UDC 004.942

V. Oliinyk, Y. Danyliuk

AUTONOMOUS CAR PARKING MODEL FOR DIFFERENT TYPES OF PARKING LOTS USING DEEP REINFORCEMENT LEARNING

Abstract: This study presents a deep reinforcement learning model for autonomous parking and simulation environment for parking process modelling and research. Proposed model utilizes PPO with Behavioral Cloning and Adversarial Imitation Learning and achieves 96.3–99.34% accuracy in virtual simulations across various parking lot types. The Unity-based environment provides advanced visualization, simulation, and modeling, proving valuable for educational and research purposes while enhancing parking strategies across diverse scenarios.

Keywords: automatic parking, self-driving car, deep reinforcement learning, Proximal Policy Optimization, ML-Agents, Unity, virtual environment.

Introduction

Actuality of parking problem and its issues

According to annual statistics, one in five (20%) vehicle accidents occurs in parking lots. Despite this, people often underestimate the risks associated with such locations, primarily due to factors like distracted driving, limited visibility, and driver impatience.

Parking can be particularly challenging for many drivers, especially in confined spaces. Insufficient coordination and, in some cases, vision impairments further complicate the process, increasing the likelihood of accidents.

The arrangement of parking spaces within a lot, particularly the angle at which they are aligned, plays a significant role in determining the ease and efficiency of parking maneuvers. Automating the parking process in lots with various parking angles, ranging from 90 degrees to 0 degrees, is therefore a crucial and actual task for reducing accidents and enhancing safety.

Solution for parking problem

A common solution to these challenges is the development of autonomous systems capable of effectively parking a car in various situations without driver involvement. These systems not only help drivers avoid inconvenience but also ensure a safer and more efficient parking process. They are particularly beneficial for individuals with disabilities or elderly drivers, who may face difficulties with parking.

Over the past few decades, extensive research on autonomous driving and self-driving cars has been conducted by both academic communities and industry stakeholders [1]. However, this research will focus specifically on the automated parking problem, encompassing both fully autonomous systems and decision-support solutions designed to enhance the parking process.

Related research

Existing methods

The problem of automatic car parking falls into the category of complex control problems in dynamic and unstable environments. A variety of approaches can be employed to address this challenge, including machine learning, evolutionary algorithms, logic programming, and others.

Among these, deep machine learning stands out as the most promising due to its state-of-the-art performance in numerous similar control tasks [2].

Considering the specific challenges of parking a car in a virtual environment, deep reinforcement learning [3] was selected as the optimal machine learning approach for this problem. Deep learning enables the consideration of diverse factors, such as the car's shape and size, the placement of obstacles, and other parameters influencing the parking process. Meanwhile, reinforcement learning adds the ability to operate in real-time and adapt through self-learning based on interactions with the environment.

Related publications

After selecting reinforcement learning as the foundational approach, several existing implementations of car parking simulators were identified and analyzed.

The implementation proposed by Leonardo Lai in [4] uses the classical Q-learning algorithm. This straightforward solution is easy to implement and configure, and it demonstrates robustness when training on various types of parking lots. However, it is significantly less efficient for complex tasks, prone to training overload, and requires a higher number of iterations to produce satisfactory results.

The implementation described in [5] utilizes deep reinforcement learning with the Deep Q-Network (DQN) algorithm. This approach achieves high accuracy in automatic parking. Nevertheless, it is less robust when applied to diverse parking lot scenarios and involves a high level of complexity in configuring and optimizing the deep learning network.

The solution proposed in [6] is based on a reinforcement learning method that incorporates a real-time approach to car parking planning. It enables rapid and precise parking even in uncertain and dynamic environments. However, this method is challenging to implement and heavily relies on the accuracy of input data and domain templates.

Overall, each of the reviewed implementations offers unique advantages and disadvantages. Some excel in resource efficiency, while others are more reliable and accurate in complex parking scenarios. We believe there is significant potential in combining modern state-of-the-art reinforcement learning algorithms with specialized simulation environments to achieve more precise and robust results.

