Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 33, Num. 1 (2026) 50-60

REVIEW ARTICLE

# A Systematic Literature Review of Autonomous Line-Following Robots using Reinforcement Learning

## Uma Revisão Sistemática sobre Robôs Autônomos Seguidores de Linha que Usam Aprendizagem por Reforço

Matheus Amorim Pereira[1]*, Hianna Araujo dos Reis Soares[1], Sara Dereste dos Santos[1], Ricardo Pires[1]

**Abstract:** Mobile robots have gained importance due to its capability to perform autonomous tasks. Line-following robots are mobile robots which have become a growing topic of research, as they serve as a platform for understanding complex concepts, such as control, which makes them suitable for the application of Artificial Intelligence (AI). There are competitions around the world focused on this kind of robots. Accurate mathematical modeling of line-following robots and the environment in which they will be used can be difficult. Reinforcement learning is an AI technique that deals precisely with creating model-free algorithms, which are suitable for this application. This review article aimed to look for articles on line-following robots using reinforcement learning in order to analyze the state of the art of this branch of artificial intelligence in robotics.
**Keywords:** Reinforcement Learning — Line-following Robot — Machine Learning — Artificial Intelligence — Education — Systematic Literature Review

**Resumo:** Robôs móveis ganharam importância devido à sua capacidade de realizar tarefas autônomas. Robôs seguidores de linha são robôs móveis que se tornaram um tópico crescente de pesquisa, pois servem como uma plataforma para a compreensão de conceitos complexos, como Controle, o que os torna adequados para a aplicação de Inteligência Artificial (IA). Existem competições ao redor do mundo focadas neste tipo de robôs. A modelagem matemática precisa de robôs seguidores de linha e do ambiente em que serão usados pode ser difícil. Aprendizado por reforço é uma técnica de IA que lida precisamente com a criação de algoritmos que dispensam o uso daqueles modelos, sendo adequados para esta aplicação. Este artigo de revisão teve como objetivo procurar por artigos sobre robôs seguidores de linha utilizando aprendizado por reforço, a fim de analisar o estado da arte deste ramo da Inteligência Artificial em Robótica.
**Palavras-Chave:** Aprendizagem por Reforço — Robô Seguidor de Linha — Aprendizagem de Máquina — Inteligência Artificial — Educação — Revisão Sistemática da Literatura

[1] *Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP), Brasil*
***Corresponding author**: engpereiramatheus@gmail.com
**DOI:** http://dx.doi.org/10.22456/2175-2745.150105 • **Received:** 08/09/2025 • **Accepted:** 20/12/2025


## 1. Introduction

In recent years, robotics has become widely used in a variety of applications. Among them, mobile robots have gained importance due to its capability to perform autonomous tasks [1, 2, 3]. Line-following robots [4, 5, 6], in particular, have become a growing topic of research, as they serve as a platform for understanding complex concepts, such as control, which makes them suitable for the application of Artificial Intelligence, including Machine Learning techniques, with Machine Learning being "a branch of artificial intelligence focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their performance and accuracy through experience and exposure to more data" [7].

An example of this are the many competitions around the world focused on line-following robots. In most of these championships, participants build their own robots, respecting stipulated standards, such as the robot's size and weight, and the winner is the one that completes the route in the shortest time. Typically, the robot needs to follow a defined route autonomously, with no restrictions on the techniques used, which is a good opportunity to test different algorithms and control techniques. These events are mostly aimed at high school students or those who are beginning their undergraduate studies and also serve as a way to excite them in the context of teaching in the areas of science, technology, engineering and mathematics (STEM) [8, 9, 10, 11, 12, 13, 14].

The problem of keeping a robot moving autonomously along a line is a control problem. Classically, a good solution to this type of problem requires having a good mathematical model of the system being controlled [15]. However, accurate mathematical modeling of line-following robots and the environment in which they will be used can be difficult. It is common for students to assemble these types of robots from commercially available components that present great variability between samples. The environment in which the robot will be used may be unknown beforehand and may present variations in lighting and other conditions that make predictability difficult during design. Therefore, it is desirable for the robot to use a control algorithm that allows it to adapt to uncertainties about its parameters and to varying environmental conditions. One branch of reinforcement learning deals precisely with creating model-free algorithms [16, 17, 18], making it suitable for this application.

In supervised learning algorithms, the system learns a mapping from input to output variables, from which example values are provided to them in a training phase. In contrast, reinforcement learning is a form of learning by which an agent learns an action strategy through trial and error, rewards and punishments received during operation [16]. The agent's goal is to develop a policy that maximizes cumulative rewards over time. This requires not only choosing actions that yield immediate rewards but also devising strategies to reach more favorable states in the future. The interaction between the agent and the environment can be described by three fundamental elements: state, actions, and rewards. The state represents a particular configuration of the agent and/or environment, the actions are the options that the agents have to interact with to modify the environment, and the reward is a signal used to define the task of the agent and is what motivates the agent to pick one action with respect to the others [17]. The line-following robot can be guided by mathematical algorithms modeled on a Markov Decision Process, common in reinforcement learning tools.

A widespread example of a reinforcement learning algorithm is Q-learning [16]. To use it, it is not necessary to have a model of the environment in which the agent will be inserted, nor a mathematical model of the agent itself. The characteristics of the environment and of the agent can change over time. The agent may constantly adapt its behavior to these characteristics.

Q-Learning uses a table of states and actions, which can have initial values equal to zero. The number of states and the number of possible actions must be finite. If either of these numbers is infinite, its range of values can be partitioned into a finite number of subranges. The Q-table is populated with real numbers as the agent learns through performing actions and receiving rewards.

