in a simulation environment created using SUMO for both a roundabout and merging scenario and demonstrated that the proposed approach had a 159% higher mean reward as compared to using DDPG directly. However, they used a relatively simple simulation environment and found that several collisions occurred during the testing phase. Therefore, further investigation of safety-related performance was required.

In [110], similar to [109], the authors proposed the use of an NMPC controller during the training phase of the DDPG agent. However, in that case, the NMPC controller was used for the safe exploration of a DDPG agent during both the training as well as test phases. The DDPG agent would output an action that would be ingested by the NMPC controller to compute the appropriate control input based on the reference input, specified constraints, and any other factors such as fuel consumption. The authors demonstrated that incorporating the proposed safety controller during the training phase resulted in a 9.85% increase in the mean reward and an overall mean reward increase from -75 to 7.8 as compared to the DDPG agent alone. They also incorporated noise within the states and demonstrated that their algorithm was robust to such uncertainties. However, they did not take into consideration vehicle dynamics and used a simple simulation environment developed in SUMO. A summary of the papers discussed in this section is shown in Table 5.

Table 5. Summary of the papers discussed in the roundabouts section.

Paper	Pros	Cons
[92]	Used low-dimensional latent states rather than RGB images. Performance compared against 3 DRL algorithms alongside baseline	Computationally expensive. Low success rate of 58%.
[94]	Combined expert knowledge with agent learning. Outputted smooth motion. Tested on roundabouts as well as unprotected left turns.	Used high-level human decisions such as lane change as input. Increased the use of hyperparameters so tuning was time consuming
[96]	Hierarchical model with a safety layer to avoid collisions. Used real-world data to test in simulation.	Low success rate. Training and testing environments were similar.
[99]	Incorporated an <i>aggressiveness</i> parameter to control conservativeness. Tested system with perception system noise.	Needs further testing in unknown environments.
[102]	Incorporated expert knowledge with agent self-exploration. Similar to [94]. Compared against 5 DRL and 3 IL methods. Incorporated safety controller to take over vehicle control in near-collision situations.	Assumed perfect perception information was available. Only manipulated longitudinal control.
[104]	Accounted for inaccuracies in localization. Tested the algorithm on a vehicle in the real world.	Only manipulated longitudinal control. Assumed that vehicles within a roundabout drove at constant speed.
[105]	Combined interval prediction with DRL to allow the model to avoid these locations.	Assumed perfect perception information was available. Only trained and tested in 2-lane roundabouts.
[109]	Improved training speed by combining NMPC controller with DDPG agent. Considered the comfort of users as part of the reward function.	Simple simulation environment. No safety controller which led to collisions.
[110]	Incorporated NMPC-based safety controller during both training and testing. Demonstrated algorithm robustness to perception noise.	Vehicle dynamics ignored.

6. Near-Accident Scenarios

Another issue with the algorithms currently developed using RL or IL is that they cannot deal with rapid phase transitions which are required in near-accident scenarios. During the literature survey, it was noticed that only a few studies are covering such scenarios. We feel that most research studies conducted in this area assume that such scenarios are already handled through their algorithms and hence do not explicitly mention or evaluate their approach for such extreme cases.

In [111], an algorithm capable of transitioning between different driving styles was devised. The optimal switch was learned by an RL policy while the driving styles were learned by IL. The authors named their algorithm hierarchical reinforcement and imitation learning (H-REIL). They argued that since IL was learning relatively low-level simple policies, fewer expert demonstrations were needed and noted that it would be difficult to use RL for these policies due to difficulties in designing the reward function. IL did not fit the high-level policy since it was unnatural for a human expert to demonstrate the switching of driving modes. The algorithm was evaluated in two different simulation tools: CARLA [62] and CARLO [111] and ran on five different near-accident scenarios each with two settings, namely, difficult and easy. They compared H-REIL with the following algorithms: π^{IL} trained on a mixture of aggressive and conservative driver behaviors using IL, π^{agg} with IL only trained on aggressive behaviors, π^{tim} trained only on a conservative driving style with IL, and Π^{rand} selecting π^{agg} and π^{tim} at every time step uniformly at random. The collision rate, average episode reward, and completion time of the different methods were compared, and it was found that H-REIL was comparable to or better than the other algorithms for all of the scenarios. It is important to note that all of the scenarios were hand-designed and therefore they lacked the variability found in real-world situations. They also only evaluated their algorithm with two modes: timid and aggressive. In addition, none of the considered scenarios included pedestrians or cyclists and were only focused on accident situations involving other vehicles.

The authors of [112] proposed a safe trajectory-planning algorithm combining lateral and longitudinal acceleration vectors to ensure AVs maintains stability. Since the friction limit between the road and the tire determines the maximum acceleration that a vehicle can generate [113], it is crucial to keep the vehicle's acceleration within this limit to maintain stability and prevent slippage. As compared to a path based on global coordinates, they combined the two acceleration vectors to produce a relative trajectory. In a simulation, they assessed their approach with a single obstacle that appeared in the same lane as the ego car. They compared the performance of their algorithm to those of autonomous emergency brake (AEB) and autonomous emergency steering (AES) systems, demonstrating that their algorithm could avoid risky trajectories that AES and AEB were not able to handle. However, since evaluations were only conducted in a simple simulation environment (single lane and obstacle), a further investigation needed to be performed to ensure safe behaviors.

