

Special Topic: Embodied Intelligence

Embodied intelligence in RO/RO logistic terminal: autonomous intelligent transportation robot architecture

Yongkang XU^{1,2}, Runjiao BAO^{1,2}, Lin ZHANG^{1,2},
Junzheng WANG^{1,2} & Shoukun WANG^{1,2*}

¹*School of Automation, Beijing Institute of Technology, Beijing 100083, China*

²*National Key Lab of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100083, China*

Received 31 July 2024/Revised 1 February 2025/Accepted 11 March 2025/Published online 25 April 2025

Abstract In this paper, an autonomous intelligent transportation robot equipped with intelligent perception and autonomous planning modules is proposed to meet the requirements of roll-on/roll-off (RO/RO) logistics terminals for the autonomous transportation of various types of finished vehicles. Based on the robot operating system (ROS) framework and a modular design approach, the robot integrates several key components, including a visual perception module, an autonomous planning module, a motion control module, an execution module, and an energy module. First, an improved convex hull algorithm efficiently extracts point cloud data, enhancing the efficiency of point cloud processing and addressing the issue of excessive target data. Three different evaluation methods were employed to determine the optimal bounding box of the target vehicle and to obtain its pose state. Second, improved proximal policy optimization (PPO) and generalized advantage estimation (GAE) algorithms, which are utilized in this paper, were implemented to autonomously plan collision-free trajectories for robots, facilitating their movement from an initial position to the target transportation location, and subsequently to the final placement position. Finally, physical experiments confirmed that the robot effectively mimics the three fundamental components of humanoid functionality: perception, cognition, and action, thus enabling autonomous and intelligent handling of complex vehicle transportation tasks. These demonstrations validate the feasibility and effectiveness of the proposed robot.

Keywords embodied intelligence, transportation robot, RO/RO logistic terminal, detection, autonomous planning

Citation Xu Y K, Bao R J, Zhang L, et al. Embodied intelligence in RO/RO logistic terminal: autonomous intelligent transportation robot architecture. *Sci China Inf Sci*, 2025, 68(5): 150210, <https://doi.org/10.1007/s11432-024-4395-7>

1 Introduction

In recent years, the transportation of finished vehicles, particularly at roll-on/roll-off (RO/RO) logistics terminals, automobile logistics transfer bases, and automobile production plants, has predominantly relied on manual operation [1]. Specifically, when RO/RO ships dock at ports, transport personnel are required to utilize their driving skills to ensure accurate and precise parking of vehicles. These tasks require experienced operators with highly focused attention, leading to increased operational and labor costs [2]. Although autonomous unmanned systems, such as mobile transportation robots, have emerged as potential solutions for vehicle transportation tasks [3], their functionality remains limited. Current robots typically lack embodied intelligence, which refers to the capability of perceiving and adapting to environmental uncertainties, interpreting complex sensory data, and making context-aware decisions. Without these capabilities, transportation robots struggle to autonomously detect and correct errors, recognize objects in dynamic environments, and flexibly plan actions to ensure robust task execution [4]. Integrating principles of embodied intelligence into robotic design could effectively address these limitations, significantly enhancing their autonomy and reducing dependency on human operators within complex logistics environments.

Autonomous intelligent mobile transportation robots are increasingly being implemented and are significantly advancing warehousing logistics and factory transportation [5]. Highly valued by both academic

* Corresponding author (email: bitwsk@bit.edu.cn)

and industrial sectors, these robots are appreciated for their (i) compact size, robust flexibility, and capability to navigate confined spaces efficiently, (ii) diverse motion modes tailored to a variety of restrictive operational environments, (iii) ability to effectively collaborate with other robots. Five distinct robot types have been specifically developed for finished vehicle transportation and are predominantly utilized in indoor and outdoor parking facilities [6]. Among these are lifted robots, split robots, comb robots, and gripper robots, the latter operating by maneuvering beneath or above vehicles [7]. Most of these robots require external sensors installed in designated parking zones to facilitate accurate vehicle positioning and transportation. Furthermore, there exist robots capable of transporting vehicles along with pallets or brackets [8]. Traditional robotic control methods, lacking the necessary capabilities to enhance robotic intelligence and humanoid features, are increasingly inadequate to satisfy the evolving demands of the automotive logistics industry and emerging productive forces.

In intelligent logistics and supply chain management, embodied intelligent robots optimize transportation, warehousing, and distribution processes, enhancing logistics efficiency and service quality. Similarly, intelligent manufacturing, urban management, and environmental monitoring benefit from collaborative optimization using embodied intelligent robots, improving overall system efficiency and service levels. Recent studies have examined various autonomous robots designed for transporting finished vehicles, including automated valet parking robots, autonomous handling robots, and automated guided vehicle systems (AGVs) [9].

Automated valet parking robots typically require vehicle loading platforms for retrieving and placing vehicles, resulting in substantial hardware costs. Additionally, their application is limited to indoor parking areas, making them unsuitable for the outdoor conditions of port automotive logistics. For instance, the comb-tooth parking robot “GETA”, developed in [10], has a maximum speed of 1.5 m/s and a single pick-up duration of 120 s. The second category involves vehicle-arm-based systems, such as the RAY robot [11, 12]. The RAY robot employs adjustable front and rear arms to grip two front tires, accommodating different wheelbases. Stanley Robotics [13] developed an autonomous robot capable of outdoor positioning, vehicle retrieval, path planning, obstacle detection, and human-machine interaction, achieving speeds up to 4 m/s and pick-up times of approximately 60 s. The third category comprises multiple robot systems, such as iCART [14], iCART2 [15], and AVERT [16]. These robots achieve precise positioning, navigation, object recognition, and obstacle avoidance, enabling autonomous driving, parking space identification, and warehouse entry and exit. However, these robots require precise control, and their maximum driving speed is limited to 1.2 m/s.