At each step of the algorithm's execution, the Q-table is updated using the expression (1):

$$Q^{\text{new}}(s_t,a_t) \leftarrow (1-\alpha)\cdot Q^{\text{old}}(s_t,a_t) + \alpha\cdot\left(r_{t+1} + \gamma\cdot\max_a Q(s_{t+1},a)\right) \tag{1}$$

in which

$s_t$ is the state at time $t$ and indexes a row in Q-table,

$a_t$ is the action executed at time $t$ and indexes a column in Q-table,

$s_{t+1}$ is the state at time $t+1$, reached as a result of action $a_t$,

$Q^{\text{old}}(s_t,a_t)$ is the value that was in $Q(s_t,a_t)$ at time $t$, before the execution of action $a_t$,

$Q^{\text{new}}(s_t,a_t)$ is the updated value in $Q(s_t,a_t)$, after the execution of the action $a_t$,

$\alpha$ is the learning rate,

$r_{t+1}$ is the reward value, known after performing action $a_t$,

$\gamma$ is the so-called *discount value*,

$\max_a Q(s_{t+1},a)$ is the largest value available in the $Q$ table in the row of state $s_{t+1}$ reached after executing action $a_t$.

Thus, each time an action $a_t$ is executed in state $s_t$ at time $t$, the cell $Q(s_t,a_t)$ in the table is updated.

The learning rate $\alpha$ determines the relative weights that will have the old value of $Q(s_t,a_t)$ and an update to its new value. $\alpha$ may start with a value of 1 and have this value gradually reduced over time, until it reaches a minimum value chosen by the algorithm's user. When $\alpha$ is 1, that is, when the system needs to learn a lot (because it hasn't learned anything yet), $Q^{\text{old}}(s_t,a_t)$ will have no influence on $Q^{\text{new}}(s_t,a_t)$. As time passes and $\alpha$ decreases, the experience accumulated in $Q^{\text{old}}(s_t,a_t)$ begins to have increasing weight in determining $Q^{\text{new}}(s_t,a_t)$. Thus, the system continues to incorporate, in part, new experiences, but increasingly values the experience already acquired [16].

The value $\max_a Q(s_{t+1},a)$ in the expression (1) takes into account the extent to which the state reached as a result of an action is on the path to a sequence of states that provides good rewards. It causes an action to tend to be considered good if it leads the agent to a row of the Q-table where there is a large positive number in a certain column, because, in that row (in that state), the agent can, soon after, choose the action corresponding to that large number. The factor $\gamma$ must satisfy $0 \le \gamma < 1$ and can vary over time. If it is 0, only the immediate reward will contribute to updating the Q-table, with no vision of the future. The higher the value of $\gamma$, the more the long-term value will be valued in the agent's behavior. However, the value of $\gamma$ must remain below 1, so that there is no numerical instability in the algorithm's execution, to prevent that the values in the Q-table explode over time. As the agent experiments with different actions in different states, a "backward" propagation of information about the best experiences occurs. For example, if, in a certain state $s_X$, a certain action $a_X$ provides a very good (very positive) reward, the value in cell $Q(s_X, a_X)$ should increase. Later, if the agent is in state $s_Y$ and performs an action $a_Y$ that takes it to state $s_X$, the high value $\max_a Q(s_X, a)$ will tend to raise the value of $Q(s_Y, a_Y)$, indicating that, when the agent is back in $s_Y$, it will be good for it to choose action $a_Y$ because it will take it to state $s_X$, where action $a_X$ provides a high reward. Then, when the system is in state $s_W$ and executes an action $a_W$ that takes it to state $s_Y$, $Q(s_W, a_W)$ will tend to increase because of $\max_a Q(s_Y, a)$ and so on. Thus, the agent will learn a good sequence of actions to execute when it finds itself in those states again.

A limitation of Q-learning is that it can only handle a finite number of states. An alternative reinforcement learning system capable of handling a continuous state space is called Deep Reinforcement Learning [19]. Its basic characteristic is that it replaces the Q-table with a neural network trained with data of the same types as Q-learning.

The problem of the line-following robot can become quite complex and sophisticated, which is why it has been common to work on it in academic environments with multidisciplinary groups. Competitions, in turn, end up encouraging the search for increasingly better solutions, which results in the advancement of several areas. Although a classic solution is perfectly possible to control a line-following robot, it is undeniable that solutions based on Artificial Intelligence algorithms are increasingly being used in these tasks, since they allow for faster problem solving. In the specific case of reinforcement learning, it is even more beneficial, considering that line-following robots can typically be exposed to environments with variations in light or routes and can quickly adapt to different scenarios, something that a classical control algorithm will have difficulty to handle. For this reason, this review article aimed to look for line-following robots using reinforcement learning in order to analyze the state of the art of this branch of artificial intelligence in robotics.

## 2. Method

The adopted search strategy consisted in searching papers using the Google Scholar search engine, due to its comprehensiveness. The keywords were defined based on the results of some preliminary tests, after which most of the papers returned were related to reinforcement learning in line-following robots. The resulting search string was

> "reinforcement learning" "line tracking OR follower OR following" "education" 2020-

As this was a state-of-the-art study in a rapidly developing field, it was decided to focus the analysis on articles published since 2020. This search process was concluded in March 2025.

The selection of articles from the search results on reinforcement learning in line-following robots was performed with the following inclusion and exclusion criteria:

1. Articles that present the methods or algorithms of reinforcement learning used in line-following robots were included.
2. Articles published in conferences or journals were included. Books and theses were excluded.
3. Articles that allow open or institutional access to their full-text by the authors of the present article were included, otherwise excluded.
4. Articles that are original research were included, reviews were excluded.
5. Only articles that were published between 2020 and 2025 were included.
6. Only peer reviewed articles were included.

## 3. Results and Discussion

This section presents and discusses the results of this systematic review. Few of the papers found are strongly related to its scope. Nevertheless, some of the less related works still provided valuable information and deserve mention. Figure 1 represents the articles selection procedure, with its quantitative results.

Among the nineteen articles selected, the one that proved most relevant to the scope presented was the article by Cota et al [20]. The authors used a line-following robot model from Amazon, with its development environment, with educational purpose, at undergraduate level, employing deep Q-learning [21] in its control. The Amazon's Deepracer kit used had two cameras to generate stereo vision, a LiDAR to detect objects in 360 degrees, accelerometer and gyroscope. That kit contained a computer with an Intel processor, 4GB of RAM and 32GB of storage, with Ubuntu OS 16.04.3 LTS. It integrated deep learning environments into Intel hardware, optimizing their performance. The Robot Operating System (ROS) Kinetic distribution was present, facilitating the integration of the different modules, both I/O and the reinforcement learning environment. The experiments began with simulations and then executions in the real physical environment. Reward functions available in the DeepRacer environment were used. They took into account the robot position in relation to the line and the distance to obstacles on the line. Later, with better results, the research group's own functions were used, discretizing the action space and with the reward taking into account the time spent on the line. The actions were discrete: five angles and three speeds. The performance metrics were percentages of the route completed and times spent. As can be seen, in this approach the authors managed to combine reinforcement learning in a line-following robot for educational purposes. The results proved promising, although they were limited to the scope of five angles and three speeds.