In [114], it was argued that although there were major research activities on emergency collision avoidance for specific scenarios such as straight or curved roads, there was no method investigated for general applications. Hence, the authors proposed a safety controller based on artificial potential function (APF) and MPC in which the controller was tasked to find a safe path for the ego vehicle in emergency situations. The APF was used as the cost function for the MPC and was adaptable to various scenarios and configurations. Even when a collision could not be avoided, the controller determined a course that would protect the area of the vehicle occupied by humans. They used 192 cases from eight different scenarios to evaluate the method and compared the results with those obtained from the Hamilton–Jacobi (HJ) reachability [115]. The HJ reachability is a formal verification method for guaranteeing the performance and safety properties of dynamic systems. The proposed MPC+AFP method had a 20% higher success rate and also decreased 43% of collisions in the driver's position. A summary of the papers discussed in this section is shown in Table 6.

Table 6. Summary of the papers discussed in the near-accident scenarios section.

Paper	Pros	Cons
[111]	Hierarchical policy using both RL and IL to simplify reward function design. Combined both human expert demonstration and agent self-exploration.	Hand-designed all the simulation scenarios. Only tested the algorithm with 2 modes, timid and aggressive.
[112]	Produced a relative trajectory as compared to a global coordinate-based path. Avoided risky trajectories where even AES and AEB systems failed.	Only tested within a single lane and single-obstacle environment.
[114]	Considered the case where collision is unavoidable (collision mitigation). Developed a general method, not limited to specific scenarios.	Tests only limited to simulation.

7. Challenges and Potential Future Research Directions

This section presents and discusses the challenges in designing decision-making and control systems for AVs in a real-world setting and also the shortcomings within the current research literature.

7.1. Benchmarking

An important aspect of the AV algorithm development is the need for metrics to benchmark and compare the performance of the algorithm [116]. Researchers can assess the efficiency, safety, and reliability of their algorithms based on such metrics, and this will further advance the development of future AV algorithms. Since there is presently no accepted universal standard for benchmarking autonomous driving decision-making and control algorithms, contrasting various techniques is particularly challenging.

There are two main ways of benchmarking AV algorithms, either in simulations or using datasets. Although datasets represent the actual dynamics of the environment, using them is a time-consuming and expensive process, and particularly for vehicles, it is difficult to obtain data that cover edge cases while abiding by the rules of the road. For instance, data in a situation where a car is drifting off the road or into oncoming lanes must be gathered to learn an appropriate response. These datasets are also inherently skewed due to the collection's locations and sensor configurations, therefore, policies trained on them will not generalize effectively without taking specific considerations. Hence, simulation tools play an important role in AV research. For further detailed information on datasets, benchmarking, and simulations, please refer to [14,117–121].

To cover unconventional, complex, and challenging scenarios, we propose the integration of datasets such as INTERACTION [98] with high-fidelity simulation tools such as CARLA and defining certain challenging scenarios to serve as a benchmark. By integrating real-world interactions along with simulation tools, the considered scenarios can be more accurately represented and used effectively for benchmarking AV algorithms.

7.2. Interpretability

Interpretability in machine learning refers to the ability to understand and explain the decisions made by a machine learning algorithm. This directly relates to the transparency which in turn relates to ethical decision making in AI. For consumers and regulators to understand how judgements are made, it is crucial that the algorithms' decision-making process is visible. This can increase public trust in technology and ensure that it is used sensibly and safely. Particularly in the case of neural-network-based algorithms, wherein the algorithm acts similar to a black box [122]. For example, [123] proposed an interpretable end-to-end RL algorithm for driving in complex urban scenarios. Using a latent state space, the driving policy was learned in tandem with a sequential environment model. As a result, the RL agent not only outputted control commands for the considered AV but also provided a semantic mask describing the map, routing information, detected objects, and the ego vehicle state [124].

Although interpretability is a general topic for most machine learning algorithms, by using rule-based techniques and hybrid approaches, which combine rule-based and machine learning approaches, a better understanding of AV control and decision-making processes based on machine learning needs to be pursued. The rationale behind each control action can be displayed to the user by combining machine-learning-based decision-makers with deterministic algorithms such as MPC as well as using safety controllers.

7.3. Safety

It is also essential to have feedback from human drivers and incorporate collision mitigation methods, similar to [114], since there will be cases where a collision is inevitable. Due to the increased usage of deep neural networks, developing such safety standards and incorporating them during both the learning and deployment phases are crucial.

Although there have been developments in verification methods for AV algorithms [125], they usually make assumptions that are not valid in the real world to speed up the process of verification or simplify it. Hence, developing a real-time safety verification algorithm that can take into account the safety considerations during the motion planning stage is an open research question.

7.4. Road User Interactions

Since human-driven vehicles and AVs will be traveling on the same roads for the foreseeable future, it is essential that vehicle flows in simulation tools represent the real-world situation. The real-life vehicle flow comprises both societal and individual behavior types, known as *social behavior* and *driving habit*. The social behavior describes how a vehicle interacts with others while the driving habit refers to the individual differences in, for example, cruising comfort and driving safety.