In the embodied intelligence paradigm, an intelligent agent, considered a behavioral subject, employs a “perception-execution” mechanism in which embodied perception algorithms deliver comprehensive outputs to an execution module [17]. Actions performed by this module generate feedback, dynamically adjusting the model’s perceptual domain and establishing a closed-loop “perception-execution” process. Through environmental interaction, embodied algorithms dynamically optimize models, efficiently allocate computational resources, enhance perception accuracy, and iteratively refine intelligent algorithms via feedback [18].

In the execution phase of embodied intelligence, the agent must not only optimize paths but also integrate environmental perception and cognitive data to predict and mitigate risks. Current approaches primarily fall into model-based and learning-based planning methods [19]. Unlike traditional system identification or adaptive control methods for uncertain dynamics, machine learning methods frame trajectory planning, defined as finding optimal strategies, as reinforcement learning problems using Markov decision models, allowing agents to achieve more effective outcomes through continuous environmental interaction. Zhu et al. [20] employed reinforcement learning with deep neural networks as nonlinear approximators for visual navigation, where navigation targets are inputs to the model.

To address low sample utilization and non-standard hardware descriptions in transfer learning, two hardware encoding strategies, explicit (HCP-E) and implicit (HCP-I), were introduced by Chen et al. [21]. These strategies integrate encoding vectors, current states, and actions into strategy selection to enhance adaptability and efficiency. Luck et al. [22] applied the soft actor-critic (SAC) algorithm, combining deep reinforcement learning-derived Q-functions with reward evaluation, formulating a new objective function optimized via global particle swarm algorithms. Ha [23] presented a population-based policy gradient method enabling agents to simultaneously optimize both behavioral and morphological policies, surpassing traditional models limited to behavioral policies alone. Fager et al. [24] provided an integrated solution for intelligent mobile collaborative robots, combining mobile platforms, sensors, computational units, robotic arms, and comprehensive models. Zhao et al. [25] developed the AeroAgent

system, employing multimodal, large-scale models for drone control inspired by brain and cerebellum functions to enhance intelligence and stability.

These advancements aim to equip mobile robots with specialized intelligence for complex environment perception and autonomous decision-making. Despite significant progress in perception, decision-making, and human-language interactions enabled by large model technologies, most existing systems remain conceptual with limited practical implementation. Key challenges include inadequate comprehension of open scenarios, which rely on human-provided instructions. Continued advancements in embodied intelligence are expected to promote autonomous learning and evolution, driving comprehensive autonomy and facilitating industrial upgrades.

In this study, the objective is to efficiently accomplish transportation tasks while adhering to system robustness constraints. A novel vehicle type has been developed to operate effectively in challenging RO/RO logistics terminal environments while transporting heavy payload vehicles. The contributions of this research are summarized as follows. (i) A novel intelligent architecture for robots used in finished vehicle transportation has been proposed, incorporating perception, cognition, and autonomous action capabilities. This architecture significantly differs from traditional control methods by enabling robots to autonomously acquire information, analyze problems, and respond intelligently in complex and dynamic operational scenarios. (ii) A simulation method based on LiDAR point clouds has been developed to precisely detect the relative positions and outer boundaries of finished vehicles. The method enhances the robots' ability to identify targets accurately and has been experimentally validated, confirming its feasibility and accuracy in practical applications. (iii) An advanced planning algorithm integrating generalized advantage estimation (GAE) and proximal policy optimization (PPO) reinforcement learning was developed and implemented in the robot's controller. This algorithm effectively addresses the challenges associated with dynamic transportation environments, providing a robust approach for real-time decision-making. Physical experiments verified the algorithm's capability to manage complex automated tasks efficiently and reliably.

The content distribution of this work is as follows. Section 2 presents the overall framework and detailed design of the proposed robot system. Section 3 introduces the algorithms developed for target localization and autonomous planning. Section 4 describes the working scenarios and physical experiments conducted with transportation robots. Finally, Section 5 provides a summary of the findings and outlines recommendations for future research.

2 Design and framework of the transportation robot

2.1 Framework of embodied intelligence in transportation robots

In robotics, embodied artificial intelligence refers to systems capable of interacting with their environment and engaging in exploratory learning. These robots, integrated with advanced human-machine interfaces, execute tasks such as perception-based exploration, autonomous navigation, and task allocation through human-machine interaction commands. Within these operational environments, robots iteratively refine their task execution strategies through continuous physical interaction. Whether in simulated environments or real-world settings, the extensive data available for embodied robots provide a solid foundation for enhancing their learning capabilities in practical applications.

Embodied intelligence introduces a novel paradigm in transportation robot operations, transforming autonomous driving systems from passive observers into active participants within intelligent transportation. This interactive learning framework facilitates the acquisition of deeper abstract knowledge, addresses the challenge of scene transfer, and leverages the significant potential of implicit intelligence in RO/RO logistics terminals. Furthermore, by adapting small-scale datasets, urban autonomous driving agents can be effectively tailored to docking scenarios, promoting advancements in the autonomous operation of terminal transportation robot.

Implementing embodied intelligence in transportation robots requires integrating several essential modules, as depicted in Figure 1. Physical transportation robot operations are centrally managed by domain controllers, enabling seamless coordination. Interactions between robots and roadside equipment are facilitated via V2X modules, and remote control modules synchronize navigation tasks remotely. Effective decision-making by transportation robot critically depends on diverse data sources, including LiDAR, camera inputs, digital twins, scheduling systems, and control frameworks. Deployment of embodied