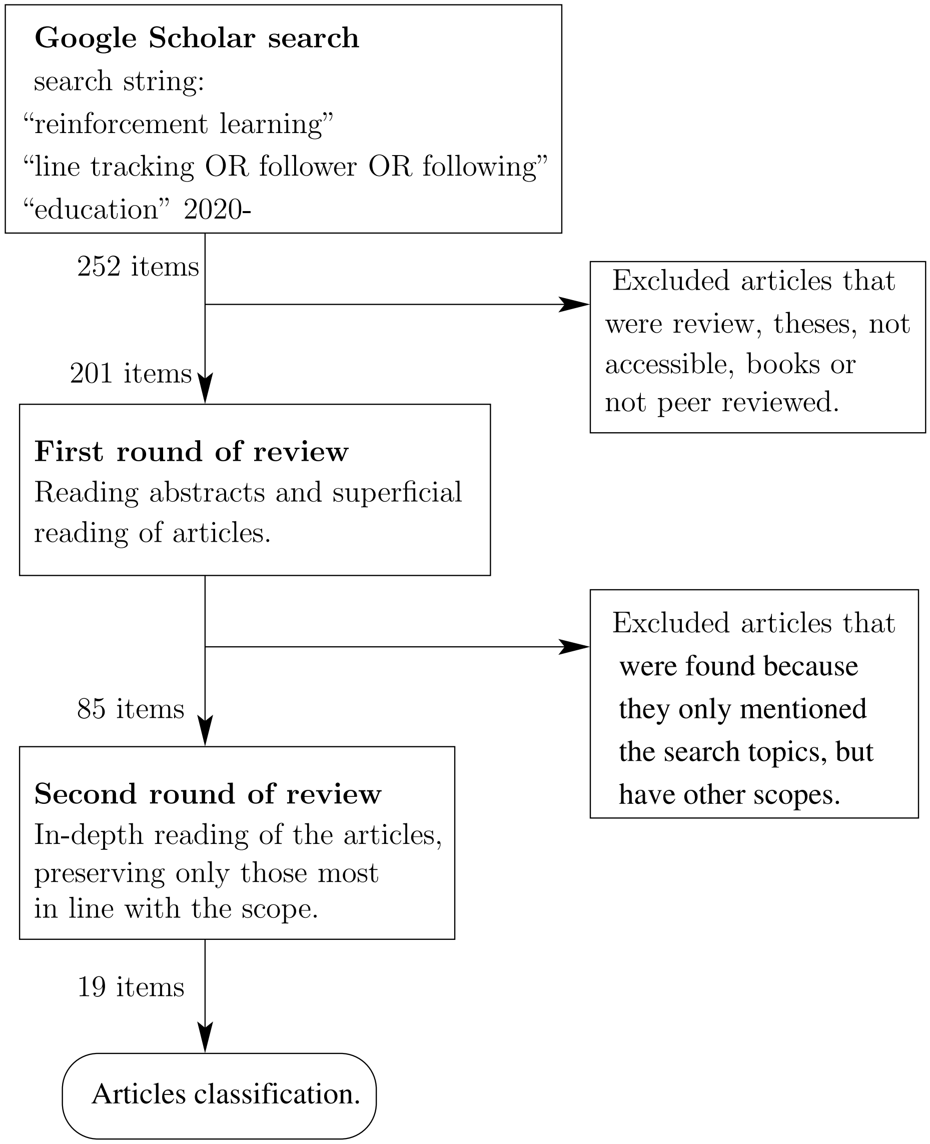


**Figure 1.** Article selection procedure, with quantitative results.

It is expected that by increasing these combinations, greater control over the robot will be obtained. On the other hand, the state space will increase significantly, which should be evaluated in terms of training time, computational cost and response time.

The next six articles utilize reinforcement learning in line-following robots but do not have an educational focus.

In order to mitigate practical limitations of training physical robots with reinforcement learning, such as high costs, long training durations and the risk of damaging the robot, the authors in [22] proposed a methodology based on the use of a digital twin. They developed a virtual replica of the physical robot Cozmo within the Unity simulation environment. Cozmo is equipped with a camera, a cliff sensor, an accelerometer and a WiFi module for communication with external devices. After training the model in simulation, the learned behavior was successfully transferred to the real-world line following robot. To assess the performance of the trained model, the agent was tested on a physical track designed for line following. The results highlight that the physical robot was able to follow the line using a model trained in the simulated environment. In addition, authors concluded that the digital twin framework provides a cost-effective and time-efficient alternative for reinforcement learning.

In reinforcement learning, a replay buffer is an element that stores and replays experiences that occurred during the interaction of the agent with its environment. Krawczyk et al [23] proposed an implementation of Q-learning without such a buffer to be used in non-stationary environments, being this the work main contribution. The experiments were performed in Gazebo simulations, using a cluster of 40 computers. The robot sensor was a camera. The reward was calculated as a function of the deviation from the line to be followed. The exploration level was adjusted to decay with the time.

Saadatmand et al [24] used the simulated annealing algorithm in Q-learning to control a line follower robot which used 32 infrared sensors and an ATMega64 microcontroller. The simulated annealing algorithm controlled how much exploration there would be and how much use would be made of what was already learned. The authors mathematically modeled the robot and created a simulator. After simulation, tests in real physical conditions were also performed. The speeds of two wheels were controlled. The state was defined by the measurements of the infrared sensors. At each execution step, an action was chosen probabilistically, with an increasing weight for the best learned action and decreasing for a random action. The reward took into account the misalignment in relation to the line to be followed and the speed. The performance metric took into account the ability to follow the line and the time spent in the trajectory.