 ISSN 1560-8956

Proposed solution

Parking simulation environment

For a high-quality and realistic car parking simulation, the Unity game engine [7] was utilized. This choice was driven by its ease of use, rapid development capabilities, extensive library of modules and tools for machine learning, comprehensive documentation, and strong support from the developer community.

The Machine Learning Agents (ML-Agents) plugin [8] was used to integrate machine learning into the project. This enabled the creation of agents capable of learning and adapting their behavior in response to environmental changes. At the time of development, the latest stable version of the plugin, 2.0.1, was used.

Parking lots modelling

Since this article focuses on enhancing parking performance across various lot types, it is essential to examine their modeling in our implementation. As high detail was not required, we opted to use Unity's built-in tools to create the models ourselves. To ensure realism, we referenced drawings of popular parking lot layouts from the Dimensions.com database, which includes accurate schematics and dimensions [9].

The database includes layouts for both American and European parking standards. For all parking lots in this study, the European standard was used, as it provides less room for maneuvering, making parking more challenging and suitable for testing.

Following the same approach, parking lots were created for 60°, 45°, 30°, and 0° configurations. Fig. 1 illustrates supported mock-ups of parking lots with their corresponding reproductions, created using Unity.

Agent (cars) modelling

All game objects were designed using Unity's built-in tools, except for the cars. For vehicles, pre-made models from the official Unity Asset Store, specifically, the "Low Poly Soviet Cars Pack" [10] were utilized.

Figure 1. Parking Lot Mock-ups from Dimensions.com (top),
Unity Models (bottom)

Proposed intelligent model

The general scheme of the proposed simulation environment with an intelligent model for autonomous parking is shown in Fig. 2.

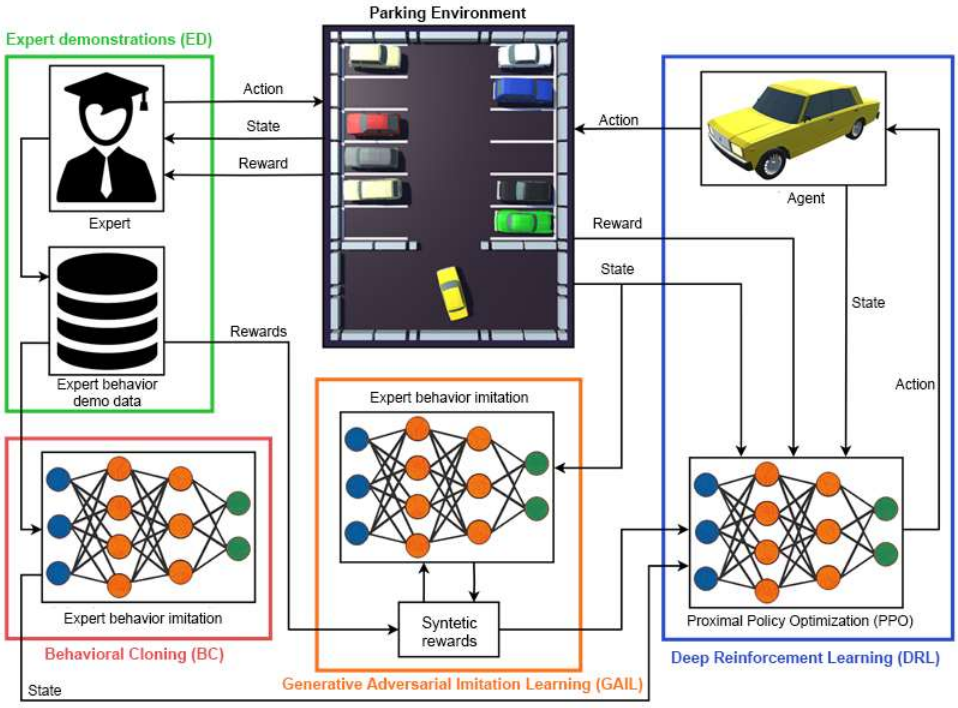

Figure 2. Proposed autonomous car parking model and simulation environment

The primary actor in the parking simulation environment is the **Agent**. In reinforcement learning, the key characteristics of an agent are its observations, actions, and rewards.