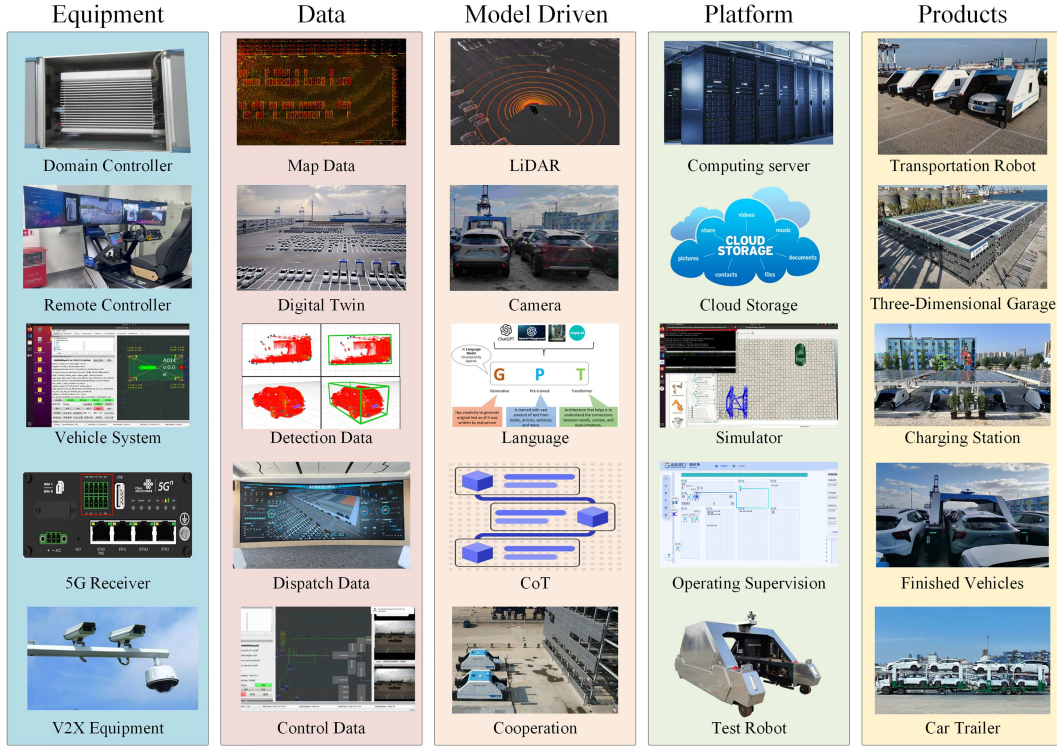


Figure 1 (Color online) Application of embodied intelligence in the field of robotics, autonomous driving, and smart transportation.

intelligence algorithms utilizes model-driven approaches, synthesizing multimodal data from visual and linguistic inputs and incorporating embodied question-answering capabilities. The introduction of chain-of-thought (CoT) mechanisms enhances inference and decision-making processes, establishing an end-to-end mapping from perception to control, thus ensuring logical coherence in planning and execution workflows. To satisfy computational and operational demands, the platform must support high-performance computing, extensive data storage, efficient algorithm training, realistic simulations, and comprehensive supervisory functionalities. Transitioning from theoretical models to practical implementation necessitates collaboration among transportation robots, three-dimensional garages, charging stations, finished vehicles, and car trailers, collectively advancing intelligent RO/RO logistics terminals.

The framework of the integrated robotic system is presented in Figure 2, comprising primarily an autonomous intelligent transportation robot, a cloud supervision platform, an intelligent connected cloud, intelligent security infrastructure, and intelligent networked roadside facilities. At its core, the autonomous transportation robot integrates sensors such as LiDAR and cameras to perceive the environment and autonomously perform tasks including docking and transport. These robots utilize embodied intelligence algorithms, which synthesize LiDAR and linguistic data to enable context-aware decision-making and control. The cloud supervision platform supports these robots by providing real-time monitoring, system diagnostics, and decision-making capabilities through the integration of robot-derived data and information from other system components, ensuring efficient operation and coordination. The intelligent connected cloud ensures seamless communication between robots and the cloud platform, facilitating low-latency, real-time data flow and adjustments throughout the system. Security infrastructure employing surveillance systems, secure communication protocols, and safety sensors on the robots safeguards the system from potential hazards. Additionally, intelligent roadside facilities including charging and docking stations support robot operations by providing essential services and real-time terminal condition updates. Through embodied intelligence integration, robots continuously adapt to environmental changes, optimizing operational performance and safety. This framework offers a scalable solution for autonomous logistics at RO/RO terminals, enhancing efficiency, safety, and sustainability.

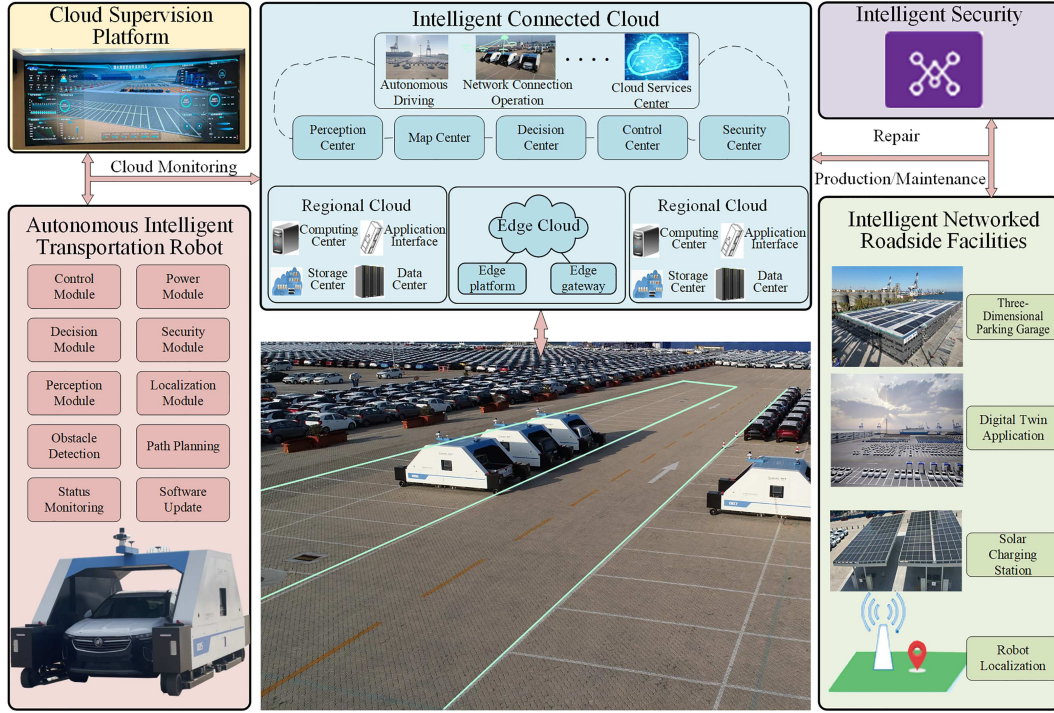


Figure 2 (Color online) Framework of the robot integrated system.

