Autonomous Vehicle Path Planning Using Deep Reinforcement Learning

By Dr. Sarah Hassan

Professor of Computer Science, Zewail City of Science and Technology, Egypt

1. Introduction

Deep reinforcement learning (DRL) has successfully been used to solve various nonlinear and complex problem domains such as those characterized by continuous states or actions. Atari games, robot control, and path planning are working examples of such cases. The quality of the learned controller, however, is heavily dependent on both the accuracy and size of the data used for learning. In addition, the generation of DRL trajectories is extremely slow and their quality and smoothness can vary greatly depending on the discrete nature of the chosen algorithm [1] [2]. Given these challenges, we aim to introduce an advanced adaptation of the existing DRL algorithm for solving the path-planning issue in an autonomous vehicle-targeting environment safety-critical problem domain. This research also targets reduced transit time and smoother and less aggressive behavior.

Reinforcement learning (RL) is a branch of machine learning designed for training agents to exhibit a high degree of competence in a given environment [3]. RL has attracted significant interest in the vehicle control field as a method for undertaking tasks such as path planning and vehicle control. Successful path planning is a crucial element in autonomous navigation, providing a "database of rules" for driverless cars. Path planning must handle the unknown environment, estimated local surroundings, and non-constant external dynamics. Beyond that, it must also account for various dynamic constraints, traversing from one point to another in an environment characterized by dynamically changing obstacles. Unfortunately, the traditional planning techniques for path planning are hard-pressed to solve this issue, so numerous pathplanning algorithms have been developed based on different learning paradigms.

1.1. Background and Motivation

In this paper, we adopt deep reinforcement learning (DRL) as the learning algorithm for the global route planner. DRL is a learning-based method that integrates the deep neural network as the value-function approximator of a conventional reinforcement learning method [4]. It has shown great advantages in solving path planning problems of mobile robots. Several studies have been conducted to verify the effectiveness of the proposed method in dealing with different path planning tasks, including dynamic environments (Bhattacharyya and Khilar 2017; Jiao et al. 2018), crowded scenarios (Xuan et al. 2020), and perceptually challenging conditions (Kang et al. 2020). Although DRL has been successfully used in the aforementioned fields, global and dynamic planning of the autonomous vehicle remains less well-explored, and more importantly, the reinforcement learning in existence increases the complexity with the growth of state space. Different from common optimization methods which require to calculate state space and action space, imitation learning bypasses this difficult task by learning the policy directly from expert demonstrations. But imitation learning-based approaches may provide sub-optimal solutions and face the suffer of overfitting, so we introduce a DRL-based global route planner in the existing framework.

Planners in autonomous vehicles are usually classified into global and local [5]. This paper focuses mainly on the global path planner that plans the vehicle's route in the static map by considering the static environment. The traditional model-based methods for the global planner can guarantee the optimality but may lead to local feasible minimums in challenging environments, resulting in a plan with poor performance. On the other hand, imitation learning can bypass the difficult task of learning the policy directly from expert demonstrations, but may provide sub-optimal solutions and overfitting issues. Inspired by the success of combining local planning and global planning methods, in this work, similarly, we use deep reinforcement learning (DRL) to guide the trained policy directly towards the global planning.

1.2. Research Objectives

[6] The primary objective of this research is to investigate and design an efficient, Autonomous Ground Vehicles (AGVs) path planning system in dynamic and complex environment based on deep reinforcement learning (DRL). Precisely, the following four objectives shall be accomplished. (i) Propose a DRL model to predict the movement directions of potential field. (ii) An enhanced DRL system is developed to address reinforcement learning instability issue

and employ a novel noise injection method to investigate the feasible paths in a stochastic decision-making manner. (iii) A highly efficient replay buffer mechanism is developed to improve the learning memory of DRL models, thereby significantly accelerating DRL learning efficiency and the simulation speed in path planning. (iv) A practical DRL framework is envisioned, which integrates between local area-based deterministic path planning algorithms framework and global area-based autonomic planning framework, to determine continuous global path planning and Dynamic environment obstacle avoidance in a dynamic complex environment posture, such that the serious greedy learning of DRL model and the robustness of practical path planning are fully taken into consideration.[1] To achieve the objective of this work, a dynamic threat assessment has been proposed to take into account the random effect of traffic vehicles and has also a robust driving strategy for Automated Driving Vehicles. For the future work, the analysis of performance parameters under a real-world data environment and sim-to-real testing techniques will be discussed to cover the complete aspects of this project. The future research aims to design a policy that involves deep reinforcement learning and also incorporates automated driving functionalities, such as lane changing, overtaking, lateral movement of the vehicle, etc. These functionalities can be implemented using a deep Q-network (DQN) with reinforcement learning to overcome the identified challenges of path planning and the dynamic threat assessment in autonomous vehicles.

1.3. Scope and Limitations

Therefore, it was proposed to take advantage of a Predictive State Representation partially Observable Markov Decision Process (POMDP) framework that directly takes into account the fact that the state of the autonomous vehicle is influenced not only by the present control action and environment state but also by the sequence of upcoming environmental actions that are inferred from the data, by learning their temporal auto-regression relation. This technique takes into account at once both the observation of the current state and the predicted environment states, and should allow the DRL technique to better plan its future actions and to reduce the likelihood of long-term generated actions by a chaining of short-term thinking ones that will be part of the sequential planning. Combining these two techniques, we proposed two separate parallel architectures named, respectively, FLUID and SPIGHI ideas, online applied on the reference RL agent.

Nevertheless, a case in which environment mapping is not strictly feasible was explored. Many modern sensors, employed for example in autonomous driving tasks, may fail when used under adverse environmental conditions. Starting from the recognition that the presence of near walls leads to complex to map dynamics in the agent/vehicle state-space, which must be instead completely avoided to avoid collisions, it is proposed a first path planning method based on the DRL paradigm that bypasses the above stated problems by directly learning the optimal (avoiding walls) path without making the environment map static assumptions [7]. This mode of operation eases the planning problem, since the decision is taken by considering only the current state of the environment. Such a structure may be perturbed by the addition of very simple and straightforward complementary logic. On the other hand, if the environment is known but a long plan must be made, some errors may occur if DRL is exploited standalone. In this case, it was shown that a proactive inference of environment state actions that should be undertaken to keep the system in an optimal state, during a time step, may help with the problem.

Recent years have witnessed a true revolution of robotic systems. Robots became more and more involved in complex dynamic systems, where it was not enough to program some deterministic reaction to some expected situation. In most of these cases a dynamic approach should be followed to obtain an optimal or near optimal action. This problem can be solved by the adoption of reinforcement learning (RL) principles. RL is a learning paradigm where an agent learns how to interact with a dynamic and possibly partially observable environment [3]. Over the years many algorithms have been developed as solutions to this problem: from value-based algorithms, over policy optimization, until the development of true models for the possible interactions in the environment, which can be considered the actual state of the art in the field. This work proposes a possible solution technique for autonomous vehicle path planning based on the Deep Reinforcement Learning (DRL) paradigm, including some of its most recent and promising developments. Of course, the basic principles and architectures will be briefly discussed. Then, contributions with respect to some of the main issues faced during path planning will be discussed. The worked showed that DRL was able to learn path planning policies that were better than some traditional methods like Dijkstra's algorithm or A* search with heuristics. They applied the DRL again to two more complex setups to plan a path in a modular structure with an unknown number of modules and space to explore and optimize a car trajectory within an environment [8].