In addition to vehicles, the road traffic participants also include pedestrians and cyclists. It was seen that most AV control studies did not take into account pedestrians during the decision-making stage, even in cases such as intersections in which it is very common to see crosswalks. In the cases where pedestrian interactions were incorporated, only simple, low-fidelity models were used which were not representative of interactions in a real-world setting. Although there are effective models for an ideal walking behavior, high-level psychological and social modeling of pedestrian behavior is still an unresolved research problem that has to be defined on a number of conceptual levels [69]. Thus, for further research and accounting for these interactions, we propose the following:

1. The development of proper road user interaction models including pedestrian–vehicle, pedestrian–pedestrian, as well as vehicle–vehicle interactions. This could be achieved using methodologies such as game theory, which take into account interactions of various agents at each time step update of an environment.
2. Incorporating the devised models in a high-fidelity simulation tool such as CARLA to allow for the development of both modular and end-to-end decision-making and control pipelines.

7.5. Incorporating Uncertainties in Perception

Incorporating uncertainties in perception is a critical component in developing AV decision-making and control algorithms. The perception of objects, traffic signs, and road markings by AVs depends greatly on their sensors. Cameras, LiDAR, and radar are a few examples of the sensors utilized in AVs. However, all these sensors are subject to uncertainties and inaccuracies. These uncertainties may be brought on by the sensor noise, occlusions, environmental conditions, and other elements.

Since the decision-making and control process uses the output of the perception stack, it is important to take into account perception uncertainties, using them alongside the control constraints. Most of the previous work in this field, for instance [99], assumes that perfect perception information is available which is not the case in the real world. Incorporating these uncertainties enables a more accurate awareness of AV surroundings in real time, enabling the vehicle motion planning process to foresee potentially risky circumstances by taking into account unknowns at the sensor data acquisition and perception stages.

All in all, when developing new AV decision-making algorithms, it is crucial to consider these uncertainties and incorporate them in the local planning or control stages.

8. Conclusions

In conclusion, the development of AVs is a rapidly growing field that holds great promise for the future of transportation. However, navigating these vehicles in complex and unconventional scenarios is still a major challenge. It was found that the most difficult scenarios for AV algorithms, as well as human drivers, to react to were: harsh weather conditions, unsignalized intersections, traversing crosswalks, navigating roundabouts, and

near-accident situations. This survey paper provided a comprehensive overview of the AV decision-making and control techniques suggested for each of these situations.

It was realized that the main challenges that need to be addressed include: the development of benchmarking tools and datasets, ensuring the algorithm's interpretability, adding a safety layer, incorporating road user interactions, and considering uncertainties and noise for perception information within AV algorithms. Finally, although there are analyses and reports on the collision statistics of deployed AVs in California, more investigations into the causes of these accidents and relating them with the corresponding scenarios would provide further insights.

Author Contributions: Conceptualization, F.S., N.L.A. and K.R.; methodology, F.S., N.L.A. and K.R.; investigation, F.S., N.L.A. and K.R.; writing—original draft preparation, F.S.; writing—review and editing, F.S., N.L.A. and K.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was undertaken, in part, thanks to funding from the Natural Sciences and Engineering Research Council of Canada (NSERC).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

%SPET	Percentage short post encroachment time
A3C	Asynchronous advantage actor–critic
ACG	Automatic curriculum generation
AEB	Autonomous emergency brake
AES	Autonomous emergency steering
APF	Artificial potential function
Att-LSTM	Attention mechanism LSTM
AV	Autonomous vehicles
BC	Behavior cloning
B-MPC	Behavior-aware MPC
CAV	Connected autonomous vehicle
CDC	US Center for Disease Control
CL-RRT	Closed-loop-RRT
CTE	Cross-tracking error
CTR	Collision-to-timeout ratio
D-A3C	Delayed A3C
DARPA	Defense Advanced Research Projects Agency
DDPG	Deep deterministic policy gradient
DDQN	Double deep Q-network
DMV	Department of Motor Vehicles
DQN	Deep Q-learning
DRL	Deep RL
FC	Fully connected
FSM	Finite state machines
GAIL	Generative adversarial imitation learning
GNSS	Global navigation satellite system
GPR	Gaussian process regression
HD	High-definition
HJ	Hamilton–Jacobi
H-REIL	Hierarchical reinforcement and imitation learning
IDM	Intelligent driver model
IL	Imitation learning

IP-SAC	Interval prediction and self-attention mechanism
IRL	Inverse reinforcement learning
LiDAR	Light detection and ranging
LKS	Lane-keeping system
LSTM	Long short-term memory
MPC	Model predictive control
MSE	Mean squared error
MSFM	Modified social force model
NHTSA	National Highway Traffic Safety Administration
NMPC	Nonlinear MPC
POMDP	Partially observable Markov decision process
PPO	Proximal policy optimization
QP	Quadratic programming
RGB	Red, green, blue
RL	Reinforcement learning
ROS	Robot Operating System
RRT	Rapidly exploring random tree
SAC	Soft actor-critic
SAE	Society of Automotive Engineers
SHAIL	Safety-aware hierarchical adversarial imitation learning
SM	Sliding mode
SQIL	Soft-Q imitation learning
SUMO	Simulation of urban mobility
TD3	Twin-delayed deep deterministic policy gradient
TTC	Time to collision
UAV	Unmanned aerial vehicle
V2V	Vehicle-to-vehicle
VAE	Variable autoencoder
WHO	World Health Organization