An agent's observations refer to the environmental information that it processes, serving as inputs to the deep reinforcement learning neural network (DRLNN). Our agent uses the following observations:

– The **location** of the intelligent agent, i.e., the car it controls in the environment (position and turning angle).

– The **location** of the target parking space in the environment (position and rotation angle).

– The **speed** of movement and rotation of the agent in the environment.

– The **distance** to obstacles, captured by 12 sensors placed around the car.

We relied solely on local devices for observations, though incorporating external information, such as GPS [11, 12] or video cameras for object detection and identification [13] and tracking [14], could provide additional benefits

The outputs of the Deep Reinforcement Learning Neural Network (DRLNN) are the agent's actions, which determine its ability to change its state and navigate toward the goal. The following actions are used:

 ISSN 1560-8956

– The **torque force** of the wheels of the agent's car (the force with which the car moves forward or backward).

– The **steering wheel rotation angle** of the agent's car (the angle at which the car turns left or right).

– The **brake state** of the agent's car (a logical true or false value that controls the braking process).

At each step of the simulation, the agent's performance is evaluated, and a series of rewards are assigned. The rewards reflect the quality of the actions taken by the agent and are used as inputs to the DRLNN. Specific quantitative values for the rewards are shown in Fig. 3.

The Agent is driven by proposed intelligent model where the core component is a deep reinforcement learning algorithm. The Proximal Policy Optimization (PPO) algorithm [15] was chosen as a widely recognized, state-of-the-art solution that delivers the most accurate and reliable results for similar tasks and scenarios.

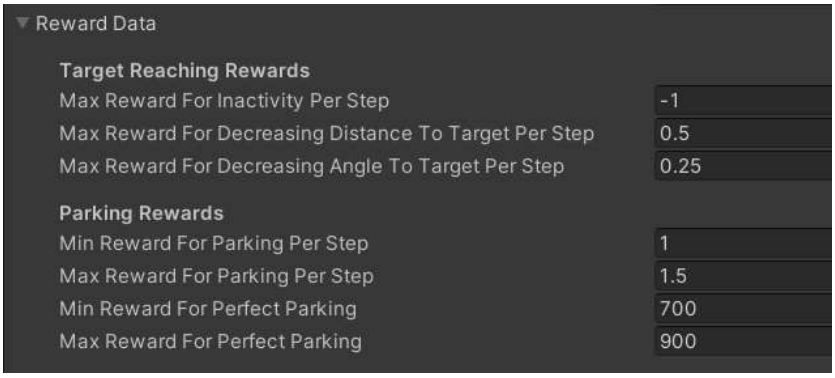

Figure 3. Configuration of parking agent rewards in Unity

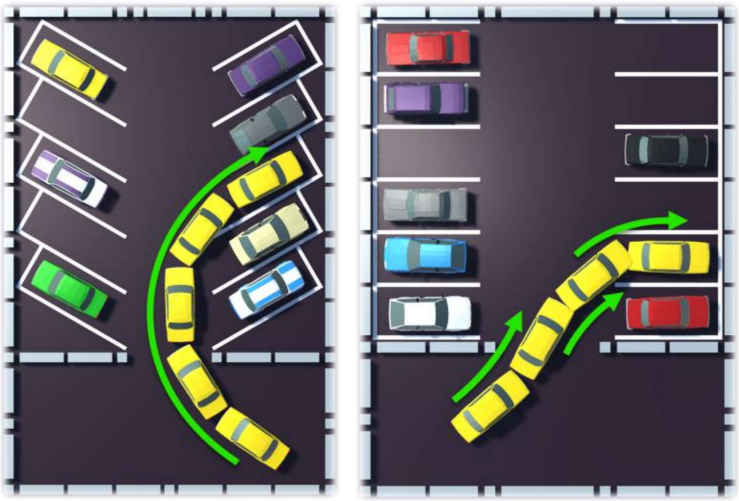

Figure 4. The parking process for the final model in parking lots with angles of 60° and 90°

To further enhance training outcomes and improve parking quality, two additional deep learning methods were implemented within their respective modules:

1. **Behavioral Cloning**: This method was applied to accelerate agent training [16]. During the initial 750,000 steps, agents were trained to replicate expert behavior recorded in a demo file.