2. Literature Review

Path planning is a fundamental component of controlling autonomous vehicles [9]. This ultimately determines a trajectory for the vehicle by avoiding obstacles and ensuring that critical vehicle performance parameters are within prescribed limits. Therefore, path planning is essential to the vehicle's safety and operational efficiency [10]. Multiple path planning strategies have been explored in the literature, particularly for autonomous vehicles; these include Dynamic Programming, A* search Water flow algorithms and Neural Networks based techniques [11]. Nevertheless, these literatures demonstrate challenges associated with path planning such as equal flexibility in dynamic environment; vehicle parameters will saturate at certain time due to the limitation of search based algorithms; neural Network based systems have very lengthy training and testing period. Autonomous vehicle technology provides an opportunity for improving transportation. The goal of the technology is to create a transportation system that is safer, more pleasurable, more convenient, and environmentally friendly than that of traditional vehicles. Autonomous vehicles are capable of identifying the surrounding environment and performing real-time operations using multiple sensors, GPS, and control systems. The data collected from these sensors is processed and then used for recognizing and making situational judgments, making decisions, and executing actions through algorithms and models trained by Machine Learning and Deep Learning methods. Despite the advantages of autonomous vehicle technology, it also encompasses various challenges, one of which is to make deep decisions when selecting appropriate actions.

2.1. Traditional Path Planning Methods

People have well adapted DRL to the trajectory planning and control of autonomous driving vehicles, which has generated a thorough systematic investigationat [4]. Although trajectory predicting and timing plan have covered the carrying environment to a reasonable extent, to satisfy the increasing environmental complexity: Be vulnerable to diversified controllable conditions only within the current traffic trajectory predicting and hard to cover the underlying general rules. Many related applications, such as vehicle controller and driver aiding, also have good mileage from trajectory predicting. Despite the full load of attack for this local single-minute station issue, the autonomous driving still has huge quantitative and qualitative carrying to reckon with.

Trajectory planning and path planning form the core aspect of autonomous driving by ensuring safety while covering the shortest path [12]. Algorithms in trajectory planning can be divided into three types, namely rule-based methods, optimization methods and artificial intelligence based methods. The first type of methods are the rule-based methods, their advantage lies in that they can generate feasible and collision-free path based on given knowledge of traffic rules and regulations. In addition, these optimization based approaches mostly focus on achieving minimum time or energy by looking for an adaptive trajectory among multiple choices. Optimal control and sequential quadratic programming exploit optimization techniques to solve the trajectory planning for autonomous vehicles. Finally, the researchers from artificial intelligence community have devised a variety of machine learning techniques to enable autonomous driving. By taking historical data of vehicle trajectory, trajectory predicting can take full advantage of current results and yield potential capabilities to deal with uncertainty. Recent research on trajectory predicting has made academia aware of the strong ability of deep neural network (DNN) in this field [9]. Besides trajectory predicting, some research have worked out the optimization model of the timing plan, this approach is able to handle complicated planning problems caused by the large size of approaches.

2.2. Reinforcement Learning in Autonomous Vehicles

Applying machine learning techniques to challenging problems involve finding relevant features and domain knowledge to build upon feature representation, since deep learning uses gradient descent to adjust its weights given feedbacks, while reinforcement learning typically optimizes a policy to generate feedbacks. In real world applications, the use of neural networks and DRL makes the acquisition of a huge dataset on a single scenario prohibitive – thus needing off-line data – while new situations may likewise require complete adjustments of the established structure. In addition, the best policies found may not be stable and resilient to different situations, requiring the user to retrain the model from scratch for new scenarios [2].

Path planning for autonomous vehicles is a major research topic in robotics, where reinforcement learning (RL) has been gaining much attention in recent works [9]. Despite being an im-portant tool in several scientific and engineering areas due to the potential of developing optimal con-trol policies, RL has not yet been thoroughly explored in the context of autonomous vehicles [12]. The most widely used path planning and decision making techniques in this area are based on searching algorithms, which face a major issue when solved using deep reinforcement learning (DRL). The results are usually dependent on the initial state of each run and were produced off-line, and in large scale simulations recent works investigated hierarchical DRL algorithms, demonstrating such dominance in performance and ease of design. It is worth noting that UV are similar challenging problems.

3. Fundamentals of Deep Reinforcement Learning

Additionally, the basic algorithm of this study is specifically designed and improvements can be made with the goal of applying this algorithm to the path planning stage in a real-world setting. This is the first study and our main problem is deep reinforcement learning to optimize the parameters in the polynomial curve interpolation process during global path planning for autonomous driving [8].ivityManager to solve this problem. A DRL agent's objective in this state is to produce an optimal stable navigation trajectory either to avoid current accidents or to deduce the maximum information about the RND from surrounding traffic.

[12]The autonomous vehicles have to navigate safely and effectively through unprecedented traffic in contrastive terrains and to handle complex driving situations effectively and efficiently [13]. The underlying problems can be outlined in planning and control planning. The precise path planning ensures that smooth, efficient, and safe routes negotiate the predictable environment. Once paths are calculated, the control algorithms regulate safe car operation. But a continuous lFL algorithm must be designed for path tracking. Planning schemes for autonomous vehicles can be divided into two categories: rule-based and intelligent algorithms. Among rule-based approaches, polynomial curve fitting based on B+ splines is an extensively studied piece of intelligent approaches that evolve with scale and decision-making ability. B+ system of splines is a piecewise network of polynomials with minimal piecework with uniform integral coefficients and edge limits. B+ continuities arise in both sides with additional parameters called nodes. It helps users solve numerable differentiable points by changing only a few knots, allowing the piecewise carrier to remain unchanged.

3.1. Reinforcement Learning Basics

This survey aims to provide a comprehensive review of existing DRL based unmanned vehicle path planning, which involves from very basic path planning to several scenarios, such as pedestrian avoidance, intersection dynamic, multi-level and so on. Indirectly restricting outputs through researching preprocessing input data also make the autonomous learning agent can be kept away from the hazardous way without any policy restrictions which simultaneously might be limited exploration ability. Learning from the same or similar agent along with the environment may provide the robot with various performances. Reinforcement learning (RL) methods also shows that it supports the environment with limited prior knowledge like random and nonlinear sensor noise. Artetxe [7] stated that traditional path planning algorithms do not have the ability to plan in an unknown environment, so learning and learning-based methods were combined with RL to get good performance. Various works have combined RL with different methods to achieve different tasks such As stated that the homotopy classes structure of the free configuration space governs the way to construct a particular path from the other. It is expected significant modelling improvements by a graph-based RMPT technique as it is planning on the manifold of free configuration spaces, and it is expected the near future merges or unifying these two schemes. We believe that by resolving the issue of safe and smart planning within the passive environment, environment-friendly robot navigation can finally be achieved. In this sense, it is also crucial to compile information about planning and integration with the other planning methods with existing interface as follows Jaiswal et al. [12].

Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment to maximize a notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. The reinforcement signal of previous experiments was mostly prepared by researchers or given to agents completely randomly, otherwise generated by a simple algorithm. Recently, the reinforcement signals are set up based on deep reinforcement learning (DRL) techniques on the premise of deep learning methods and the hybrid method combining with the rule-based and the model-based method which is feasible for problems prone to the curse of dimensionality in a complex environment. DRL can achieve RL for actions on the state space, domain adaptation, and Robustness to states typo et al. [14].

3.2. Deep Learning Basics

The universal function approximation properties of deep architectures allow them to learn representations of the input space that map decisively to output targets. Representation layers are the key to learning abstractions, where input data gets transformed in terms of progressively smaller chunks of information at each layer intra-network. The architecture has a certain number of input and output nodes (often much larger than the number of input and output nodes), with zero or more hidden interneurons or layers in between, and interconnections associated with weights. In such a model, the learning method consists of adjusting the weights of the interconnections between nodes so as to produce desired output from inputs [10].

This section starts with an introduction to some deep learning basics to provide a foundation to understand reinforcement learning, as the focus of this review is on deep reinforcement learning methods. The deep learning revolution emerged in the field of machine learning. Broadly speaking, deep learning focuses on learning multiple levels of abstraction from data with the help of artificial neural networks [14]. In control problems, convolutional neural networks (CNNs) are typically used to feed visual inputs into a deep network to learn different level representations. In contrast, traditional methods require a human to hand-craft these representations. Thus, deep networks operate in a given framework of features and learn a hierarchy of abstraction during training [15].

4. Deep Reinforcement Learning Architectures

Path planning within autonomous driving systems is a critical function responsible for establishing an efficient and safe trajectory for the vehicle. The objective of searching for the optimal path becomes even more complex for self-learning agents who adapt their decision policies based on interactions with the environment. In this survey, a comprehensive review of the DRL-based methods applied to driving applications has been presented under various categories of road architectures, vehicle representations, planning levels, reward shaping, control paradigms, safety constraints, training techniques, and emulation environments [12]. The reviewed papers on DRL-based path planning and control within the autonomous vehicles demonstrate the applicability of proposed approaches within different contexts, such as heterogeneous multi-agent systems, multi-goal multi-agent scenarios, urban environments, as well as open-world environments.

Recent advances in deep reinforcement learning (DRL) have shown promising improvements in the functional safety domain for driving scenarios where unexpected events can be encountered, and/or control tasks and environmental properties are complex and can change significantly over time. DRL methods can overcome the need for precise, perfect domain knowledge and bypass planning steps in the sense-plan-act process. Methods from the DRL community applied in the decision-making part of an autonomous vehicle (AV) have shown promising results. Here, we survey DRL methods applied to planning and control tasks within autonomous cars.

4.1. Deep Q-Networks (DQN)

For instance, a local navigation system was developed based on the Deep Q-Network (DQN) and its extension Double Deep Q-Network (DDQN) to apply to Unmanned (Lunar) Vehicles (UVs). Different from classical algorithms, the proposed method proposed in this paper is suitable for the UVs (IRVs) in order to adapt to the obstacle environment [16]. In addition, models such as whether double-net, dueling network, frameworks, model-based DQN, and a combination of the Monte Carlo tree search were presented. In the existing review researches, most of the literatures are about the research overview, without strengths of the comparison between different models of DQN in autonomous driving systems.

[15] Autonomous vehicles (AVs) have drawn increasing attention since they aim to reduce traffic accidents, enhance traffic flow, push forward sustainable transport and so forth. However, the driving ability of humans is still irreplaceable. Traditional algorithms, such as hand-crafting route rules, find it difficult to meet the requirements of fast-driving scenarios in terms of how to complete the fastest and least cost tasks. To cope with traffic density and variability environment, model-based approaches are restricted to static environment localization and lead to error accumulation. Reinforcement learning algorithms based on deep neural networks have the advantages of mapping sensory inputs to the action, extracting essential features of the environment, and generalizing to new, unseen states. In this way, both the model and policy of the agent can be learned directly without knowing the dynamics of the environment. Deep Q-Networks (DQNs) and its extensions, such as Duelling DQN, Double DQN (DDQN), Multi-step DDQN, Continuous-action DQN, etc., have been applied to develop autonomous driving systems [17].

4.2. Policy Gradient Methods

After achieving reasonable navigation accuracy under certain conditions, such methods will encounter overfitting with environment, insufficient generalization ability with environmental changes, and lack of real adaptability when used in practice, and it is difficult to achieve good results when in visual environment with the task of navigating as learning objectives. Some methods include adding noise to observations or rewards, improving the task setting or modulating the intrinsic or extrinsic goals so that the agent can explore the environment and learn reward function or increase learning diversity via the probability distribution of position.个目的提供了一个新颖的使用随机网络辅助 (Random Network Distillation, RND) 来增强基准的探索能力。 An assessment has been proposed which introduces a universal task related random target generator which can be structured for easier application in AGVs. The suggested model based IDE method introduces the scaling parameter β and universal task and explorer acquisition functions which enhances the exploration ability of AGV. Furthermore, the performance of the suggested method is compared to the state-of-the-art theories of the RL framework and the development in the visibility of A*. This comparison was operated using an image based in-sensed AGV restraint with several static and dynamic surroundings.

Using the most essential technique that is Policy Gradient Methods like Proximal Policy Optimization (PPO) in AGV [18] can achieve major classifications. applied Random Network Distillation to PPO based on traditional sensor configurations for AGVs to increased exploration bonuses to the agent, which was also a variety of methods for enhancing exploration behavior. Liang Ze's method utilized a better scratch pad and image observation for AGV navigation system, but the scratch has more visual noise and redundancy for path planning task of AGV in a controlled environment [19]. Recent research has also begun to explore other policy gradient methods for AGV path planning. With the continuous improvement of machine learning methods and deep reinforcement learning ability, the autonomous navigation of mobile robots has been widely studied, especially some of the methods based on imitation learning, reinforcement learning and end-to-end training. Traditional methods often require hand-designed models, and environmental perception, path planning and vehicle control are usually combined to decompose a large enough problem into small ones.