References

1. World Health Organization. *Global Status Report on Road Safety 2018*; Technical Report; World Health Organization: Geneva, Switzerland, 2018.
2. Fagnant, D.J.; Kockelman, K. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transp. Res. Part A Policy Pract.* **2015**, *77*, 167–181. [\[CrossRef\]](#)
3. Singh, S. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*; Technical Report; National Highway Traffic Safety Administration: Washington, DC, USA, 2018.
4. Montgomery, W.D.; Mudge, R.; Groshen, E.L.; Helper, S.; MacDuffie, J.P.; Carson, C. America's Workforce and the Self-Driving Future: Realizing Productivity Gains and Spurring Economic Growth. 2018. Available online: <https://trid.trb.org/view/1516782> (accessed on 8 November 2022).
5. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*; The MIT Press: Cambridge, MA, USA, 2005.
6. Koopman, P.; Wagner, M. Autonomous Vehicle Safety: An Interdisciplinary Challenge. *IEEE Intell. Transp. Syst. Mag.* **2017**, *9*, 90–96. [\[CrossRef\]](#)
7. California Department of Motor Vehicles. *Autonomous Vehicles Testing with a Driver*; California Department of Motor Vehicles: Sacramento, CA, USA, 2022.
8. California Department of Motor Vehicles. *Autonomous Vehicle Collision Reports*; California Department of Motor Vehicles: Sacramento, CA, USA, 2022.
9. Leilabadi, S.H.; Schmidt, S. In-depth Analysis of Autonomous Vehicle Collisions in California. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 889–893.
10. McCarthy, R.L. Autonomous Vehicle Accident Data Analysis: California OL 316 Reports: 2015–2020. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part B Mech. Eng.* **2022**, *8*, 034502. [\[CrossRef\]](#)
11. Society of Automotive Engineers. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. *Soc. Automot. Eng.* **2021**, 4970, 1–5.
12. Kalra, N.; Paddock, S.M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.* **2016**, *94*, 182–193. [\[CrossRef\]](#)
13. National Center for Statistics and Analysis. *Traffic Safety Facts 2020: A Compilation of Motor Vehicle Crash Data*; Technical Report; National Highway Traffic Safety Administration: Washington, DC, USA, 2022.

14. Kuutti, S.; Bowden, R.; Jin, Y.; Barber, P.; Fallah, S. A survey of deep learning applications to autonomous vehicle control. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 712–733. [\[CrossRef\]](#)
15. Aradi, S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 740–759. [\[CrossRef\]](#)
16. Ye, F.; Zhang, S.; Wang, P.; Chan, C.Y. A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 1073–1080.
17. Liu, W.; Hua, M.; Deng, Z.; Huang, Y.; Hu, C.; Song, S.; Gao, L.; Liu, C.; Xiong, L.; Xia, X. A Systematic Survey of Control Techniques and Applications: From Autonomous Vehicles to Connected and Automated Vehicles. *arXiv* **2023**, arXiv:2303.05665.
18. Trenberth, K.E.; Zhang, Y. How often does it really rain? *Bull. Am. Meteorol. Soc.* **2018**, *99*, 289–298. [\[CrossRef\]](#)
19. Stevens, S.E.; Schreck, C.J.; Saha, S.; Bell, J.E.; Kunkel, K.E. Precipitation and fatal motor vehicle crashes: Continental analysis with high-resolution radar data. *Bull. Am. Meteorol. Soc.* **2019**, *100*, 1453–1461. [\[CrossRef\]](#)
20. Arthi, V.; Murugeswari, R.; Nagaraj, P. Object Detection of Autonomous Vehicles under Adverse Weather Conditions. In Proceedings of the 2022 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAIAI), Chennai, India, 8–10 December 2022; Volume 1, pp. 1–8.
21. Azam, S.; Munir, F.; Jeon, M. Channel boosting feature ensemble for radar-based object detection. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 762–769.
22. Li, Y.J.; Park, J.; O'Toole, M.; Kitani, K. Modality-agnostic learning for radar-lidar fusion in vehicle detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 918–927.
23. Zhang, Y.; Carballo, A.; Yang, H.; Takeda, K. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS J. Photogramm. Remote Sens.* **2023**, *196*, 146–177. [\[CrossRef\]](#)
24. Zang, S.; Ding, M.; Smith, D.; Tyler, P.; Rakotoarivelo, T.; Kaafar, M.A. The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car. *IEEE Veh. Technol. Mag.* **2019**, *14*, 103–111. [\[CrossRef\]](#)
25. Vargas, J.; Alsweiss, S.; Toker, O.; Razdan, R.; Santos, J. An overview of autonomous vehicles sensors and their vulnerability to weather conditions. *Sensors* **2021**, *21*, 5397. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Yoneda, K.; Suganuma, N.; Yanase, R.; Aldibaja, M. Automated driving recognition technologies for adverse weather conditions. *IATSS Res.* **2019**, *43*, 253–262. [\[CrossRef\]](#)
27. Kordani, A.A.; Rahmani, O.; Nasiri, A.S.A.; Boroomandrad, S.M. Effect of adverse weather conditions on vehicle braking distance of highways. *Civ. Eng. J.* **2018**, *4*, 46–57. [\[CrossRef\]](#)
28. Shibata, Y.; Arai, Y.; Saito, Y.; Hakura, J. Development and evaluation of road state information platform based on various environmental sensors in snow countries. In Proceedings of the Advances in Internet, Data and Web Technologies: The 8th International Conference on Emerging Internet, Data and Web Technologies (EIDWT-2020), Kitakyushu, Japan, 24–26 February 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 268–276.
29. Šabanović, E.; Žuraulis, V.; Prentkovskis, O.; Skrickij, V. Identification of road-surface type using deep neural networks for friction coefficient estimation. *Sensors* **2020**, *20*, 612. [\[CrossRef\]](#)
30. Panhuber, C.; Liu, B.; Scheickl, O.; Wies, R.; Isert, C. Recognition of road surface condition through an on-vehicle camera using multiple classifiers. In *Proceedings of the SAE-China Congress 2015: Selected Papers*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 267–279.
31. Abdić, I.; Fridman, L.; Brown, D.E.; Angell, W.; Reimer, B.; Marchi, E.; Schuller, B. Detecting road surface wetness from audio: A deep learning approach. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 3458–3463.
32. Taş, Ö.Ş.; Stiller, C. Limited visibility and uncertainty aware motion planning for automated driving. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1171–1178.
33. Taş, Ö.Ş.; Salscheider, N.O.; Poggenhans, F.; Wirges, S.; Bandera, C.; Zofka, M.R.; Strauss, T.; Zöllner, J.M.; Stiller, C. Making bertha cooperate—team annieway's entry to the 2016 grand cooperative driving challenge. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 1262–1276. [\[CrossRef\]](#)
34. Peng, T.; Su, L.L.; Guan, Z.W.; Hou, H.J.; Li, J.K.; Liu, X.L.; Tong, Y.K. Lane-change model and tracking control for autonomous vehicles on curved highway sections in rainy weather. *J. Adv. Transp.* **2020**, *2020*, 8838878. [\[CrossRef\]](#)
35. Joa, E.; Hyun, Y.; Park, K.; Kim, J.; Yi, K. Model Predictive Control based Stability Control of Autonomous Vehicles on Low Friction Road. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 419–424.
36. He, S.; Xu, X.; Xie, J.; Wang, F.; Liu, Z. Adaptive control of dual-motor autonomous steering system for intelligent vehicles via Bi-LSTM and fuzzy methods. *Control Eng. Pract.* **2023**, *130*, 105362. [\[CrossRef\]](#)
37. Hamid, U.Z.A.; Kyyhkynen, A.; Peralta-Cabezas, J.L.; Saarinen, J.; Santamala, H. All-Weather Autonomous Vehicle: Performance Analysis of an Automated Heavy Quadricycle in Non-Snow and Snowstorm Conditions using Single Map. In *The IAVSD International Symposium on Dynamics of Vehicles on Roads and Tracks*; Springer: Cham, Switzerland, 2019; pp. 1100–1106.