2.2 Physical implementation of embodied intelligence in transportation robots

In recent years, intelligent port construction has gained increasing attention. Embodied intelligence is anticipated to play a significant role in autonomous driving at RO/RO logistics terminals, offering novel solutions for vehicle transportation automation. Robots equipped with embedded artificial intelligence capabilities interact with and learn from their environments, significantly enhancing transportation performance. Figure 3(a) illustrates robots transporting finished vehicles at the RO/RO logistics terminal. As depicted in Figure 3(b), the vehicle transportation robot primarily consists of a control module, a perception and positioning module integrating multiple sensors, a power module, a lifting module, an independent motion and independent steering (IMIS) module, and a structural frame. The robot measures 6000 mm × 3000 mm × 2800 mm, with the control module packaged in dimensions of 360 mm × 360 mm × 200 mm. Constructed from aluminum alloy with an anodized outer surface, it meets an IPV67 waterproof rating and offers protection against salt spray corrosion. Mounted atop the robot is a two-degree-of-freedom servo turntable integrating sensors such as cameras, infrared cameras, LiDAR, radar, and IMU. The servo turntable expands sensor coverage through pitch and azimuth movements, facilitating localization, environmental perception, and target vehicle positioning.

This study introduces a novel mechanical arm module specifically designed for finished vehicle transportation robots, focusing on vehicle lifting capabilities. The module includes a stationary arm and a movable arm, engineered to accommodate vehicles of varying wheelbases. The mechanical arm addresses the challenge posed by vehicle diversity. Using 2D LiDAR scanning, only wheel contours are detected, with minimal depth variation between adjacent point clouds. Precise wheel coordinates are extracted from processed point cloud data by calculating their lengths and central points. To enhance adaptability, the robot features fixed and adjustable mechanical grippers on each side for lifting vehicles. A 2D LiDAR at each gripper's center identifies and positions target vehicle wheels.

An electronic system manages power distribution, interfaces, and main control loop operations. Encased within the robot frame's protective shell, the electronics are shielded from salt, alkali, and humidity common to RO/RO terminals. The electronic system provides power supply, Ethernet, serial port, and CAN connections for component integration and operation. Additionally, a 5G communication interface enables real-time command and data exchange with the cloud. The electronic system of the prototype is shown in Figure 4.

An eight-core CPU operating at 2.26 GHz within the robot operating system (ROS) environment

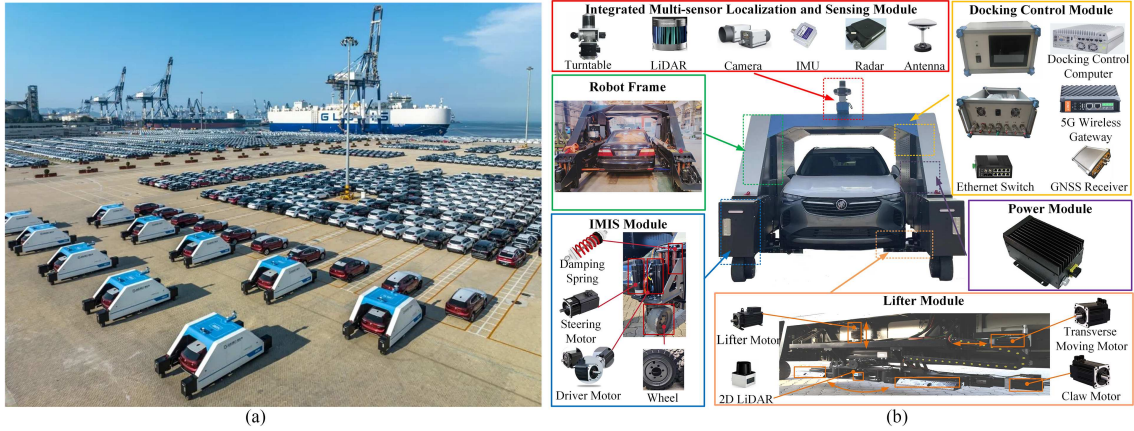


Figure 3 (Color online) System architecture for robot applications used in finished vehicle transportation. (a) Multiple integrated robots applied in RO/RO logistics terminals; (b) basic distribution of components in the robot.