5. Simulation Environments for Autonomous Vehicles

[10] Different from tradition methods, Deep Reinforcement Learning (DRL) searches parameters of a Deep Neural Network directly from trial and error. To avoid large computational cost, most previous researches simplify the DRL to tabular method or reduce the model scale. Besides, a three-stage process with partial expert demonstration is often used to ensure the safety of driving. In this paper, a new simulation framework is developed for autonomous vehicles using DRL for path planning with Focus Stacking enhancement. First, we primarily build a simulator in which focus-stacking technique is used to improve the visual perception of autonomous vehicles. In the simulator, the upper view images, center view images and below view images at the front of the vehicle are generated, and then focus stacking is performed for each of the three images. Next, we design a new senor fusion model by which 44D hybrid state is obtained. We manage to accomplish deployed and deployed environmental perception better. The improvement of the environmental perception makes the arriving of the passing point more stable and more certain, and we also achieved the virtual grasping of stop lines and pedestrians for the first time. Finally, we set a new reward mechanism without using the important regional detection algorithm, but by which the cumulative reward turns more stable and the training time is greatly shortened.[20] Simulating driving with DRL has always been a challenging task. The current basic problem is how to incorporate the road environment model of the Geo-coordination system, the compromise between the degree of freedom of the actor network, how to design an appropriate reward function and balance the impact of their stability with the importance of various constraints. The object of this work is to enrich the types of location awareness modeling from the most populated one with vector jointly modulated by Euclidean coordinates and speed by adding polar and geometric network coordinate systems of the DRL agent. We specialize in several path planning tasks such as saving fuels by tracking the reference path, and holding on the given lane and Velocity Profile Speed Planning. To make our environment a temporal variant, we do a simple model of pursuing target vehicles that are in the viewer with intelligent behaviors using Multi-Agent Systems (MAT) and intelligent vehicles yielding the autoregressive vector observation for the actor-critic model. Additionally, as the follow distance is important we remind the critic to provide information about collision checks by following procedures: intersection, minimum and maximum distance, stopping distance, and trajectory checking.

5.1. OpenAI Gym

In this work, the classical artificial intelligence term task is solved, which is path planning, in the context of reinforcement learning using an off-the-shelf pseudophysics based software environment. In the first Part of this chapter, it is introduced how OpenAI Gym can be used as the ground primordial environment for e.g. benchmarking methods and algorithms [21]. All the characteristics of the Gym are reviewed and examples are shown. In the second Part of this chapter, a training session is revealed from collecting a data-set through training a machine learning model Finally, the performances of the model on the test environment are inspected. A case study, using one of the Gym environments, is put forward through a typical machine learning benchmark task.

Reinforcement learning is a type of machine learning that is concerned with how software agents ought to take actions within an environment in order to maximize some notion of cumulative reward. Within this setting, deep reinforcement learning methods have been developed and put to practical use in various fields of robotics and artificial intelligence. One recent example is training a soft finger robot hand to grasp a bubble of water inside a world of rigid objects [15]. General-purpose software frameworks for reinforcement learning are also widely used, with OpenAI Gym being a famous and extensively used example. Its noteworthy attributes include a wide range of available environments to train and evaluate reinforcement learning algorithms, ability to easily interface with various deep learning frameworks, and integrated benchmarks and leaderboards.

5.2. CARLA Simulator

The CARLA system tries to provide different features to simulate real-world environments. It encompasses a graphical engine allowingfor high-fidelity visual appearance of the environments, open and scalable physics and dynamics models, using between street andoff-road conditions, and a cross-platform Nature implementation and exhaustive sensors for capturing the whole signals characters in our scene. It offers several sensors such as RGBCAMERA sensory system, the LiDAR sensor, theDepth Camerasensor, theCollision Sensor, and theSemantic Segmentation vehiclesensors making it possible for reinforcement learning. With these sensors it is possible to choosevehicle states such as Depth Images, RGB camera control, LiDAR, and RGB sensor and LiDAR Depth Camera sensor which are interesting problems and domains of research. It was chosen among several other simulators such as AirSim, TORCS, SUMO, or Vissim by [4].

The training environment used is an open source project developed by IntelLabs that simulates urban activities such as vehicle control,lights and camera sensors, allowing access to two different maps. The simulator allows manual and automatic control of a vehicle. In manualmode, the vehicle can be controlled directly using the four arrowkeys on the keyboard. In automatic control mode, a user-defined platform with driving policies defined using Python API is available. Therefore, it integrates experiences through various sensors of autonomous vehicles. We perform data preprocessing to feed the DRL model by expanding the number of sensorsplotted path points, decrease path deviation and the state space. We use three front-facing RGB cameras that take images every 0.1 seconds. We stack 10 camera observations for temporalconvolution. Thedeep Q network is trained and tested on all scenarios of the [22]. CARLA simulator. CARLA is one of the most widely used open-source driving interactive simulators.

6. Data Collection and Preprocessing

DRL is heavily reliant on the quality of the data used [12]. The quality of a given dataset can be highly variable depending on how it is collected, since driving behavior can vary significantly based on the choice of ego state representation and the scarcity of certain scenarios. To address data scarcity and distribution shift, in this paper we propose a high-level timeline of the training process in which each phase is tailored towards addressing different shortcomings. In the data collection phase, we show that we are able to amass large and diverse data by using a relatively easy to train imitation learning model as a way of generating pseudo-demonstrations. The policy of the imitation learner is used to generate data from which the true DQL agent could benefit. While effective with the introduction of the imitation learner, this data collection procedure does not fully mitigate potential distribution shift that might originate from the use of on-policy RL methods. However, we show that the use of curiosity bonuses produces a follow-up agent that is more adept at data collection and has stronger general- ization to diverse and previously unobserved scenarios. More concretely, in the pretraining phase, we focus on distilling the knowledge of the expert agent in order to help the DRL agent learn faster. By solving the task using only the expert demonstrations, it was possible to split the learning from what was essentially a vanilla DQL training process into two easy-to-optimize phases [23].

Deep reinforcement learning (DRL) is a viable approach for generating autonomous driving behavior from data and provides the potential to scale beyond imitation learning [24]. Our work focuses on a navigation and path planning expert for navigating between two way-points in a 90° intersection environment, which would be extremely challenging to learn directly from demonstration data.

6.1. Sensor Data Fusion

Note that the traffic state knowledge is distributed among each vehicle. It is thus necessary to define the necessary traffic states which have to be communicated. However, the traffic state information is unknown in the literature, and there is lacking a common network protocol of traffic state information. The situation deteriorates when all vehicle transit through an urban traffic controlled scenario, thus dealing with equilibrium intersections from the perspectives of all vehicles. Therefore, the researched cluster of CAVs intend to solve, how dynamic and strategic active sensors equipped, interconnected, automated vehicles could share and spread traffic knowledge, through a proper transmission technique, during the development of a refined environment perception module in reaching a reliable and time effective information exchange method and developing a class of advanced, inter-vehicle communication based traffic optimization algorithm. For these motivations, in this paper the authors propose an appropriate sensor traffic knowledge acquisition process, and try to reduce the number of transmitted elements to improve the quality of dedicated environment perception module in both static and dynamic road network geometrical and reflective state.