Lee et al [25] proposed Deep Reinforcement Learning (DRL) as the basis for modulating the gain parameters of the PID controller with fuzzy control. In order to reduce the maximum overshot and shorten the tracking time compared to PID control systems that use DRL, the Wheeled Mobile Robot (WMR) used has an Atmega328-AU microcontroller of the Atmel AVR series as the control core, a TB6612 motor driver, and a set of 3 CNY70 circuit modules, each containing an infrared transmitter and a phototransistor. In this study, the Deterministic Policy Gradient (DPG) algorithm was adopted in the DRL to improve the input space and output space of the tracking motion control based on the Markov decision. As the author's objective, the difference between the target state and the current state of the WMR was decisive to obtain control. The experimental test used a straight-line simulation circuit to obtain the step response data and an S-curve tracking path, both created by black adhesive tape. In comparison with the adaptive changes, three metrics were employed in the experimental tests: fast response, steady-state error correction, and prevention of error judgment by artificial intuition. The objectives demonstrate that the DRL-Fuzzy-PID controller is adapted to correct the parameter values quickly when compared to the PID controller. The authors propose a more effective control based on the experimental results, as mentioned above. The tracking stability of the system using the DRL-Fuzzy-PID is improved by 15.2% compared to the conventional PID control, the maximum overshoot is reduced by 35.6%, and the tracking time ratio is reduced by 6.78%. If reinforcement learning is added, the convergence time of the WMR system is about 0.5 s and the accuracy rate reaches 95%. This control could become efficient in future research and needs improvement, according to its authors.

The paper [26] proposes the use of the Rotational Obstacle Avoidance Method (ROAM), an innovative approach to control nonlinear dynamics in robots when dealing with obstacles. ROAM adapts the initial dynamics to avoid star-shaped obstacles in spaces of arbitrary dimensions by rotating them towards the space. The control method used in the study is based on dynamical systems and vector fields (VFs), which allows the generation of a nonlinear control field evaluated in real time at the robot's position. This allows the robot to react instantly to disturbances and perceive changes in the environment. The algorithm used in reinforcement learning in the study is Deep Reinforcement Learning (DRL), which combines the use of neural networks with reinforcement learning. For testing purposes, Panda DoF robots were run using a combination of infrared sensors and photosensors. The simulation tools used included platforms such as VRT (Virtual Robot Toolkit) and ROS [27], a widely used platform for robotics software development. The open-source machine learning library TensorFlow was used to implement deep learning algorithms in addition to Scikit-learn, another Python language library. The trajectories are compared by observing the local minima, the distance to the desired limit cycle, and the similarity between the velocity after obstacle avoidance and the initial velocity. After the experiment was completed, the authors mentioned that the algorithm converged in an improved way in the tests, demonstrating similarity of movement compared to the applied metrics.

To investigate the Deep Deterministic Policy Gradient (DDPG) applied to control an industrial robotic arm, the authors in [28] proposed a tracker-line simulation. DDPG is a model-free algorithm which is basically a combination of the two algorithms: Deep Q-Network (DQN), and the Deterministic Policy Gradient (DPG). DQN is an adaptation of the Q-learning algorithm which uses a neural network. Focused on developing the learning responses to optimize the tracker-line, the simulated robot Kuka KR 6 R700 has been trained to follow 2D-trajectory custom on the Pybullet simulator. The simulation was implemented using the OpenAI Gym environment and Stable Baselines, a set of Python-based implementations of RL algorithms, were used as the neural network architectures were created with TensorFlow. To obtain these results, the authors' analyses were based on the training method of the RL algorithm, which provided performance metrics based on the change in reward in each episode, in addition to the prototype's ability to not deviate from the proposed trajectory. Additionally, the robot's deviation from the trajectory and the time required to complete a task were considered, and an observation of how the robot handled sharp turns while following the trajectory was included. As all reward functions are predictive, it was considered the robot's progress along the trajectory and the deviation from it. The authors mention the difficulty of adapting the control to situations involving real robots and propose validating the study as a future goal.

From the analysis of these last six studies that combine reinforcement learning and line following, it is observed that most works adopted simulated environments before transferring the learned behavior to physical robots. Some studies also explore combining reinforcement learning with other control strategies, such as fuzzy logic or PID, which could be explored, indicating potential areas for future work.

The focus on education appears in the next three papers, which also use reinforcement learning, such as in [29], that propose a framework characterized by the use of Proximal Policy Optimization (PPO) enhanced by Curriculum Learning (CL), within a virtual training and learning environment. These techniques are combined in order to improve the behavior of autonomous robots in the context of the IEEE Very Small Size Soccer (VSSS) competition. The proposed methodology creates representative scenarios of the competition in a simulated environment, including the behavior of external agents. The modeling starts with the Markov Decision Process (MDP) and integrates CL into different scenarios, allowing a faster and more efficient training process. The authors conclude that the RL-based framework surpassed the performance of an agent trained with manually designed heuristic rules. Furthermore, the agent trained using the CL-augmented technique achieved better performance than the one trained without CL. As a future research direction, the use of Self-Play techniques and/or Multi-Agent training is suggested, as these approaches hold potential to further improve learning efficiency and robustness. The authors point out that the existing literature involving Artificial Intelligence (AI) applied to VSSS competition was still limited at that time. Due to the scarcity of dedicated studies, the authors consider literature from RoboCup's Small Size League (SSL), which is a competition with similar game mechanics. This limitation indicated a broader research gap in the application of AI, including reinforcement learning, to autonomous robotic competitions. This gap was exploited by the researchers in this group who, a few years later, employed reinforcement learning in their competing robots [30]. Although they are not line followers, the robots act cooperatively to accomplish a task, which is to score a goal. Robot soccer competitions are also quite popular and encourage the use of sophisticated techniques in robots. In this case, a virtual learning framework for the IEEE Very Small Size Soccer (VSSS) competition based on Deep Reinforcement Learning (DRL) to train and improve the behavior of agents within the game strategy was proposed. The agent was trained in a scenario where a pair of robots must cooperate to score a goal on the opponent's field. To achieve this, the concept of Self-Play was used, where the agent learns to cooperate with a copy of itself. In addition, the Proximal Policy Optimization technique was used during the training process. The authors claim that this is the first work to successfully combine Self-Play and the PPO technique in IEEE VSSS.