38. Samokhin, S.; Mehndiratta, M.; Hamid, U.Z.A.; Saarinen, J. Adaptive Fuzzy Tuning Framework for Autonomous Vehicles: An Experimental Case Study. In Proceedings of the 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring), Helsinki, Finland, 25–28 April 2021; pp. 1–5.
39. Hoffmann, G.M.; Tomlin, C.J.; Montemerlo, M.; Thrun, S. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 2296–2301.
40. Tominaga, K.; Takeuchi, Y.; Tomoki, U.; Kameoka, S.; Kitano, H.; Quirynen, R.; Berntorp, K.; Cairano, S. *GNSS Based Lane Keeping Assist System via Model Predictive Control*; Technical Report, SAE Technical Paper; SAE: Warrendale, PA, USA, 2019.
41. Takeuchi, Y.; Hideyuki, T.; Kazuo, H.; Tomoki, U. *Development of Autonomous Driving System Using GNSS and High Definition Map*; Technical Report, SAE Technical Paper; SAE: Warrendale, PA, USA, 2018.
42. National Highway Traffic Safety Administration. *Traffic Safety Facts 2019: A Compilation of Motor Vehicle Crash Data*; Technical Report; NHTS Administration: Washington, DC, USA, 2019.
43. Al-Dabbagh, M.S.M.; Al-Sherbaz, A.; Turner, S. The impact of road intersection topology on traffic congestion in urban cities. In Proceedings of the SAI Intelligent Systems Conference, London, UK, 6–7 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1196–1207.
44. Lian, F.; Chen, B.; Zhang, K.; Miao, L.; Wu, J.; Luan, S. Adaptive traffic signal control algorithms based on probe vehicle data. *J. Intell. Transp. Syst.* **2021**, *25*, 41–57. [\[CrossRef\]](#)
45. Wu, T.; Zhou, P.; Liu, K.; Yuan, Y.; Wang, X.; Huang, H.; Wu, D.O. Multi-Agent Deep Reinforcement Learning for Urban Traffic Light Control in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8243–8256. [\[CrossRef\]](#)
46. Sahu, S.P.; Dewangan, D.K.; Agrawal, A.; Sai Priyanka, T. Traffic Light Cycle Control using Deep Reinforcement Technique. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 697–702. [\[CrossRef\]](#)
47. Kamal, M.A.S.; Imura, J.i.; Hayakawa, T.; Ohata, A.; Aihara, K. A Vehicle-Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1136–1147. [\[CrossRef\]](#)
48. Li, S.; Shu, K.; Chen, C.; Cao, D. Planning and decision-making for connected autonomous vehicles at road intersections: A review. *Chin. J. Mech. Eng.* **2021**, *34*, 1–18. [\[CrossRef\]](#)
49. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
50. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Computer Science Department, Iowa State University: Ames, IA, USA, October 1998.
51. Karaman, S.; Frazzoli, E. Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.* **2011**, *30*, 846–894. [\[CrossRef\]](#)
52. Kuwata, Y.; Teo, J.; Karaman, S.; Fiore, G.; Frazzoli, E.; How, J. Motion planning in complex environments using closed-loop prediction. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008; p. 7166.
53. Ma, L.; Xue, J.; Kawabata, K.; Zhu, J.; Ma, C.; Zheng, N. Efficient sampling-based motion planning for on-road autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 1961–1976. [\[CrossRef\]](#)
54. Yoon, S.; Lee, D.; Jung, J.; Shim, D.H. Spline-based RRT* using piecewise continuous collision-checking algorithm for Car-like vehicles. *J. Intell. Robot. Syst.* **2018**, *90*, 537–549. [\[CrossRef\]](#)
55. Yang, K.; Sukkarieh, S. An analytical continuous-curvature path-smoothing algorithm. *IEEE Trans. Robot.* **2010**, *26*, 561–568. [\[CrossRef\]](#)
56. Wu, X.; Nayak, A.; Eskandarian, A. Motion planning of autonomous vehicles under dynamic traffic environment in intersections using probabilistic rapidly exploring random tree. *SAE Int. J. Connect. Autom. Veh.* **2021**, *4*, 383–399. [\[CrossRef\]](#)
57. Wang, Y.; Liu, Z.; Zuo, Z.; Li, Z.; Wang, L.; Luo, X. Trajectory Planning and Safety Assessment of Autonomous Vehicles Based on Motion Prediction and Model Predictive Control. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8546–8556. [\[CrossRef\]](#)
58. Hult, R.; Zanon, M.; Gros, S.; Falcone, P. Optimal Coordination of Automated Vehicles at Intersections: Theory and Experiments. *IEEE Trans. Control. Syst. Technol.* **2019**, *27*, 2510–2525. [\[CrossRef\]](#)
59. Liu, C.; Lee, S.; Varnhagen, S.; Tseng, H.E. Path planning for autonomous vehicles using model predictive control. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 174–179. [\[CrossRef\]](#)
60. Moreau, J.; Melchior, P.; Victor, S.; Moze, M.; Aioun, F.; Guillemard, F. Reactive Path Planning for Autonomous Vehicle Using Bézier Curve Optimization. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1048–1053. [\[CrossRef\]](#)
61. Kamran, D.; Lopez, C.F.; Lauer, M.; Stiller, C. Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1205–1212.
62. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.