2. **Generative Adversarial Imitation Learning (GAIL)**: GAIL integrates deep learning and generative tools to train agents based on expert demonstrations [17]. It extends behavioral cloning by considering the negative consequences of actions, enabling the development of complex strategies that may not be explicitly defined by experts. Furthermore, GAIL equips agents to effectively interact with unfamiliar environments.

Table 1. **Interpretation of Parking Efficiency Criteria**

Criterion of parking efficiency	Interpretation of meaning
Episode length (steps)	A period of time consisting of a sequence of steps in which an intelligent agent makes decisions and interacts with the environment. This reflects parking speed.
Parking accuracy (%)	The percentage ratio of cars that were successfully parked compared to those that were not parked.
Perfect parking accuracy (%)	The percentage ratio of cars that were parked perfectly to those that were parked satisfactorily. This reflects the quality of parking.
Parking distance (meters)	The distance from the centre of the car to the centre of the designated parking space.
Parking angle (degrees)	The angle between the car and the ideal parking angle for the corresponding space.
Number of agent collisions with cars/curbs (%)	The percentage ratio of cars that collided with foreign objects compared to those that did not collide.

Experimental Results

Over 100 model trainings were conducted to test and fine-tune the proposed model and our solution overall. The training involved various types of parking lots, ensuring a comprehensive coverage of parking scenarios and enabling the analysis of their impact on parking performance. As a result, we achieved a fully functional and robust model optimized for high performance.

Fig. 4 visualizes the process of parking for different parking lots. We can see high quality trajectories of the car driven by the final model. These examples are rather straightforward but in more complex environments, especially in case of parallel parking, the agent needs to make more manoeuvres, often using reverse.

Parking Efficiency Criteria

The primary objective of the work performed was to enhance the efficiency of the car parking process within the virtual environment. This goal was achieved by improving key criteria such as parking accuracy and speed, parking distance and angle, and minimizing

collisions with obstacles. Additionally, the accuracy of perfect parking was considered. Table 1 outlines the meaning of each of these criteria.

Final results of agent parking

After the fine-tuning process was completed, several final test sessions were conducted on different types of simulated parking lots. Example for parallel parking is shown in Fig. 5.

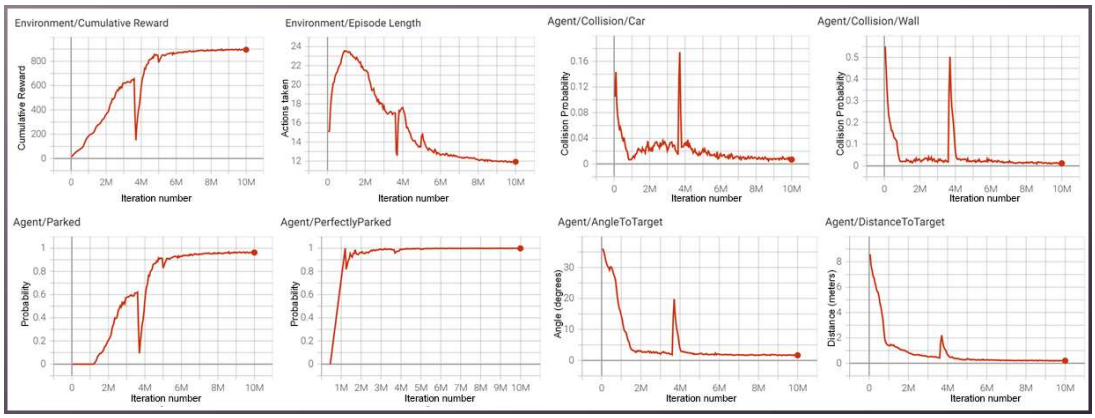

Figure 5. Graphs of key performance indicators
of the final model during parallel parking simulation

The registration of the selected quality indicators was performed programmatically using the ML-Agents API [12]. This process generated a substantial amount of statistical data and various graphs, providing insights into the learning progress of the intelligent agent.