In a similar way the authors in [31] focused on the evaluation of the effectiveness of using robots to evolve machine learning concepts and consolidate learning-to-learn skills on students. The study applies AlphAI robots, which are capable of line tracking and navigating around a designated arena. The robots are equipped with a wide-angle camera, an ultrasound sensor and infrared line tracking sensors. These components data are integrated with AlphAI software, allowing a graphical representation of the AI algorithm in use. Reinforcement learning is one of the techniques applied by the authors. The connection with AlphAI also provides a direct control interface for the robot. The training environment took place in an arena, designed to facilitate movement and behavior learning. Additionally, the Attention-Engagement-Error-Feedback-Reflection (AEER) pedagogical framework was implemented to guide the educational process. The authors conclude that, through analyzing the robots performance and training, students were able to identify the benefits and characteristics of machine learning.

One of these last three studies confirmed the superiority of applying reinforcement learning over heuristic methods in line-following robots, while also noting the scarcity of publications on the topic prior to 2020, which justifies focusing this review on works published from that year onward. Another interesting contribution found in these last works is the use of reinforcement learning with agents operating cooperatively, a promising approach in competitive scenarios that could be extended to future applications.

The low cost tracking robot and the machine learning techniques used in the platform presented in [32] are pointed out as a valuable tool for education but it does not employ reinforcement learning as in the previous cases. The authors proposed a low cost autonomous mobile robot with educational purposes, such as teaching and learning in the domain of machine learning, computer vision and intelligent automatic control. The Andruino robot is able to track and follow a determined path by using Android and Arduino in its software and control systems. The robot contains an integration with the Robot Operating System (ROS) and features a variety of sensors, including the usage of an external smartphone, such as ultrasonic sensors, gyroscope and accelerometer. The authors implemented a fuzzy logic system to prevent collisions, while also introducing a WiFi positioning system.

Focusing on younger students, the authors in [33] demonstrated that incorporating numerical simulators in the initial phase of robot development can boost participant motivation and prevent unnecessary expenditures and time wastage. This method was implemented with K-12 students who virtually explored robot construction during the COVID-19 pandemic before physically building them. In this way, the group of students who initially worked with the CoppeliaSIM simulation environment could later build the robots (sumo and line-follower), using the tools and learnings acquired in the previous experience. This resulted in greater autonomy for the students in relation to the availability of teaching kits, for example. Specifically regarding the line follower robot, it was implemented with three to five sensors positioned around the chassis to recognize the contrast between white and black and a PID control loop was configured. No artificial intelligence algorithm was employed.

In the same trend, for even younger children, during a week-long summer program, elementary school children assembled robot cars from kits including infrared and ultrasonic sensors [34]. The students then enhanced their robots using artificial intelligence and IoT techniques. An image processing model to control the robot is proposed, using block programming and, later, at a lower level, on the Arduino microcontroller. After these steps, students move on to a cloud-based AI platform (Google Teachable Machine) to train the robot with images of their hand gestures with the aim of making the cars make decisions based on the gestures learned, thus integrating all the knowledge from the different modules. In the final phase, students are free to change/increase functions in their robots, showing that they have mastered the topics covered. Although it is not a line-following robot, the use of AI was responsible for the robot's response, in the sense of making the decision to follow a certain route or not. Furthermore, the work was developed with children with the aim of awakening their interest in STEM areas.

The existence of many studies using line-following or similar robots in educational environments suggests that this approach enhances the engagement of the students with learning robotics and control algorithms, including reinforcement learning techniques. This can be attributed to the fact that such robots naturally serve as effective tools for playful and competitive activities, resembling sports-like challenges.

In a more distant context, the authors in [35] employed continuous action reinforcement learning in a two-wheeled robot with a single track. In addition to the coordinates, parameters such as center of mass, angles, speed and acceleration, wheel pressure, among other variables, make up the states. From these, actions are designed so that the actuators accelerate or brake the wheels. The rewards are determined from a model developed by the authors based on the evolution of the states, as well as the penalties. The control method was validated through simulation experiments performed in VehicleSIM (VS), from the BikeSim module of Matlab, and successfully performed the ramp jumping task of the two-wheeled robot with a single track. One year later, the same group proposes an improvement in the control strategy for single-track two-wheeled (STTW) robots operating in narrow and complex terrains [36]. The proposed approach integrates local path planning, trajectory tracking, and balance control into a unified framework. To train the controller, the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm was applied, using specifically defined state, action, and reward functions suited for narrow terrain navigation. The authors also demonstrate the flexibility of the framework by evaluating alternative reinforcement learning algorithms, including Double Deep Q-Network (DDQN), Deep Deterministic Policy Gradient (DDPG), and Soft Actor-Critic (SAC), all tested with consistent neural network architectures. Simulations were conducted using MATLAB/Simulink and BikeSim, validating the controller robustness and its ability to generalize across multiple terrain configurations. These characteristics highlight the control strategy potential for real-world scenarios requiring agility, precision, and adaptability. Although it is not a line-following robot, the trajectory implemented in the simulation could easily be adapted for this function.

Wang, Ling and Ma [37] propose a method and framework for controlling line and triangular vehicle formation for group formation such as convoys. A Linear Quadratic Regulator (LQR) controller based on the Ackermann model is developed for lateral distance control and a Proportional-Derivative (PD) controller for longitudinal distance control. To increase the effectiveness of the LQR controller, the Deep Deterministic Policy Gradient Derivative (DDPG) method is introduced into the control system. The DDPG networks are trained in a simulation environment and can subsequently predict LQR parameters in real-time experiments. These are not line followers but robots following other robots, which can be understood as a similar task. In addition to the simulations in the Gazebo 11 environment, two dedicated computers were used. The practical application of the method is also presented. The DDPG algorithm is implemented on the main control board, which is equipped with a Rockchip RK3588S processor. Meanwhile, the latitudinal distance controller (LQR algorithm) and the PD longitudinal distance controller operate on the base control unit, which is powered by an STM32F767IGT chip operating at a clock frequency of 216 MHz. The formation data is transmitted via a local wireless network facilitated by a WiFi6 module integrated into the main control board. The embedded formation control system is built on Robot Operating System 2 (ROS2), enabling the decoupling of intricate functionalities including communication, perception, and control. This method can be implemented on small vehicles that have limited computational resources and is also suitable for scenarios requiring dynamic motion control with higher tracking accuracy and stability. In this work, vehicles were controlled to effectively dislocate as a convoy. This kind of coordinated action differs from the one presented by [30] to soccer robot competitions. In the convoy study, the system elements are able to perform similar actions all over the operation, while in a robotic soccer team, individual robots may execute very different, yet complementary, tasks simultaneously, all aimed at achieving a common objective