63. Qiao, Z.; Muelling, K.; Dolan, J.M.; Palanisamy, P.; Mudalige, P. Automatically generated curriculum based reinforcement learning for autonomous vehicles in urban environment. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1233–1238.
64. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 41–48.
65. Sadigh, D.; Sastry, S.; Seshia, S.A.; Dragan, A.D. Planning for autonomous cars that leverage effects on human actions. In Proceedings of the Robotics: Science and Systems, Ann Arbor, MI, USA, 18–22 June 2016; Volume 2, pp. 1–9.
66. Tram, T.; Batkovic, I.; Ali, M.; Sjöberg, J. Learning when to drive in intersections by combining reinforcement learning and model predictive control. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 3263–3268.
67. Tram, T.; Jansson, A.; Grönberg, R.; Ali, M.; Sjöberg, J. Learning negotiating behavior between cars in intersections using deep q-learning. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3169–3174.
68. Koschi, M.; Pek, C.; Beikirch, M.; Althoff, M. Set-based prediction of pedestrians in urban environments considering formalized traffic rules. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2704–2711.
69. Camara, F.; Bellotto, N.; Cosar, S.; Nathanael, D.; Althoff, M.; Wu, J.; Ruenz, J.; Dietrich, A.; Fox, C.W. Pedestrian Models for Autonomous Driving Part I: Low-Level Models, From Sensing to Tracking. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 6131–6151. [\[CrossRef\]](#)
70. Camara, F.; Bellotto, N.; Cosar, S.; Weber, F.; Nathanael, D.; Althoff, M.; Wu, J.; Ruenz, J.; Dietrich, A.; Markkula, G.; et al. Pedestrian Models for Autonomous Driving Part II: High-Level Models of Human Behavior. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5453–5472. [\[CrossRef\]](#)
71. Centers for Disease Control and Prevention. WISQARS: Web-Based Injury Statistics Query and Reporting System. 2021. Available online: <https://www.cdc.gov/injury/wisqars/index.html> (accessed on 21 June 2022).
72. Stewart, T. *Overview of Motor Vehicle Crashes in 2020*; Technical Report; National Highway Traffic Safety Administration: Washington, DC, USA, 2022.
73. Chao, E. *Autonomous Driving: Mapping and Behavior Planning for Crosswalks*. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2019.
74. Kapania, N.R.; Govindarajan, V.; Borrelli, F.; Gerdes, J.C. A hybrid control design for autonomous vehicles at uncontrolled crosswalks. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1604–1611.
75. Thornton, S.M. *Autonomous Vehicle Motion Planning with Ethical Considerations*. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2018.
76. Kochenderfer, M.J. *Decision Making under Uncertainty: Theory and Application*; MIT Press: Cambridge, MA, USA, 2015.
77. Zhu, H.; Iryo-Asano, M.; Alhajjaseen, W.K.; Nakamura, H.; Dias, C. Interactions between autonomous vehicles and pedestrians at unsignalized mid-block crosswalks considering occlusions by opposing vehicles. *Accid. Anal. Prev.* **2021**, *163*, 106468. [\[CrossRef\]](#)
78. Zhou, B.; Zhang, C.; Peng, H.; Lv, C.; Qiu, T.Z. *Research on Pedestrian Crossing Behaviors at Unsignalized Multi-Lane Mid-Block Crosswalk: A Case Study in China*; Technical Report; Transportation Research Board: Washington, DC, USA, 2016.
79. Zhang, C.; Zhou, B.; Qiu, T.Z.; Liu, S. Pedestrian crossing behaviors at uncontrolled multi-lane mid-block crosswalks in developing world. *J. Saf. Res.* **2018**, *64*, 145–154. [\[CrossRef\]](#)
80. Lopez, P.A.; Behrlich, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wießner, E. Microscopic traffic simulation using sumo. In Proceedings of the 2018 21st international conference on intelligent transportation systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582.
81. Chen, H.; Zhang, X. Path Planning for Intelligent Vehicle Collision Avoidance of Dynamic Pedestrian Using Att-LSTM, MSFM, and MPC at Unsignalized Crosswalk. *IEEE Trans. Ind. Electron.* **2021**, *69*, 4285–4295. [\[CrossRef\]](#)
82. Li, Y.; Lu, X.Y.; Wang, J.; Li, K. Pedestrian trajectory prediction combining probabilistic reasoning and sequence learning. *IEEE Trans. Intell. Veh.* **2020**, *5*, 461–474. [\[CrossRef\]](#)
83. Brunoud, A.; Lombard, A.; Zhang, M.; Abbas-Turki, A.; Gaud, N.; Koukam, A. Comparison of Deep Reinforcement Learning Methods for Safe and Efficient Autonomous Vehicles at Pedestrian Crossings. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 2556–2562.
84. Jayaraman, S.K.; Robert, L.P.; Yang, X.J.; Pradhan, A.K.; Tilbury, D.M. Efficient behavior-aware control of automated vehicles at crosswalks using minimal information pedestrian prediction model. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 4362–4368.
85. Jayaraman, S.K.; Tilbury, D.M.; Yang, X.J.; Pradhan, A.K.; Robert, L.P. Analysis and prediction of pedestrian crosswalk behavior during automated vehicle interactions. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 6426–6432.
86. Chen, Y.; Li, S.; Tang, X.; Yang, K.; Cao, D.; Lin, X. Interaction-Aware Decision Making for Autonomous Vehicles. *IEEE Trans. Transp. Electr.* **2023**. [\[CrossRef\]](#)
87. Steyn, H.J.; Ashleigh, G.; Lee A., R. *A Review of Fatal and Severe Injury Crashes at Roundabouts*; Technical Report; United States Department of Transportation: Washington, DC, USA, 2020.