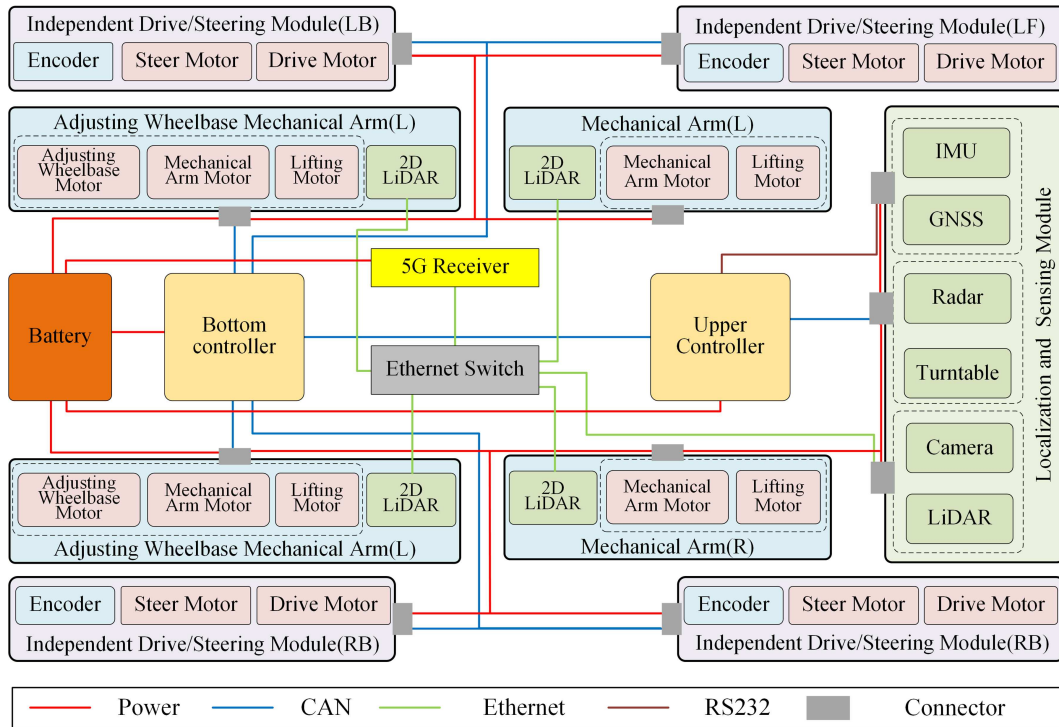


Figure 4 (Color online) Wiring diagram of the robot electronic system. External power is supplied through the bottom battery, distributing power to each electronic component. The Ethernet connects to the upper controller, 5G receiver, camera, and LiDAR, enabling remote robot control and task execution. Servo motors for the independent drive/steer and mechanical arm module are controlled using the CAN protocol, ensuring rapid response and synchronization.

handles remote communication, sensor data processing, and algorithm execution. Cameras, LiDAR, and 2D LiDAR modules connect to the main controller through Ethernet interfaces. A DC/DC converter supplies power from a 380 V source to the microcontroller board, IoT module, and network switch. These components are housed in a sealed electronics compartment. Detailed structural parameters are presented in Table 1.

Table 1 Specification of the finished vehicles transportation robot.

Description	Value	Unit
Robot length	6000	mm
Robot width	3000	mm
Robot height	2800	mm
Robot weight	6	T
Robot wheel diameter	343	mm
Robot wheel base	5438	mm
Robot tread	2570	mm
Maximum turning angular velocity	0.57	rad/s
Maximum acceleration	2.31	m/s ²
Maximum deceleration	1.16	m/s ²

3 Methodology

3.1 Detection of the finished vehicle

In cases involving single-frame point cloud data, the target point cloud often exhibits sparsity, making precise pose estimation challenging. Determining the minimum bounding box for LiDAR point clouds has become central to tasks involving 3D object detection and localization. Convex hull algorithms play a pivotal role in this context due to their efficiency in computing the smallest convex set containing all points, crucial for accurately approximating boundaries of irregular shapes typically observed in LiDAR datasets. Such algorithms establish a robust geometric foundation for constructing minimum bounding boxes, providing straightforward methods to define outermost point-cloud limits while minimizing computational complexity, particularly in dynamic, real-time applications. For instance, efficient convex hull algorithms in 3D space, as described in [26], offer effective boundary detection solutions for autonomous systems, enabling LiDAR data processing with low computational cost and high precision.

A primary advantage of employing convex hull algorithms for LiDAR point cloud bounding box calculations lies in their capability to efficiently handle large, noisy, and sparse datasets. Their computational efficiency makes them particularly suitable for real-time processing demands in autonomous systems and robotics. Furthermore, their inherent scalability allows them to manage extensive LiDAR datasets generated within complex environments. Specifically, Dyllong et al. [27] explored applying convex hull algorithms in octree-based object encoding, demonstrating robust and precise distance calculations. In [28], a heuristic algorithm for anchor placement is provided, which optimizes convex hull-based localization in 3D space.

Additionally, convex hull algorithms substantially contribute to improving vehicle pose estimation accuracy, especially under challenging environmental conditions. Using 3D LiDAR data, convex hulls offer straightforward yet effective representations of object boundaries, significantly aiding precise localization and path planning tasks. The iterative refinement methods of convex hulls, outlined by Garloff et al. [29], facilitate optimal minimum bounding box determination, thereby enhancing robustness in pose estimation for accurate navigation in dynamic urban settings. Moreover, Heng et al. [30] demonstrated the effectiveness of convex hull-based methods in accurately localizing vehicles, even within non-convex regions, further highlighting their applicability in precise 3D object tracking and localization necessary for autonomous navigation systems.

The stacked target point cloud set, defined as \mathbf{P}_{sum} , consists of extracted target point cloud data for each frame, represented as \mathbf{P}_i . The transformation matrix from the LiDAR coordinate system to the turntable coordinate system is defined as \mathbf{T}_L^T ; the matrix from the turntable coordinate system to the GNSS/IMU coordinate system is represented as \mathbf{T}_L^G ; and the transformation matrix from the vehicle coordinate system to the GNSS/IMU coordinate system is designated as \mathbf{T}_U^G . Based on these definitions, the following relationships can be derived:

$$\mathbf{P}_{\text{sum}} = \sum_{i=1}^n \mathbf{T}_U^{G-1} \mathbf{T}_L^G \mathbf{T}_L^T \mathbf{P}_i. \quad (1)$$

It is crucial to calculate the Y -direction deviation and azimuth angle of the target finished vehicle, whereas the Z -direction solely provides height information and its deviation is not significant. Consequently, the Z -direction is compressed, the point cloud is transformed into a two-dimensional plane, and