[ref: f24e88fe-2d2b-4fd5-a293-47531af608b1, 12020b22-78f0-4a33-9733-51ad0ae70c97] In this research, we define stress events, human reaction time, automatically planned vehicle path, destination position, and the planned route. Stress events are events during driving where the vehicle becomes highly unstable or crashes in an instant. Such an event can normally be caused by older people who can react slowly and by entering a narrow passage. In this research, older people can use the autonomous vehicle. The word of autonomous vehicle is used a lot to indicate that vehicle included of the reaction that human have only learned. When the stress events occur, the vehicle should immediately change its driving scenario to a stable state. As time goes by, the vehicle should change the driving method to a stable state considering the current position and speed. The time required to respond to the human eye from the system occurs when a stress event occurs. This may vary for everyone. We assume that the human driver's response time to stress events is 4 s. If there are people who have a slower response time, the system has more time to choose the longest path. No matter how fast the vehicle recognizes and evades minimax t[25] Many algorithms have been proposed for path planning in dynamic and static environments. In literature, two strategies have been followed to design path planning algorithms for autonomous vehicles. The first strategy, named Optimal Control Strategies, solves an optimization problem to find the best path, which minimizes a cost function that is consistent with the provided vehicle. Scientific research is very active in path prediction and automatic car driving. Thus, the simulation tool for vehicle thesis calculation is also the latest mesoscopic simulation traffic flow tool which is the future research trend. The second strategy, Informed Searching strategies, builds a graph (simulating the road network) for the environment, and uses the search algorithms to detect an efficient path connecting the vehicle with the target. However, many practical systems, to guarantee maximum safety, follow a mixed strategy, i.e. integrating informed-search methods and heuristic cost functions. Consequently, an issue of path planning, which has been given only scarce attention in the scientific literature, is the more efficient, control enforced route tables.

6.2. Data Augmentation Techniques

To increase this capability, different environments are used for training the DRL algorithms and the inference in simulation and real driving especially for safety relevant scenarios [26]. When complex surrounding scenarios with different road user behaviors are frequently presented, the AD is able to predict traffic situations in complex surrounding of multi-lane highways in road traffic instead of only well-known normal highway traffic situations. In this work, a modular training process and system is done to show how different modules can modularly learn and later be assembled and combined.

There are only a few data augmentation techniques available for path planning to support either pure ego-motion prediction or reinforcement learning tasks [27]. Recent algorithms are lacking realistic and critical scenarios such as vehicle interactions in road traffic including emergencies, disfunctions of sensors, or actions of other road users. More testing on synthetic or real-world scenarios can enhance the knowledge of an AD system. Simulations can be especially made for safety-relevant scenarios which occur rarely in the real world and are therefore hard to experience in real driving or training data collection. We propose an extension to the available path planning system using user-specified simulated critical and rare scenarios to generate a diverse and balanced additional augmented data set. Further, the decision-making policy for AD is defined by the observations of the system's environment. It depends on the capability of the actor policy to observe the environment.

7. Designing the Reward Function

To further adapt the agent's policy to the environment, in this paper authors propose a novel path planning algorithm named Smart Search System of Autonomous Flight UAVs (S3AFU). In actual uncertain and complex problems, the UAVs are completely unfamiliar with the map, so it requires the UAV system to have the ability to plan safely and efficiently. An example of this mobile agent problem is shown in Fig. 1. Can reinforcement learning (RL) be used to learn optimal policies for autonomous vehicles so that they can find safely and efficiently routes to a designated destination through unknown and unstructured environments?

Training DQN in this paper is also challenging because the large continuous action space can generate many useless actions. In addition, the agent may continue to optimize the part of the state space found in the early training stage, and the Q value given in the final state space is overestimated. Take the advantages of these two methods, authors propose the PMR-Dueling DQN algorithm. The algorithm adopts a DDPG method to generate a relatively good policy as an initial policy and uses prioritized experience replay to train a DQN to use the policy learned by the DDPG. Meanwhile, a new loss function is designed according to the original DQN loss to adjust the Q value. This can effectively guide the agent to learn a more stable and high-performance path planning policy.

Different authors address the challenge of suitable reward function design for reinforcement learning path planner, and offer different methods [28] [29] [30]. All these researchers highlight potential difficulties associated with the design of the reward function in finding suitable policy rules.

7.1. Safety Constraints

An autonomous vehicle should be able to navigate through different terrains and cities and is expected to be safe as well. Traffic guidelines and differently shaped obstacles can pose challenges for the vehicle [31]. We propose a mid-level architecture that uses deep reinforcement learning (DRL) to get high-level commands from a low-level neural network planner, ensuring collision avoidance. A geometric controller, which tracks the obtained high-level commands, maintains the suggested vehicle trajectory and prevents it from colliding with the obstacles [32]. By breaking the task down to low-level control (i.e., controller), path finding (i.e., mid-level architecture), and obstacle collision avoidance (i.e., neural network planner), the complexity of the proposed system will be reduced. This makes for easier training, and enables safe driving in new scenarios by model extension to a new environment [7].

7.2. Efficiency Metrics

In [4], it is given that planning in a structured environment is divided into three levels; high level route planning which quickly picks the route to the destination based on speed and distance, mid-level lane change decision, and low level vehicle control, to name a few. For route planning, algorithms like A* search, dijkstra and other navigation methods are employed. Neighborhood scenarios based on vehicle fusion and forward simulation are taken under consideration for mid-level lane change decisions in the urban traffic scenario. Using these algorithms, the vehicle plans the trajectory taking safety constraints into account. Real time planning, off-line and on-line planning are used for various complexity of environments. To handle unstructured environment, obstacle avoidance, dynamic path planning are developed. Here, we added a new study list in which the relationship between the efficiency of autonomous vehicle path planning and deep learning is considered and reviewed.

Efficiency is an extremely important concern when it comes to deploying an autonomous path finding system in the real-world. Besides the overall computational expense of the training and inference of the DRL model, we need to take into account the real world constraints such as the safety distance of the vehicle from other objects and the time planners take to come up with an appropriate action to perform i.e., the iteration time. In [31], efficient path chaining is achieved using DDPQ, however they only consider a fixed distance to target to plan the path for. As a consequence, the path chosen might not be the most efficient in terms of time taken to reach the goal. The paper does not consider computational costs of planning as well. There is an extensive literature about efficient planning for both symbolic/numeric and DRL-based approaches in the context of mobile robots and AGV robots. Here, we focus on existing mainstream path planning models and discuss their limitations in terms of efficiency. These efficiency metrics can serve as a theoretical basis to combine different DRL approaches in order to come up with more efficient path planning model for AVs.

8. Training Deep Reinforcement Learning Models

In the physical world, few vehicles were set aside to act as retrieval cars in grid distributions. However, interference and blockage occured frequently, which made it difficult for the approaching vehicle to arrive at the store. Essentially, it is well known that Q-Learning has a well-known issue called overestimation. It means that each action's Q-value is consists of the approximated target state value and the approximated action value. They use the same set of parameters so that action values related to a state tend to be overestimated compared to other action values. Some previous studies mainly focus on how rewards affect the performance of the learning process, but they treat serving vehicles as overwhelming. Double DQN, which is regarded as a solution to mitigate the overestimation issue of DQN, is applied to this task in order to solve the Bellman approximation and double sampling error in order to stabilize the learning process [2].

The environment of a real autonomous vehicle (AV) is partially observable and has a continuous action space. A lot of DRL methods for AVs use pixels as input for the perception process, and most of these methods are inspired by DQN (Double Deep Q-Network) DRL-based algorithms. Neural networks are employed to approximate Q values, which could be then computed by acting on the current state and for all available actions. RoadMapGAN is a map -assisted GAN-based method that attempts to predict a static spatial semantic map. It can integrate with any DRL algorithms and select the best action. Tesla stopped using radar and switched to a pure camera approach for viewing. The appearance of the new driving style makes the AV steering policy continuously change. Engineers expose the vehicle to plenty of typical accident scenarios to help train a more robust steering system [10].