88. Daniels, S.; Brijs, T.; Nuyts, E.; Wets, G. Extended prediction models for crashes at roundabouts. *Saf. Sci.* **2011**, *49*, 198–207. [CrossRef]
89. Insurance Institute for Highway Safety. *Roundabouts*; Insurance Institute for Highway Safety: Ruckersville, VA, USA, 2022.
90. Hang, P.; Huang, C.; Hu, Z.; Xing, Y.; Lv, C. Decision Making of Connected Automated Vehicles at an Unsignalized Roundabout Considering Personalized Driving Behaviours. *IEEE Trans. Veh. Technol.* **2021**, *70*, 4051–4064. [CrossRef]
91. Lodinger, N.R.; DeLucia, P.R. Does automated driving affect time-to-collision judgments? *Transp. Res. Part F Traffic Psychol. Behav.* **2019**, *64*, 25–37. [CrossRef]
92. Chen, J.; Yuan, B.; Tomizuka, M. Model-free Deep Reinforcement Learning for Urban Autonomous Driving. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 2765–2771. [CrossRef]
93. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
94. Huang, Z.; Wu, J.; Lv, C. Efficient Deep Reinforcement Learning With Imitative Expert Priors for Autonomous Driving. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–13. [CrossRef] [PubMed]
95. Zhou, M.; Luo, J.; Villella, J.; Yang, Y.; Rusu, D.; Miao, J.; Zhang, W.; Alban, M.; Fadarar, I.; Chen, Z.; et al. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv* **2020**, arXiv:2010.09776.
96. Jamgochian, A.; Buehrle, E.; Fischer, J.; Kochenderfer, M.J. SHAIL: Safety-Aware Hierarchical Adversarial Imitation Learning for Autonomous Driving in Urban Environments. *arXiv* **2022**, arXiv:2204.01922.
97. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540.
98. Zhan, W.; Sun, L.; Wang, D.; Shi, H.; Clausse, A.; Naumann, M.; Kummerle, J.; Konigshof, H.; Stiller, C.; de La Fortelle, A.; et al. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. *arXiv* **2019**, arXiv:1910.03088.
99. Capasso, A.P.; Bacchiani, G.; Molinari, D. Intelligent roundabout insertion using deep reinforcement learning. *arXiv* **2020**, arXiv:2001.00786.
100. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, Singapore, 20–23 February 2016; pp. 1928–1937.
101. Bacchiani, G.; Molinari, D.; Patander, M. Microscopic traffic simulation by cooperative multi-agent deep reinforcement learning. *arXiv* **2019**, arXiv:1903.01365.
102. Liu, H.; Huang, Z.; Wu, J.; Lv, C. Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. In Proceedings of the 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 4–9 June 2022; pp. 921–928.
103. Reddy, S.; Dragan, A.D.; Levine, S. Sqil: Imitation learning via reinforcement learning with sparse rewards. *arXiv* **2019**, arXiv:1905.11108.
104. Masi, S.; Xu, P.; Bonnifait, P. Roundabout Crossing With Interval Occupancy and Virtual Instances of Road Users. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4212–4224. [CrossRef]
105. Wang, Z.; Liu, X.; Wu, Z. Design of Unsignalized Roundabouts Driving Policy of Autonomous Vehicles Using Deep Reinforcement Learning. *World Electr. Veh. J.* **2023**, *14*, 52. [CrossRef]
106. Leurent, E. An Environment for Autonomous Driving Decision-Making. 2018 Available online: <https://github.com/eleurent/highway-env> (accessed on 21 January 2023).
107. Seong, H.; Jung, C.; Lee, S.; Shim, D.H. Learning to drive at unsignalized intersections using attention-based deep reinforcement learning. In Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 19–22 September 2021; pp. 559–566.
108. Riccardi, M.R.; Augeri, M.G.; Galante, F.; Mauriello, F.; Nicolosi, V.; Montella, A. Safety Index for evaluation of urban roundabouts. *Accid. Anal. Prev.* **2022**, *178*, 106858. [CrossRef] [PubMed]
109. Alighanbari, S.; Azad, N.L. Deep Reinforcement Learning With NMPC Assistance Nash Switching for Urban Autonomous Driving. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2604–2615. [CrossRef]
110. Alighanbari, S.; Azad, N.L. Safe Adaptive Deep Reinforcement Learning for Autonomous Driving in Urban Environments. Additional Filter? How and Where? *IEEE Access* **2021**, *9*, 141347–141359. [CrossRef]
111. Cao, Z.; Biyik, E.; Wang, W.Z.; Raventos, A.; Gaidon, A.; Rosman, G.; Sadigh, D. Reinforcement Learning based Control of Imitative Policies for Near-Accident Driving. In Proceedings of the Robotics: Science and Systems (RSS), Virtual, 12–16 July 2020.
112. Lee, K.; Kum, D. Longitudinal and Lateral Integrated Safe Trajectory Planning of Autonomous Vehicle via Friction Limit. In Proceedings of the 2020 20th International Conference on Control, Automation and Systems (ICCAS), Busan, Republic of Korea, 13–16 October 2020; pp. 1177–1180. [CrossRef]
113. Milliken, W.F.; Milliken, D.L.; Metz, L.D. *Race Car Vehicle Dynamics*; SAE International: Warrendale, PA, USA, 1995; Volume 400.
114. Shang, X.; Eskandarian, A. Emergency Collision Avoidance and Mitigation Using Model Predictive Control and Artificial Potential Function. *IEEE Trans. Intell. Veh.* **2023**, *8*, 3458–3472. [CrossRef]