All collected information was thoroughly analyzed, and the most significant results from training the intelligent parking model were summarized in **Table 2**.

Table 2. **Final Results of Agent Parking on Different Types of Parking Lots**

Parking efficiency criterion	**Type of parking lot**				
	90°	**60°**	**45°**	**30°**	**0°**
Episode length (steps)	219.2	182.16	177.84	154.32	238.6
Parking accuracy (%)	98.47	99.19	99.19	99.34	96.3
Perfect parking accuracy (%)	95.73	97.01	95.81	99.37	99.87
Parking distance (meters)	0.16	0.145	0.163	0.161	0.201
Parking angle (degrees)	2.044	1.476	1.538	1.466	1.673
Number of agent collisions with cars (%)	0.693	0.182	0.099	0.035	0.673
Number of agent collisions with curbs (%)	0.693	0.444	0.533	0.558	1.190

From Table 2, it is evident that the agent trained using Unity ML-Agents can successfully park cars at angles ranging from 0° to 90° with high accuracy. Specifically, the parking accuracy averages between **96.3%** and **99.34%,** demonstrating high performance.

Analyzing the data from the table, several conclusions can be made regarding the agent's training effectiveness and characteristics:

1. The agent performed better at smaller parking angles, such as 30° and 45°, compared to larger angles like 90° and 60°. This indicates that the agent’s parking accuracy improves as the parking angle decreases. For instance, when parking at a 30° angle, the agent achieved an accuracy of **99.34%,** whereas at a 90° angle, the accuracy was slightly lower to **98.47%.** This trend can be explained by the increased difficulty of parking at larger angles. At higher angles, the car needs to execute more complex turns and maneuvers to align perfectly within the designated space. Conversely, smaller parking angles simplify the process, enabling the agent to achieve greater accuracy and efficiency.

2. The highest parking accuracy (**99.34%**) and the lowest number of collisions (**0.593%**) were achieved at a parking angle of 30°. This result can be attributed to the simplification of the parking task, as it involves fewer possible agent states and a reduced number of required maneuvers to complete the parking process.

3. The model demonstrated its lowest performance during parallel parking, primarily due to the inherent complexity of this parking type. Parallel parking requires navigating extremely limited spaces, executing numerous precise maneuvers, and maintaining minimal distance to surrounding vehicles. Experimental results confirm that parallel parking is one of the most difficult tasks for any driver, including AI based solutions.

It is important to note that the agent does not achieve zero collisions with cars or curbs in any of the experiments. However, these collision values are exceptionally low and do not significantly impact the agent's overall performance.

Special attention should be given to parking lots designed for 90° and 0° angles. These configurations are not only the most common but also among the most complex to navigate. At the same time, they offer a highly compact design, making them advantageous for space-saving purposes. Through research and experimental validation, it becomes evident that modern parking lot architects should consider these characteristics when designing parking facilities.

Conclusion

Proposed solution demonstrates highly efficient and high-quality car parking in a virtual environment. Our fine-tuned intelligent model shows high accuracy, speed and safety of car parking. The obtained results clearly confirm the success of using reinforcement learning and the PPO algorithm in the parking simulator. Experiments were conducted on different types of parking lots and was high in all scenarios. It was found that with a decrease in the initial parking angle, the agent shows better results (if parallel parking is not taken into account).

Based on the results obtained, the developed parking simulator demonstrates significant potential for integration into educational and training programs for drivers. The

proposed model can also serve as a component of autonomous vehicle control systems, enhancing parking efficiency and reducing the risk of accidents. Furthermore, owing to the flexibility of the simulation environment and its state-of-the-art performance, the software product holds substantial value for research purposes, such as exploring and identifying optimal car movement strategies across various scenarios.