Algorithm 1 Convex hull point extraction algorithm.

```

1: function Convexhull( $P$ );
2:    $p_0 \leftarrow \text{lowest}(P)$ ;
3:    $P \leftarrow \text{sort}(p_0, P)$ ;
4:    $m \leftarrow 1$ ;
5:   for  $i \leftarrow 1$  to  $n$  do
6:     while  $i < n - 1$  and  $\text{ccw}(p_0, p_i, p_{i+1}) = 0$  do
7:        $i \leftarrow i + 1$ ;
8:     end while
9:      $p_m \leftarrow p_i$ ;
10:  end for
11:  Push  $p_0, p_1, p_2$  to stack
12:  for  $p_0, p_1, p_2$  for  $i \leftarrow 3$  to  $m$  do
13:    while  $\text{stack.size} \geq 2$  and  $\text{ccp}(\text{stack.next}_{\text{top}}, \text{stack.top}, p_i) \leq 0$  do
14:      Pop stack;
15:    end while
16:    Push  $p_i$  to stack;
17:  end for
18: end function
    
```

the point cloud is subsequently downsampled by calculating the convex hull. The convex hull point, as described in Algorithm 1, denoted as $\text{CH}(P)$, is defined by the following conditions:

$$\text{CH}(P) = \left\{ \sum_{i=1}^n \alpha_i p_i \mid p_i \in P, \sum_{i=1}^n \alpha_i = 1, \alpha_i \in [0, 1] \right\}, \quad (2)$$

where $P = \{p_i(x_i, y_i) \in \mathbb{R}^2 \mid i = 1, 2, \dots, n\}$ is represented as a set of two-dimensional points, which lie on the same plane. Given a set of two-dimensional point sets P , consideration is given to dividing it into two disjoint sets, S and Q :

$$\begin{aligned} & \underset{\theta, c_1, c_2}{\text{minimize}} \sum_{i \in S} (x_i \cos \theta + y_i \sin \theta - c_1)^2 + \sum_{i \in Q} (-x_i \sin \theta + y_i \cos \theta - c_2)^2, \\ & \text{subject to} \quad S \cup Q = P \ \& \ S \cap Q = \emptyset, \\ & \quad c_1, c_2 \in \mathbb{R}, 0^\circ \leq \theta < 90^\circ. \end{aligned} \quad (3)$$

For each point p_i in the target point cloud, its coordinates, which are (x_i, y_i) in the coordinate system XOY , are represented as (x_i^θ, y_i^θ) in the coordinate system $X_\theta O Y_\theta$,

$$\begin{bmatrix} x_i^\theta \\ y_i^\theta \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \quad (4)$$

$$\begin{aligned} c_x^{\max} &= \max \{C_x\}, c_x^{\min} = \min \{C_x\}, \\ c_y^{\max} &= \max \{C_y\}, c_y^{\min} = \min \{C_y\}, \end{aligned} \quad (5)$$

where $C_x = x_i$, and $C_y = y_i$, the angle θ , which is between the longitudinal direction of the minimum bounding box and the XOY coordinate system, is considered. Thus, the optimal size of the bounding box, which is determined through a series of computational steps, can be calculated as

$$d_x = c_x^{\max} - c_x^{\min}, d_y = c_y^{\max} - c_y^{\min}. \quad (6)$$

The equation for the straight line, which passes through this point in the direction of θ and is perpendicular to that direction, can be derived as follows:

$$L_1 : x \cos \theta + y \sin \theta - 0.5 (c_x^{\max} + c_x^{\min}) = 0, \quad (7)$$

$$L_2 : -x \sin \theta + y \cos \theta - 0.5 (c_y^{\max} + c_y^{\min}) = 0. \quad (8)$$

The coordinate values, which are derived according to the intersection formula of two straight lines, can be obtained as follows:

$$\begin{bmatrix} c_x \\ c_y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} 0.5 (c_x^{\max} + c_x^{\min}) \\ 0.5 (c_y^{\max} + c_y^{\min}) \end{bmatrix}. \quad (9)$$

Algorithm 2 Minimum bounding box of the nearest neighbor criterion.

```

1: function Boundingbox( $P$ );
2:    $M \in \mathbb{R}^{n \times 2}$ ;
3:   for  $\theta = 0$  to  $\pi/2$  step  $\delta$  do
4:      $e_1 \leftarrow (\cos\theta, \sin\theta)$ ,  $e_2 \leftarrow (-\sin\theta, \cos\theta)$ ;
5:      $C_x \leftarrow M \cdot e_1^T$ ,  $C_y \leftarrow M \cdot e_2^T$ ;
6:      $q \leftarrow \text{CalculatecriterionX}(C_x, C_y)$ ;
7:     Insert  $q$  into  $Q$  with key( $\theta$ );
8:   end for
9:   Select key  $\theta^*$  from  $Q$  with max value;
10:   $C_x^* \leftarrow M \cdot (\cos\theta^*, \sin\theta^*)^T$ ,  $C_y^* \leftarrow M \cdot (-\sin\theta^*, \cos\theta^*)^T$ ;
11:   $a_1 \leftarrow \cos\theta^*$ ,  $b_1 \leftarrow \sin\theta^*$ ,  $c_1 \leftarrow \min C_x^*$ ,  $a_2 \leftarrow -\sin\theta^*$ ,  $b_2 \leftarrow \cos\theta^*$ ,  $c_2 \leftarrow \min C_y^*$ ;
12:   $a_3 \leftarrow \cos\theta^*$ ,  $b_3 \leftarrow \sin\theta^*$ ,  $c_3 \leftarrow \max C_x^*$ ,  $a_4 \leftarrow -\sin\theta^*$ ,  $b_4 \leftarrow \sin\theta^*$ ,  $c_4 \leftarrow \max C_y^*$ ;
13:   $c_x^{\max} \leftarrow \max\{C_x\}$ ,  $c_x^{\min} \leftarrow \min\{C_x\}$ ,  $c_y^{\max} \leftarrow \max\{C_y\}$ ,  $c_y^{\min} \leftarrow \min\{C_y\}$ ;
14:   $\lambda \leftarrow -(c_x^{\max} - c_x^{\min}) \cdot (c_y^{\max} - c_y^{\min})$ ;
15:  return  $\lambda$ ;
16: end function

```

Assuming that the shape of the vehicle resembles a rectangle, as described in Algorithm 2, the proximity of these point clouds to the two vertical edges should be concentrated on.

$$D_x = \min(|c_x^{\max} - C_x|, |c_x^{\min} - C_x|), \quad D_y = \min(|c_{\max_y} - C_y|, |c_{\min_y} - C_y|), \quad (10)$$

where D_x and D_y denote the distances from a point to two boundaries, with the boundary that has the closer distance range being selected from the two. The evaluation value function is defined as

$$d = \max\{\min\{D_x, D_y\}, d_0\}, \quad v_{\text{Closeness}} = -\sum_{i=1}^n \frac{1}{d}, \quad (11)$$

where d is the distance from the point to its nearest edge. Simultaneously, a minimum distance threshold, d_0 , was designed to avoid the division by zero of points on the boundary and the significant weight that is generated by points that are very close to the edge.