115. Bansal, S.; Chen, M.; Herbert, S.; Tomlin, C.J. Hamilton-jacobi reachability: A brief overview and recent advances. In Proceedings of the 2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 2242–2253.
116. Caesar, H.; Kabzan, J.; Tan, K.S.; Fong, W.K.; Wolff, E.; Lang, A.; Fletcher, L.; Beijbom, O.; Omari, S. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv* **2021**, arXiv:2106.11810 .
117. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2020**, *37*, 362–386. [[CrossRef](#)]
118. 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](#)]
119. Xu, H.; Gao, Y.; Yu, F.; Darrell, T. End-to-end learning of driving models from large-scale video datasets. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2174–2182.
120. Hecker, S.; Dai, D.; Van Gool, L. End-to-end learning of driving models with surround-view cameras and route planners. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 435–453.
121. Tampuu, A.; Matiisen, T.; Semikin, M.; Fishman, D.; Muhammad, N. A survey of end-to-end driving: Architectures and training methods. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1364–1384. [[CrossRef](#)]
122. Zhang, Y.; Tiño, P.; Leonardis, A.; Tang, K. A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**, *5*, 726–742. [[CrossRef](#)]
123. Chen, J.; Li, S.E.; Tomizuka, M. Interpretable End-to-End Urban Autonomous Driving With Latent Deep Reinforcement Learning. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 5068–5078. [[CrossRef](#)]
124. Wang, H.; Gao, H.; Yuan, S.; Zhao, H.; Wang, K.; Wang, X.; Li, K.; Li, D. Interpretable Decision-Making for Autonomous Vehicles at Highway On-Ramps With Latent Space Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 8707–8719. [[CrossRef](#)]
125. Schmidt, L.M.; Kontes, G.; Plinge, A.; Mutschler, C. Can You Trust Your Autonomous Car? Interpretable and Verifiably Safe Reinforcement Learning. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium (IV), Nagoya, Japan, 11–17 July 2021; pp. 171–178. [[CrossRef](#)]

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.