Future research will be focus on enriching observations of the agent by adding video camera [13] and processing its input and securing such solution [18].

REFERENCES

1. Level-5 autonomous driving - are we there yet? A review of research literature / Khan M. A. et al. // ACM Computing Surveys (CSUR), Vol. 55, No. 2, 2022. – P. 1–38.

2. Mrinal R. Bachute, Javed M. Subhedar. Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms // Machine Learning with Applications, Vol. 6, 2021. URL: https://doi.org/10.1016/j.mlwa.2021.100164.

3. Lapan M. Deep Reinforcement Learning Hands-On (2nd Revised edition). Packt Publishing, 2022.

4. Lai Leonardo. (2020). "Automatic parking with Q-Learning". URL: https://leoll2.github.io/Autoparking/docs/paper_short.pdf (application date: 15.01.2025)

5. Junzuo Li, Qiang Long. (2021, March 26). An Automatic Parking Model Based on Deep Reinforcement Learning // Journal of Physics: Conference Series. Volume 1883, 2nd International Conference on Computer Information and Big Data Applications, 2021.

6. Yuzheng Zhuang, Qiang Gu, Bin Wang, Jun Luo, Hongbo Zhang, Wulong Liu. (2018). Robust Auto-parking: Reinforcement Learning based Real-time Planning Approach with Domain Template // NIPS 2018 Workshop on Machine Learning for Intelligent Transportation Systems (MLITS), 2018.

7. Unity". Unity Technologies. https://unity.com (application date: 15.01.2025)

8. "ML-Agents (Machine Learning Agents)". Unity Technologies. URL: https://unity.com/products/machine-learning-agents (application date: 15.01.2025)

9. "Dimensions.com". Dimensions. URL: https://www.dimensions.com (application date 15.01.2025)

10. Tyavin Vitaly. (2020, December 8) "Low Poly Soviet Cars Pack" [Unity asset]. Unity Asset Store. URL: https://assetstore.unity.com/packages/3d/vehicles/low-poly-soviet-cars-pack-184453. (application date: 15.01.2025)

11. Олійник В.В., Яременко Є.А. Алгоритм уточненого позиціонування в навігаційних системах доповненої реальності // Міжвідомчий науково-технічний збірник «Адаптивні системи автономного управління», К: Політехніка, 2018. Т.2, №33. – С. 56-65.

12. Oliinyk V. Method for improving accuracy of mobile AR navigators // ISJ Industry 4.0, Vol. 5, Is. 1, 2020. – P. 21-22.

13. Oliinyk V., Ryzhiy A. An efficient face mask detection model for real-time applications // Adaptive systems of automatic control, 2022. Vol. 1, №40. – P. 54-64.

14. Пантелеев А.С., Олейник В.В. Метод визуального мультитрекинга в реальном времени на основе корреляционных фильтров// Міжвідомчий науково-технічний збірник "Адаптивні системи Автоматичного Управління", К: Політехніка - 2018. - Т.1, №32 – С. 97-106.

15. OpenAI Baselines PPO. URL: https://openai.com/research/openai-baselines-ppo (application date: 15.01.2025)

16. Tung Nguyen, Qinqing Zheng, Aditya Grover. (2022, October 11). Reliable Conditioning of Behavioral Cloning for Offline Reinforcement Learning. URL: https://arxiv.org/abs/2210.05158 (application date: 15.01.2025)

17. Jongcheon Park, Seungyong Han, Sangmoon Lee. Restored Action Generative Adversarial Imitation Learning from observation for robot manipulator // ISA Transactions, Volume 129, Part B, 2022. – P. 684-690.

18. Hatsan S., Oliinyk V. Computer vision based authentication model with spoofing protection // The International Conference on Security, Fault Tolerance, Intelligence ICSFTI2024 (June 07, 2024, Kyiv, Ukraine), 2024. P. 1-12. URL: https://icsfti-proc.kpi.ua/article/view/308401 (application date: 15.01.2025)

 ISSN 1560-8956