3.2 Robot autonomous planning based on reinforcement learning

Embodied intelligence, which emphasizes the integration of physical structure, sensorimotor capabilities, and adaptive control, has become a crucial paradigm in robotics. By enabling robots to seamlessly interact with their environment, this concept provides the foundation for autonomous systems to achieve real-world adaptability and efficiency. In the context of RO/RO terminals, where operational efficiency is critical to global maritime logistics, the application of embodied intelligence has the potential to revolutionize transportation robot performance. Specifically, it facilitates the optimization of motion trajectories, which is central to improving coordination and adaptability in dynamic and constrained environments.

To address the challenges associated with motion planning in RO/RO terminals, an integrated real-time trajectory optimization framework is introduced in this paper. This framework, designed to generate optimal motion trajectories, employs an expanded utilization of PPO combined with deep reinforcement learning (DRL) to approximate optimal solutions [31]. The robots and their operational environment were modeled using a Markov decision process, which captures the dynamic interactions essential for embodied intelligence. A reward function was developed to account for factors such as trajectory smoothness, time efficiency, collision avoidance, and energy consumption, ensuring that the robot's actions align with both computational and physical constraints. GAE was incorporated into the learning process to balance bias and variance, enabling the framework to converge more effectively toward optimal policies [32]. The proposed GAE combines Monte Carlo estimation with the time difference (TD) method to provide a more accurate estimation of the dominance function, thereby reducing variance without significantly increasing bias. This is particularly beneficial in the context of transportation robot navigation, as high variance in gradient estimation may lead to unstable learning and suboptimal control strategies. Using GAE, robots can achieve smoother and more reliable convergence during the training process, which is crucial for practical deployment in complex environments such as hybrid traffic scenes or signaled intersections, where the decision-making process is influenced by many dynamic and unpredictable factors [33]. By dynamically adjusting network parameters, the stability of trajectory generation and policy learning was ensured, thereby enhancing the robot's capacity to perform consistently in complex scenarios [34].

This study demonstrates that the integration of embodied intelligence principles into trajectory optimization not only improves the efficiency of RO/RO terminal operations but also establishes a scalable and generalizable approach for autonomous robots in logistics and beyond. By bridging theoretical advancements with practical applications, this research underscores the transformative role of embodied intelligence in advancing autonomous systems.

The motion dynamics of a robot is governed by its independent wheel control and steering mechanisms. The vehicle state at time t is defined as

$$s_t = \begin{bmatrix} x_t & y_t & \theta_t & v_t & \omega_t \end{bmatrix}^T, \quad (12)$$

where x_t, y_t denote the global coordinates, θ_t is the heading angle, v_t is the longitudinal velocity, and ω_t represents the angular velocity.

The control inputs are encapsulated in the action vector:

$$a_t = \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix}^T, \quad (13)$$

where τ_i and ϕ_i represent the torque and steering angle of wheel i , respectively.

The evolution of the vehicle's state can be described by the following nonlinear dynamic equations:

$$\begin{aligned} \dot{x} &= v \cos \theta, & \dot{y} &= v \sin \theta, & \dot{\theta} &= \omega, \\ \dot{v} &= \frac{1}{m} \left(\sum_{i=1}^4 f_{\tau}(\tau_i) - f_{\text{drag}}(v) \right), \\ \dot{\omega} &= \frac{1}{I} \sum_{i=1}^4 f_{\phi}(\phi_i), \end{aligned} \quad (14)$$

where m is the vehicle mass, I is its moment of inertia, and $f_{\text{drag}}(v) = c_{\text{drag}} v^2$ is the velocity-dependent aerodynamic drag.

The control objective is to minimize both trajectory tracking errors and control effort. The cumulative reward, expressed as a function of the policy π_{θ} , is given by

$$J(\pi_{\theta}) = \mathbb{E}_{\pi_{\theta}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \quad (15)$$

where $\gamma \in (0, 1)$ is the discount factor. The instantaneous reward $r(s_t, a_t)$ balances tracking accuracy and control smoothness:

$$r(s_t, a_t) = - \left(\alpha_{\text{track}} e_{\text{track}}(s_t) + \alpha_{\text{control}} \|a_t\|_2^2 \right), \quad (16)$$

where $\text{track}(s_t) = \sqrt{(x_t - x_{\text{ref}})^2 + (y_t - y_{\text{ref}})^2}$ measures the Euclidean distance to the reference trajectory. $\|a_t\|_2^2$ penalizes excessive control effort, ensuring smooth operation. This objective underscores the importance of balancing precise trajectory tracking and energy-efficient control.

Estimating the advantage function $A(s_t, a_t)$ accurately and efficiently is crucial for policy optimization. While simple advantage estimators like TD(0) or Monte Carlo returns are widely used, they suffer from high variance or bias. GAE addresses this trade-off by combining multi-step TD returns with a tunable decay factor.

The value function $V(s_t)$ represents the expected cumulative reward starting from state s_t :

$$V(s_t) = \mathbb{E}_{\pi_{\theta}} \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t \right]. \quad (17)$$

The advantage function quantifies the relative benefit of a specific action compared to the baseline value:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t), \quad (18)$$

where $Q(s_t, a_t)$ is the action-value function.

The one-step TD residual serves as a basic advantage approximation:

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t). \quad (19)$$

To mitigate the variance in single-step TD, GAE introduces a weighted sum of multi-step TD residuals:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l}, \quad (20)$$

where $\lambda \in [0, 1]$ is a tunable parameter controlling the trade-off between bias and variance. For computational efficiency, GAE is implemented recursively:

$$\hat{A}_t = \delta_t + \gamma \lambda \hat{A}_{t+1}. \quad (21)$$

This recursive formulation effectively balances short-term and long-term reward contributions, enhancing the stability of advantage estimation.