In [38], the authors propose a method based on a deep reinforcement learning algorithm and convolutional network (DQN-CNN: Deep Q-Network with Convolutional Network) to enable autonomous navigation of robots equipped with RGB cameras. The training was carried out in the Coppelia VREP simulation environment while the real practical implementation was done on the NVIDIA Jetson Nano development board. For prototyping, a navigation track was used, where the robot is controlled by a DQN reinforcement learning algorithm. The authors claim that the results were effective, since the real robot was able to successfully execute the planned mission.

Finally, the next two papers explore line-following robots but without the application of machine learning and without an educational focus.

The authors in [39] used a robust, non-chattering sliding mode control (SMC) strategy for a line-following mobile robot equipped with infrared sensors capable of detecting both straight and curved paths, and an Arduino Mega is used as a microcontroller. Numerical and experimental trials were conducted to evaluate the effectiveness of the proposed method in simulation and real-world environments. For comparison purposes, the SMC and a conventional proportional-integral-derivative (PID) controller were implemented on the robot, enabling a comparative performance analysis. Results confirm the potential of non-chattering sliding mode control to ensure a satisfactory tracking performance under challenging operating conditions and in competitive scenarios. Although reinforcement learning was not applied in this study, it contributes with a new control strategy that can enhance performance in robotics competitions, particularly in scenarios demanding high precision and robustness. Therefore, the application of reinforcement learning, even when combined with other techniques, may not represent the definitive solution to the presented problems in this study. There should always be openness to exploring alternative approaches and methodologies.

A method presented in [40] enables mobile robots to perform obstacle avoidance and target tracking within dynamic indoor and outdoor environments. In the proposed method, image processing and machine learning approaches are considered. Since obstacles have differences in color and texture and to identify non-navigable areas, an onboard monocular camera is used to capture the road lanes by handling an image processing technique (Haar cascade). The position of the robot relative to the center of the road lane during navigation is calculated based on a set of fuzzy logic rules. The effectiveness and performance of the proposed method are tested in several simulation and experimental scenarios. Simulations were performed for different scenarios, i.e., straight and curved maneuvers, single and double lane changes in the presence of obstacles with different shapes and sizes. The experimental results demonstrate that the proposed method can detect the lane and obstacles during navigation. Deep learning techniques and the use of aerial cameras or GPS systems are presented by the authors as points of improvement in future work. Although not mentioned by the authors, reinforcement learning could certainly be employed in this application.

Table 1 summarizes these works contents.

Figure 2 is a word cloud graph obtained from the titles of the 19 articles that fit the inclusion criteria.

## 4. Conclusion

In this article, a literature review on the use of reinforcement learning in autonomous line-following robots was conducted. Nineteen articles were considered relevant, according to the inclusion and exclusion criteria, indicating that the area has potential for future research. The number of articles closely related to the search terms was small probably because the search criteria used were too narrow.

The research presents results with reinforcement learning algorithms, such as Q-Learning, Proximal Policy Optimization and Deep Q-Learning, that were successfully applied to solve line-following assignments. The choice of algorithms is conducted based on the complexity of environments, robots peripherals and performance requirements.

One of the main findings of the article is that combining reinforcement learning with other control strategies is a powerful strategy to improve the robot skills, for example stability, learning speed and adaptability [25, 32, 33, 37, 39, 40].

The adoption of simulation environments before, or even instead of, testing on real robots was an identified trend in the reviewed literature, with 16 out of 19 studies employing simulated backgrounds, which allowed researchers to safely test reinforcement learning algorithms, accelerate training, and experiment with complex scenarios that would be difficult or risky in physical setups. Many tools were cited, including ROS, Gazebo, TensorFlow, OpenAI Gym, CoppeliaSIM / V-REP, VehicleSim, BikeSim, and MATLAB/Simulink, identifying a diverse set of platforms adapted to simple and advanced line-following robot systems.

The studies reveal the benefits and drawbacks of using other approaches. Reinforcement learning enables robots to learn optimal navigation policies through trial and error interactions with the environment.

Another contribution is that reinforcement learning has an important educational impact as well. Many projects are implemented in university courses and robotic competitions [20, 29, 38, 31, 32], where students gain hands-on experience in programming, control systems and artificial intelligence concepts. Competitions stimulate creativity, problem solving skills and practical application of theoretical knowledge, highlighting the role of reinforcement learning applied to autonomous robots as an effective tool for teaching and learning. As a result, similar dynamics have also been conducted in basic-level courses with children and teenagers, seeking to increase their interest in STEM fields [34, 33].

Overall, reinforcement learning can be advised as a rising approach for autonomous line-following robots, combining technical innovation with educational and competitive benefits. Future work may focus on improving learning efficiency, generalization to unseen tracks, and integration with real-world robotic systems, while continuing to explore its potential in educational background.

## Author contributions

All authors contributed equally to the preparation of the article.

## References

[1] ABDULSAHEB, J. A.; KADHIM, D. J. Classical and heuristic approaches for mobile robot path planning: A survey. *Robotics*, MDPI, v. 12, n. 4, p. 93, 2023. ⟨https://doi.org/10.3390/robotics12040093⟩.