8.1. Experience Replay

There are however some important points to criticize about the algorithm. For example, the algorithm must handle a large memory that it leaves every 2000 entries in a circular buffer and it only uses the memory with Poisson insurance probability in the learning. The Fast Learner exploits the savings from three characteristics associated with the memory: a, sharing the training of many state-action pair matches, b, reducing the total number of visited states or state transitions, and c, saving recent experience at a high level. This paper investigates the efficiency implications of these methods using two simulations with randomly generated paths, including multiple objectives [15].

There is a maxim that close friends are the best textbooks. To realize efficient navigation, autonomous vehicles (AVs) must learn the correct decisions from state transitions in a punishment mechanism. That is, the representative sample of successful decision sequences varies slowly in response to changes in the distributions of the states and such tuning makes the value function learning fast. Thus, it was a crucial problem to provide representative samples of state transitions for learning the value function since these were the building blocks of successful navigation. The reinforcement learning (RL) algorithm must generate and save these sample pairs for every episode for future learning and have replayed these samples in a random order from a large memory bank [6].

8.2. Target Networks

This is the main crucial reason behind employing DQN over vanilla DQN as it will increase the network's output information and hence make it more discriminating [28]. This will lead to concentrating on certain Q-values, hence choosing an action based on extracting the maximum Q-values from this critical information and then ignoring the rest. As a result, the used process make the agent focuses on extracting the important Q-values.

The critical part of the learning phase of the RL algorithms is the function optimization method. For Q-learning we have the Q-function Q(s, a; Θ). If we could utilize two different parameters for target and approximator network, we would expect a "fixed" point for the learning method [33]. This so-called DNN optimization method used for such a task is called 'Target Networks'. In fact, it is possible to have one network for action-value functions which is updated in each step and an (almost) identical network for obtaining target values which remains frozen for a number of received updates.

9. Evaluation Metrics and Performance Analysis

The racing simulation with 520 cm distance is the most demanding maneuver consisting of narrow turns and X-crossing, however, the uncontrollability observed in PPO logic rigonally lowers the overall assessment of path planning methods. It has been observed that up to a distance of 520 cm, DDPG, TRPO, and PPO can perform the planned path more easily. DDPG, which performs most efficiently, is the most used method to perform most of the path planning maneuvers in deep reinforcement learning studies. In the light of these results, it can be concluded that the environment can be passed with a maximum of 12% average fuel advantage observed when the entire maneuver is examined on the average line for distance.

The overall performance of 9.64 m as a controlled variable overlay to mark an obstacle avoidance. It is observed that the RL path planning methods DDPG, TRPO, PPO approaches, included in the theoretical evaluation, line of sight heuristic sequence A *, are moving the fastest on average at 17.5 m / s, compared to DRL methods. However, DDPG, TRPO, and PPO simultaneous path planning for a straight racing segment slightly outperform DRL methods. Similarly, DDPG, TRPO outstrip PPO through an obstacle avoidance maneuver. It can be noted that when the distance between the planned path and the straight racing segment overlap after the obstacle avoidance maneuver, the fastest moving average is in the RL DQN image. The longest time taken to pass both the straight racing route and obstacles as a result of path changes on planned paths with the DQN method.

[ref: 402ec7fa-5e46-48b8-bfcf-7ba546eb2577, 001c593e-d571-4512-a1e0-33a73f36ddce, 972bf393-3d0e-43ef-97f8-6b14a2dc6efc]

9.1. Success Rate

A Deep Q-Network (DQN) is a reinforcement learning technique that determines the best action steps to achieve a particular goal. In an earlier study, individual sensor networks were developed for the autonomous vehicle, with all actions determined by Q-learning methods. Both sensors include a range of ultrasonic, infrared, tactile, GPS, magnetic, gyro, accelerometer, and wheel encoder devices, although the vehicle cannot precisely perceive its location in GPS blind areas and the real-time scenario wherein the GPS system is unavailable. Irrespective of the drawbacks mentioned, the Q-learning method is used for initializing vehicle training through optimal action and finding sequence expressions.

Path planning in an unknown environment by an autonomous vehicle is one of the most important and challenging tasks in robotics. Deciding the next position to reach the goal from the current position and achieving this in the environment without encountering obstacles is called path planning [10]. This process consists of two stages: search and navigation. The aim of the search process is to determine a path and the goal positions of the vehicle in the map.

The navigation task is to determine the vehicle's path by following the previously calculated path in the search process and the searched goal position [29].

9.2. Training Time and Convergence

The would like to measure by exploring the iterate iterations with the whole iterative linearities between the coordinated perceptional structures wherever we obtain an important reward that applies to the picture with all the box model for the Markov process according to the mechanism measured by the trajectory to accomplish host duties. We have tested our earning strategy with a simulated Omnidirectional flying robot at the planet aggregate environment. We provide compara- tive analysis and sequencial chronological structures to help support Vartiation or the random colored propo- tari but not the entire reduchubility of the reduced assembly primary surveying task monitored by the billion laser radionaj algorithm. [34]

Reinforcement learning shows many appeal in robot play planning and its ought to. Experimentally, RL permits to eliminate the would like of deep data on the drawing of true atmosphere and in assigning all issues ranging of physical decisions to alternative factors for setting up. Once operative on the planning task, it'll actively acquire final destination, visit multi-modal actual frequencies of exploitation associate degreed attain centered responses supported associate degree assumed rewards function. Once optimizing the tendency for associations at a similar time, however, the particular benchmark of this learning pattern remains to be maintained. The main results exhibited in this paper is a learning-based redesign of robot path optimizations previously required under the Markovian assumption.

10. Case Studies and Applications

[6] This paper primarily focuses on the achievements and developments related to the comparison of various techniques, performance, and generalization features of navigation planning systems of robotic applications in MRO. The paper describes and analyzes various navigation planning methods using neural networks, deep learning, supervised, unsupervised, and reinforcement learning methods. The paper concludes how the fringe benefit of using learning-based algorithms for navigation planning of robots has successfully configured and downplayed the processing and computing burden in a real-time application environment.[35] The performance of reinforcement learning can be better enhanced by using

Q learning and deep learning techniques. It provides a real-time approach for navigation and flight of UVS which is depicted as a promising application in. Whereby, proposed a new training structure to train the reinforcement learning-based controller on-line using the same device. The deep reinforcement learning-based training process can be integrated and then adapted dynamically during an approach maneuver involving compactness of operations and considering additional constraints such as the limited battery life in two different mobile systems, shown in [1,36].[23] Further, reinforcement learning approaches are also compared by in order to gain insight into the learning performance of differents approaches in different domains. It has been realized that the training process of the reinforcement learning-based controller is a long and difficult process with a high demand, and that the current reinforcement learning-based controller training approaches are not efficient for training controllers for autonomous flight of UVS due to ignorant and inefficient learning.