[2] ANTONYSHYN, L. et al. Multiple mobile robot task and motion planning: A survey. *ACM Computing*

**Table 1.** Summary of the works contents.

| work | [20] | [23] | [24] | [25] | [28] | [26] | [33] | [29] | [31] | [22] | [36] | [32] | [39] | [30] | [35] | [37] | [38] | [34] | [40] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| reinforcement learning | x | x | x | x | x | x | | x | x | x | x | | | x | x | x | x | | |
| education / competition | x | | | | | | x | x | x | | | x | | x | | | | x | |
| line follower | x | x | x | x | x | x | x | | | x | | x | x | | | | | | x |
| hardware | x | | x | x | | x | | | x | x | | x | x | x | | x | x | x | |
| simulation | x | x | x | x | x | x | x | x | | x | x | x | x | | x | x | x | | x |

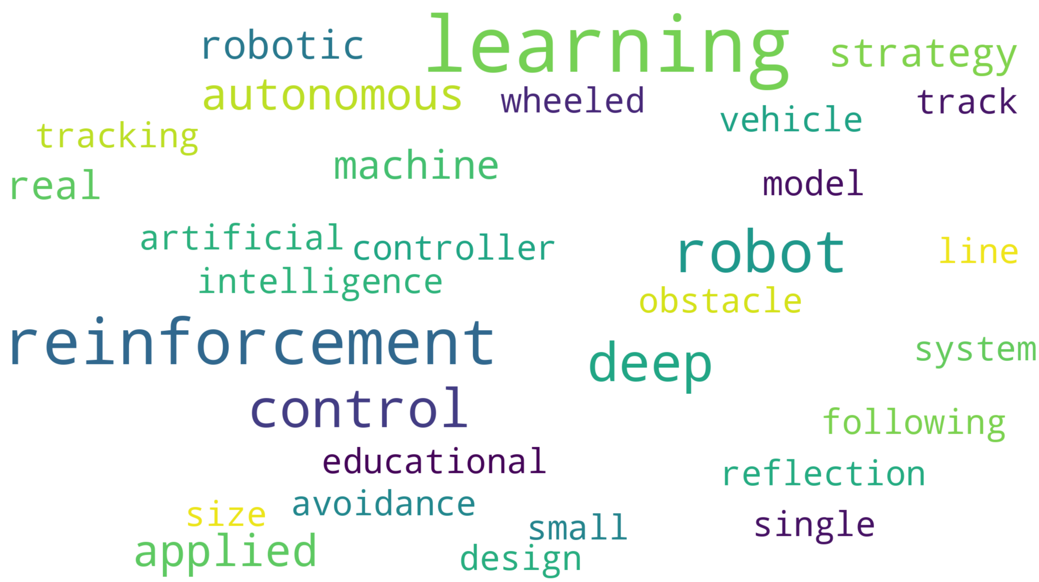


**Figure 2.** Word cloud obtained from the titles of the selected works.

*Surveys*, ACM New York, NY, v. 55, n. 10, p. 1–35, 2023. ⟨https://doi.org/10.1145/3564696⟩.

[3] KARUR, K. et al. A survey of path planning algorithms for mobile robots. *Vehicles*, MDPI, v. 3, n. 3, p. 448–468, 2021. ⟨https://doi.org/10.3390/vehicles3030027⟩.

[4] KABIR, M. A. et al. Design and implementation of automatic line follower robot for assistance of covid-19 patients. In: *Sustainable Communication Networks and Application: Proceedings of ICSCN 2021*. [S.l.]: Springer, 2022. p. 243–255. ⟨https://doi.org/10.1007/978-981-16-6605-6_17⟩.

[5] SAQIB, N.; YOUSUF, M. M. Design and implementation of shortest path line follower autonomous rover using decision making algorithms. In: IEEE. *2021 Asian Conference on Innovation in Technology (ASIANCON)*. [S.l.], 2021. p. 1–6. ⟨https://doi.org/10.1109/ASIANCON51346.2021.9544672⟩.

[6] OSWAL, S.; SARAVANAKUMAR, D. Line following robots on factory floors: Significance and simulation study using coppeliasim. In: IOP PUBLISHING. *IOP Conference Series: Materials Science and Engineering*. [S.l.], 2021. v. 1012, n. 1, p. 012008.

[7] IBM. *What is machine learning?* Available in: ⟨https://www.ibm.com/think/topics/machine-learning⟩. Accessed in March 12, 2025.

[8] NEXUS. *Speed and Precision: Follow-Line Robot Challenge at NSRC 2024*. 2024. Available in: ⟨https://news.utm.my/2024/12/speed-and-precision-follow-line-robot-challenge-at-nsrc-2024/⟩. Accessed in September 5, 2025.

[9] TECHNOXIAN. *Fastest Line Follower*. 2024. Available in: ⟨https://www.technoxian.com/fastest-line-follower⟩. Accessed in May 14, 2025.

[10] Harbour Edutech Pvt Ltd. *Fastest Line Follower Championship*. 2024. Available in: ⟨https://techradiance.in/line-follower-robot-competition/⟩. Accessed in May 14, 2025.

[11] UKMARS. *Line Follower*. 2020. Available in: ⟨https://ukmars.org/contests/line-follower/⟩. Accessed in May 14, 2025.

[12] Robotex International. *PCBWay LEGO Line Following*. 2025. Available in: ⟨https://robotex.international/lego-line-following/⟩. Accessed in May 14, 2025.

[13] Yelgea Event. *Line Follower*. 2025. Available in: ⟨http://www.robotchallenge.org.cn/competition-LineFollower.html⟩. Accessed in May 14, 2025.

[14] NASA. *Robotics Alliance Project*. 2025. Available in: ⟨https://robotics.nasa.gov/⟩. Accessed in May 14, 2025.

[15] OGATA, K. et al. *Modern control engineering*. [S.l.]: Prentice Hall, 2009.

[16] ERTEL, W. *Introduction to Artificial Intelligence, 2nd edition*. Switzerland: Springer International Publishing, 2017.

[17] CANESE, L. et al. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences*, v. 11, n. 11, 2021. ISSN 2076-3417. ⟨https://doi.org/10.3390/app11114948⟩.

[18] SHAKYA, A. K.; PILLAI, G.; CHAKRABARTY, S. Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, Elsevier, v. 231, p. 120495, 2023. ⟨https://doi.org/10.1016/j.eswa.2023.120495⟩.

[19] FRANÇOIS-LAVET, V. et al. An introduction to deep reinforcement learning. *Foundations and Trends® in Machine Learning*, Now Publishers, Inc., v. 11, n. 3-4, p. 219–354, 2018.

[20] COTA, J. L. et al. Roadmap for development of skills in artificial intelligence by means of a reinforcement learning model using a deepracer autonomous vehicle. In: *2022 IEEE Global Engineering Education Conference (EDUCON)*. [S.l.: s.n.], 2022. p. 1355–1364. ⟨https://doi.org/10.1109/EDUCON52537.2022.9766659⟩.

[21] TAN, F.; YAN, P.; GUAN, X. Deep reinforcement learning: From q-learning to deep q-learning. In: SPRINGER. *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14–18, 2017, Proceedings, Part IV 24*. [S.l.], 2017. p. 475–483. ⟨https://doi.org/10.1007/978-3-319-70093-9_50⟩.

[22] HASSEL, T.; HOFMANN, O. Reinforcement learning of robot behavior based on a digital twin. In: *ICPRAM*. [S.l.: s.n.], 2020. p. 381–386. ⟨https://doi.org/10.5220/0008880903810386⟩.

[23] KRAWCZYK, A. et al. Continual reinforcement learning without replay buffers. In: *2024 IEEE 12th International Conference on Intelligent Systems (IS)*. [S.l.: s.n.], 2024. p. 1–9. ⟨https://doi.org/10.1109/IS61756.2024.10705256⟩.

[24] SAADATMAND, S. et al. Autonomous control of a line follower robot using a q-learning controller. In: *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*. [S.l.: s.n.], 2020. p. 0556–0561. ⟨https://doi.org/10.1109/CCWC47524.2020.9031160⟩.

[25] LEE, C.-T.; SUNG, W.-T. Controller design of tracking wmr system based on deep reinforcement learning. *Electronics*, MDPI, v. 11, n. 6, p. 928, 2022. ⟨https://doi.org/10.3390/electronics11060928⟩.

[26] HUBER, L.; SLOTINE, J.-J.; BILLARD, A. Avoidance of concave obstacles through rotation of nonlinear dynamics. *IEEE Transactions on Robotics*, IEEE, v. 40, p. 1983–2002, 2023. ⟨https://doi.org/10.1109/TRO.2023.3344034⟩.

[27] Open Robotics. *ROS - Robot Operating System*. 2025. Available in: ⟨https://www.ros.org/⟩. Accessed in Aug. 16, 2025.

[28] JAFARI-TABRIZI, A.; GRUBER, D. Reinforcement-learning-based control of an industrial robotic arm for following a randomly-generated 2d-trajectory. In: . [S.l.: s.n.], 2021. p. 1–6. ⟨https://doi.org/10.1109/COINS51742.2021.9524158⟩.

[29] MEDEIROS, T. F. de; MáXIMO, M. R. O. de A.; YONEYAMA, T. Deep reinforcement learning applied to ieee very small size soccer strategy. In: *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*. [S.l.: s.n.], 2020. p. 1–6. ⟨https://doi.org/10.1109/LARS/SBR/WRE51543.2020.9306954⟩.

[30] MEDEIROS, T. F. de; YONEYAMA, T.; MÁXIMO, M. R. Deep reinforcement learning applied to ieee's very small size soccer in cooperative attack strategy. In: IEEE. *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*. [S.l.], 2023. p. 349–354. ⟨https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332923⟩.

[31] KONG, S.-C.; YANG, Y. Using the robot-assisted attention-engagement-error-feedback-reflection (aeer) pedagogical design to develop machine learning concepts and facilitate reflection on learning-to-learn skills: Evaluation of an empirical study in hong kong primary schools. In: *CSEDU (2)*. [S.l.: s.n.], 2024. p. 155–162. ⟨https://doi.org/10.5220/0012505700003693⟩.

[32] LOPEZ-RODRIGUEZ, F. M.; CUESTA, F. An android and arduino based low-cost educational robot with applied intelligent control and machine learning. *Applied Sciences*, v. 11, n. 1, 2021. ISSN 2076-3417. ⟨https://doi.org/10.3390/app11010048⟩.

[33] JOVENTINO, C. F. et al. A sim-to-real practical approach to teach robotics into k-12: A case study of simulators, educational and diy robotics in competition-based learning. *Journal of Intelligent & Robotic Systems*, Springer, v. 107, n. 1, p. 14, 2023. ⟨https://doi.org/10.1007/s10846-022-01790-2⟩.

[34] BUXTON, E.; JAVADI, E.; HAGAMAN, M. Foundations of autonomous vehicles: A curriculum model for developing competencies in artificial intelligence and the internet of things for grades 7–10. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2024. v. 38, n. 21, p. 23276–23284. ⟨https://doi.org/10.1609/aaai.v38i21.30375⟩.

[35] ZHENG, Q. et al. Continuous reinforcement learning based ramp jump control for single-track two-wheeled robots. *Transactions of the Institute of Measurement and Control*, SAGE Publications Sage UK: London, England, v. 44, n. 4, p. 892–904, 2022. ⟨https://doi.org/10.1177/01423312211037847⟩.

[36] ZHENG, Q. et al. Reinforcement learning-based control of single-track two-wheeled robots in narrow terrain. *Actuators*, v. 12, n. 3, 2023. ISSN 2076-0825. ⟨https://doi.org/10.3390/act12030109⟩.

[37] WANG, Z.; LING, Y.; MA, M. A deep learning optimized lqr method for enhanced formation control with embedded systems. *Engineering Research Express*, IOP Publishing, v. 6, n. 2, p. 025203, 2024. ⟨https://doi.org/10.1088/2631-8695/ad3c12⟩.

[38] BEZERRA, C. D. de S.; CARDOSO, I. H. L.; VIEIRA, F. H. T. Deep reinforcement learning with convolutional networks applied to autonomous navigation of real robots using virtual scenario training. In: IEEE. *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*. [S.l.], 2023. p. 302–307. ⟨https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332922⟩.

[39] YILDIZ, H. et al. Sliding mode control of a line following robot. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Springer, v. 42, n. 11, p. 561, 2020. ⟨https://doi.org/10.1007/s40430-020-02645-3⟩.

[40] SINGH, R.; BERA, T. K.; CHATTI, N. A real-time obstacle avoidance and path tracking strategy for a mobile robot using machine-learning and vision-based approach. *Simulation*, SAGE Publications Sage UK: London, England, v. 98, n. 9, p. 789–805, 2022. ⟨https://doi.org/10.1177/00375497221091592⟩.