10.1. Urban Driving Scenarios

Path planning is a key technology for route decision-making to achieve vehicle automatic driving. The traditional path planning mainly uses a state equation to build a dynamic model of the vehicle, and uses an optimization algorithm to optimize the planned path. Due to the large amount of computation and poor robustness of optimization algorithms, it is very difficult for traditional optimization algorithms to generate real-time path for vehicle automatic driving. A new method for path planning by combining modified deep deterministic policy gradient (DDPG) in the context of vector map environment is presented [7]. By comparing three learning algorithms (DQN, A3C, and DDPG), the vehicle's behavior in pedestrian interactive road environment with intersection and different orthogonal recreations—without recreation, partial orthogonal positive recreation, and orthogonal positive recreation in the low visibility of the sleet condition was studied.

Increasing attention has been paid to autonomous vehicles (AV) in recent years. Autonomous vehicles can greatly improve the safety, comfort, and efficiency of automotive transportation. Traffic situations in city regions are rich and varied, presenting additional challenges compared to highway driving. It is important to study the designed autonomous systems to adapt to complex urban traffic environments [36]. In contrast to the previous literature which used rule-based methods and human knowledge, this paper focuses on advancing the capabilities of autonomous vehicles by training a reinforcement learning (RL) agent to achieve high-level goals such as emergency situations for urban driving scenarios.

10.2. Highway Navigation

The present research mainly addresses environment perception for the highway task. However, in environment perception, need to consider the objective of driving planning when designing algorithms. Classic methods are designed to address a particular planning problem and often ignore the driver behavior. In practice, this results in apparent conflicts between trajectory and safety. To address such anomalies, the paper presents a Frenet state space based RL algorithm which circuits driver behavior straightaway. The advantage of this algorithm is that it consist a sophisticated prediction generative network and is most closely compared with previous end to end work, differs from other planned methods, this has the most robust generalization performance.

In autonomous vehicles, intelligent navigation planning and control are particularly challenging when performing in high-density urban traffic [9]. It is generally accepted that highway navigation is a subset of urban traffic and among the most challenging tasks in the domain. In the highway navigation scenario, the vehicle is expected to drive at a high speed. Also, the continuous transitions among different lanes in order to overtake the forecars make the problem of highway navigation more complicated than urban traffic [37]. To drive safely and effectively on highway environment, the trajectory of autonomous vehicle should follow some criterions like safety constraints, jerk/comfort constraints and also need to maximize robot's operation based cost like traveling time and fuel consumption. The previous research paid large attention to solve highway overtaking problem for autonomous vehicles, which they all use driving control and/or motion control to coordinate highway overtaking issues [38].

11. Challenges and Future Directions

The deployment of deep learning models has been shaping up the progress of path planning in the field of autonomous navigation scenario using both single and multiple vehicles [39]. Many new algorithms are proposed to solve the mission and vehicle motion path planning problems in various traffic scenarios by seamlessly involving deep reinforcement learning (DRL) technologies; these then overcome the limits of traditional planning algorithms, which usually require a known environment model from training time to inference time. The intelligent training phase and the dynamic calculation phase in the vehicle onboard neural network incorporate the location data and the vehicle environment perception data to judge the current uncertainty risk. Autonomously navigation the vehicle then allows the planner to estimate the path coordinates including the lane following score. Online trained neural networks have a greater ability to generalize information about similar unseen scenes to online calculate approximate paths between start ego-vehicle and termination positions.

Transportation is vital to the economy and comfort of human life. One way to solve traffic safety, congestion, and energy consumption issues in real-world transportation is to deploy autonomous vehicles (AVs) [4]. Recently, given the advancements in onboard hardware, maps, and perception modules, commercial autonomous vehicles (AVs) have finally achieved improved safety and operation modes. However, developing a safe, convenient, predictable, and comfortable path planner that accounts for various real-world complexities is still challenging. Before the global autopilot is fully open, there remain a few challenges to address, such as the ability of the developed algorithm to achieve socially efficient navigation, which considers the global optimality of the traffic flow in a graph-based space. The ability to plan online – as well as allowing the real-time dynamic changes of the environment – while ensuring the safety, scalability, and react speed of path planning are also important to consider.

11.1. Real-World Deployment Challenges

However, one of the main limitations of this methodology is that it runs without adversarial training and does not permit the selection of the best path from multiple candidates [4]. This is one of the primary potential future optimization goals. The estimates in the global frame and local frame at each position are diverged, and the gap between these errors has been identified. This error shows the ability of the model to guide search at 1/2/3+ timesteps into the future. This work demonstrates that Deep Graph Reinforcement Learning can be employed in a navigation problem. Additionally, researchers propose a loss function to blend the strengths of two popular DGL weights schemes and prove a class of results relating these models' capabilities under certain assumptions. It improves the state of the art for explicit deep learning with graph structures approach. The fact that it is possible to adjust the relative influence of the waypoint graph's topology and relative position within them allows for more freedom in the specification of behavior. Moreover, the proposed methodology allows for the search over these hyperparameters in the learning of the task decider by considering the topology within the task graph.

[12] Deployment of continuous DRL-based A* network" was demonstrated on a 1:10 RC vehicle. Researchers demonstrated proof-of-concept using both learn to plot via RSL and DQN implementations. They showed that similar results could be achieved using A* and DRL. Additionally, researchers designed and developed a robot operating system (ROS1) stack to allow DRL logistics and task planning to control real-world experiments. A basic ROS software implementation of DRL-based logistics planning and configuration processes was designed. Additionally, two smaller finite state machines (FSM) were implemented in the ROS stack; one was a controller node for task-based evaluations (e.g., starting, running, or stopping a single point or multi-point task within the universe), and the other was for logistics vehicle control, which acted as either DRL-trained reward or a navigation file's trajectory. Researchers developed a generalized methodology to integrate reward signals produced by any generic sensor.

11.2. Potential Research Directions

** [12]** The autopilot function for one self-driving car is established mainly aimed at short-haul transportation between locations. The car can automatically find the best way to arrive at the goal position according to the pick-up location and the drop-off location. Furthermore, there is no need to make intersections with complex road planning or any other hard coding, the car can easily move around to go down the shortest node of the road according to the global map, and it can also go straight towards the goal according to the optimal direction of a node. However, at some occasions, the shortest path or the most direct path is often not the most optimal pick-up and drop-off path. Sometimes the so-called shortest path is obstructed, and we have to go a long way. Consequently, this creates abundant alternative paths, but we do not have the most optimal path information. If the vehicle needs to pick up and drop off, how would the car move? Therefore, this paper researches the path plan of the autonomous vehicle based on reinforcement learning and deep neural network, and forms a method reused in the process of motion combining LSTM to handle the storage of the motion trajectory.

** [15]** In cases where an autonomous vehicle can't get to the destination directly, the pick-up location and drop-off location should be reached with a path plan. There is a compact representation of the set of paths called an adjacency matrix, which represents paths in the graph by a bit representation of its adjacency matrix. Compared with the shortest path, this method can find a path with plenty of alternatives for each alternative of the shortest one. The navigation map can provide a globally path plan instead of a concrete path which can only provide a motion based path plan using typical deviation. All these methods are generated from the navigation map. Besides hand-crafting features or learning features from a pre-trained network, there is another potential to learn such features during the network training process. This is the critical disparity of the traditional path plan method and the end-to-end network. Along with the development of the deep learning technology, people can learn specific critical driving features which are specific to driving from end to end.** [13]** Using this technique, the receivable global map information and the learned features of local images can be combined together. The autonomous vehicle can build a driving map over the local perception information without the global map any more.

12. Conclusion and Summary

In conclusion, this survey has analyzed some popular deep reinforcement-learning (DRL) algorithms used in the motion plannings. Besides well-documented algorithms, the classical learning methods like Q-learning and DQN have also been reviewed. And METRL algorithm has been discussed for the motion plannings, and difficulties and challenges in robot learning have been analyzed. Although DRL algorithms have also been widely used in the motion plannings, it is necessary to overcome some difficulties, which has also been analyzed and discussed [11]. So it is still a hot spot to improve the performance of the traditional motion plannings and to develop new algorithm for robot learning with less prior knowledge support. The situation is further complicated due to the dynamic uncertainties during the maneuver process, driving intentions, and environmental changes. Also, in a real-world navigation scenario, how to balance the real-time and computational cost is particularly crucial when we update the motion planning results and re-calculate it after obtaining new available action spaces.

Using deep reinforcement learning (DRL) in autonomous path planning has benefited much from the combination of deep learning and reinforcement learning. DRL methods effectively help agents learn good behaviors directly and optimize for longer-term performance. However, our path planning problem is more challenging due to the high dimension number of states and action space. In this review, we propose a unified framework for the discussion (Fig. 1): the process which generates a series of control instruction, usually denoted as decision-making progress, is formally named motion planning, and how to determine the whereabouts a vehicle should head for is defined as path planning. Both of them can be improved by DRL algorithms [10]. Path planning, the core research objective, may refer to some particular objectives according to different scene attributes, but limited to navigation scenarios. And we also emphasize path planning researches, so path planning scene is referenced in the text when describing algorithms.

12.1. Key Findings and Contributions

In this chapter, we discuss the details of 12+ innovative automated path-planning techniques based on deep reinforcement learning, commonly known as deep learning (DLRP) with notable examples, advantages, and disadvantages. Specifically, under the recent remarkable success of machine learning-based technologies, several researchers aim to offer an efficient collision-free path planning mechanism using DRL. The core concept of the planners in this chapter is to plan and select the most efficient path to the destination for each wheeler based on its uncompressed sensor data accumulated from the past experience in the absence of expert knowledge (related to the optimal path for each wheeler). Indeed, the pioneering deep reinforcement learning planner, which is termed a learning path planner (LPP), has been recently developed to offer dynamic movement to large and continuous environments and is used to circumvent static and dynamic constraints alongside the optimal path provided in the real world to offer collision-free path planning for the autonomous wheeler.

During the last few decades, considerable research has been devoted to the intelligent design of innovative techniques for achieving collision-free, smooth, and optimal path planning for wheelers. Specifically, graph search-based methods, including A star, rapid exploring random tree (RRT), and dynamic programming-based approaches, have been widely used to solve path planning problems in autonomous vehicles. As time goes by, deep reinforcement learning (DRL) has started to demonstrate its capabilities for intelligent decision-making in different types of vehicles, which have propelled remarkable technological advancements in multiple fields such as unmanned vehicles and autonomous driving systems Nowadays, DRL has been employed in most autonomous vehicles to represent and solve sequential decision-making problems, and deeplearning-based path-planning techniques have shown promising performance in large environments with complex and dynamic obstacles. The primary focus of this chapter is exploring the development and challenges of automated path planning using deep reinforcement learning .

12.2. Final Remarks

The present study has aimed the task of autonomous path planning with a deep reinforcement learning algorithm. A state-based method has been exploited and the results of trials over carla vehicle driving simulator datasets have been presented for both random and follow the leader second agent based path planning [10]. The second agent is used to simulate the performance of the method for multi-agent case aspherical. The learning task has been solved by a long-term reward with the purpose of specific objectives while considering the environment's reward separately. The second objective has helped to test the capability of learning exploitation-exploration strategies on different datasets through diverse streets. The resulting comparisons have shown that the results are successful based on comparing multiple observation methods utilizing their ground truth results. The explored actor-critic learning with an improved reward mechanism has shown an enhanced training agent on various datasets.

Path planning for the autonomous driving of vehicles using the deep reinforcement learning approach has been a focus of researchers in recent years. The technical requirements for such approaches also vary based on the task and the desire to provide robustness for the planning. In this context, it can be seen that different methods have been developed for solving distinct problems related to the path planning of autonomous vehicles [19]. However, in some approaches, there is a trade-off between the training time of the algorithm and the robustness and efficiency of the system. A trade-off between the proposed planning method and the decision time of the algorithm may have been made during trials in different parts of the study. Therefore, efforts continue to improve in this field by developing novel models or using different deep learning algorithms for path-planning tasks. The new deep learning algorithms provide a method that is able to make decisions more calculated and quickly in path-planning tasks, which are one of the important steps of autonomous driving.

Reference:

1. Tatineni, Sumanth, and Venkat Raviteja Boppana. "AI-Powered DevOps and MLOps Frameworks: Enhancing Collaboration, Automation, and Scalability in Machine Learning Pipelines." *Journal of Artificial Intelligence Research and Applications* 1.2 (2021): 58-88.
2. Ponnusamy, Sivakumar, and Dinesh Eswararaj. "Navigating the Modernization of Legacy Applications and Data: Effective Strategies and Best Practices." Asian Journal of Research in Computer Science 16.4 (2023): 239-256.
3. Shahane, Vishal. "Serverless Computing in Cloud Environments: Architectural Patterns, Performance Optimization Strategies, and Deployment Best Practices." *Journal of AI-Assisted Scientific Discovery* 2.1 (2022): 23-43.
4. Muthusubramanian, Muthukrishnan, and Jawaharbabu Jeyaraman. "Data Engineering Innovations: Exploring the Intersection with Cloud Computing, Machine Learning, and AI." *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* 1.1 (2023): 76-84.
5. Tillu, Ravish, Bhargav Kumar Konidena, and Vathsala Periyasamy. "Navigating Regulatory Complexity: Leveraging AI/ML for Accurate Reporting." *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* 2.2 (2023): 149-166.
6. Sharma, Kapil Kumar, Manish Tomar, and Anish Tadimarri. "Optimizing sales funnel efficiency: Deep learning techniques for lead scoring." *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* 2.2 (2023): 261-274.
7. Abouelyazid, Mahmoud. "Machine Learning Algorithms for Dynamic Resource Allocation in Cloud Computing: Optimization Techniques and Real-World Applications." *Journal of AI-Assisted Scientific Discovery* 1.2 (2021): 1-58.
8. Prabhod, Kummaragunta Joel. "Utilizing Foundation Models and Reinforcement Learning for Intelligent Robotics: Enhancing Autonomous Task Performance in Dynamic Environments." *Journal of Artificial Intelligence Research* 2.2 (2022): 1-20.
9. Tatineni, Sumanth, and Anirudh Mustyala. "AI-Powered Automation in DevOps for Intelligent Release Management: Techniques for Reducing Deployment Failures and Improving Software Quality." Advances in Deep Learning Techniques 1.1 (2021): 74-110.