PPO optimizes the policy by maximizing a surrogate objective while constraining policy updates. The clipped objective is

$$L^{\text{clip}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (22)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the importance sampling ratio. The clipping operation ensures that $r_t(\theta)$ does not deviate significantly from 1, preventing overly large policy updates.

The PPO loss includes three components:

$$L^{\text{policy}}(\theta) = -L^{\text{clip}}(\theta), \quad (23)$$

$$L^{\text{value}}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t) - \hat{R}_t \right)^2 \right], \quad (24)$$

$$L^{\text{entropy}}(\theta) = -\mathbb{E}_t \left[\sum_a \pi_\theta(a|s_t) \log \pi_\theta(a|s_t) \right], \quad (25)$$

$$L(\theta, \phi) = L^{\text{policy}}(\theta) + c_1 L^{\text{value}}(\phi) - c_2 L^{\text{entropy}}(\theta). \quad (26)$$

As described in Algorithm 3, in each iteration, the DRL network obtains states, generates actions based on the state and policy, and executes them to receive immediate rewards and the next state value. After collecting a sufficient amount of data, a trajectory dataset is generated, and GAE is used to calculate the advantage function. This information is then utilized to update the parameters of the policy and value function networks. Once the network parameters approach convergence, given the starting point and endpoint for the robot, this method is capable of generating the time-optimal trajectory.

4 Experimental verification

The operational process of the autonomous transportation robot consists of two distinct phases: the approach phase, encompassing initial navigation, and the docking phase, involving precise docking and vehicle retrieval. High-precision autonomous navigation, even under conditions with intermittent or unavailable GNSS signals, is achieved through the integration and fusion of SLAM, IMU, and GNSS data. The robot, designed specifically for transporting finished vehicles within RO/RO logistics terminals, is illustrated in Figures 5(a) and (c). Additionally, the robot is equipped with cameras and LiDAR sensors to avoid collisions with other vehicles, robots, or obstacles.

During the approach phase, as depicted in Figures 5(d) and (e), the robot utilizes a torque distribution algorithm to navigate autonomously on main roads. The docking phase is further subdivided into two primary tasks: pick-up docking and release docking. In the pick-up docking task, the target vehicle's pose information is acquired through multi-sensor fusion, facilitating docking path planning and task execution. Upon reaching the target location, the robot stops, and four grippers descend simultaneously. The front gripper, equipped with a 2D LiDAR sensor, precisely aligns with the center of the vehicle's front wheel, while the rear gripper adjusts its clamp spacing according to the rear wheel's center detected by its LiDAR. Subsequently, the four grippers lift the vehicle, completing the retrieval process.

Algorithm 3 PPO with GAE.

Require: Initial policy parameters θ , value function parameters ϕ , discount factor γ , GAE decay factor λ , PPO clipping threshold ϵ , learning rates η_θ, η_ϕ , vehicle dynamics model $f_{\text{dyn}}(s_t, a_t)$, and maximum iterations N ;

- 1: **Initialization:** Initialize θ, ϕ randomly. Set episode counter $k = 0$;
- 2: According to the policy $\pi(\theta_k)$, collect set of trajectories \mathcal{D}_k in the environment;
- 3: **Repeat until convergence:**
- 4: **Collect trajectory samples:**
- 5: Reset environment to initial state s_0 ;
- 6: **For** timestep t **do**
- 7: Sample action $a_t \sim \pi_\theta(a | s_t)$;
- 8: Execute action $s_{t+1} = f_{\text{dyn}}(s_t, a_t)$;
- 9: Record (s_t, a_t, r_t, s_{t+1}) ;
- 10: **End**
- 11: **Compute GAE advantage estimates:**
- 12: **For** timestep t **do**
- 13: Compute TD residual $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$;
- 14: Compute advantage recursively $\hat{A}_t = \delta_t + \gamma \lambda \hat{A}_{t+1}$;
- 15: **End**
- 16: Compute cumulative returns $\hat{R}_t = \hat{A}_t + V_\phi(s_t)$;
- 17: **Update policy using PPO loss:**
- 18: Compute probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}$;
- 19: Define clipped objective $L^{\text{clip}}(\theta) = \min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t)$;
- 20: $\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathbb{E}_t[L^{\text{clip}}(\theta)]$;
- 21: **Update value function using MSE loss:**
- 22: Compute value function loss $L^{\text{value}}(\phi) = \mathbb{E}_t[(V_\phi(s_t) - \hat{R}_t)^2]$;
- 23: Update value function $\phi \leftarrow \phi - \eta_\phi \nabla_\phi L^{\text{value}}(\phi)$;
- 24: **Increment episode counter:** $k \leftarrow k + 1$;
- 25: **Return final policy:** $k\theta^*$;



Figure 5 (Color online) Working process of finished vehicle transportation robots. (a) The transportation robots pick up finished vehicles; (b) the transportation robot places the finished vehicles in the parking space; (c) the transportation robot uses LiDAR to recognize the pose of finished vehicles; (d) the bird's eye view of the RO/RO logistics terminal; (e) the transportation robot moves to the stereo garage; (f) the transportation robot places the finished vehicle in the connection area of the stereo garage.

For the vehicle release docking phase, the target area includes either the yard or the multi-level garage docking region, shown in Figure 5(b). A virtual docking target is generated according to the vehicle release task, guiding the robot to maneuver precisely over this reference point. After positioning, the grippers descend and release the vehicle, finalizing the release task. Within the yard area, vehicles must maintain a minimum separation distance of 30 cm. If the robot's battery level falls below a preset threshold, it autonomously proceeds to the charging station upon task completion. Figure 5(f) shows mobile transportation robots stationed at the RO/RO logistics terminal in Yantai, Shandong, China.

In initial single-frame point clouds, only partial external features of the target vehicle can be captured, presenting challenges for high-precision positioning. However, by sufficiently accumulating point cloud data across multiple frames, most external features of the target vehicle become identifiable. A convex hull is then calculated to extract the target vehicle's convex hull point cloud, significantly reducing the number of points required for subsequent computations. Finally, an optimal bounding box is computed

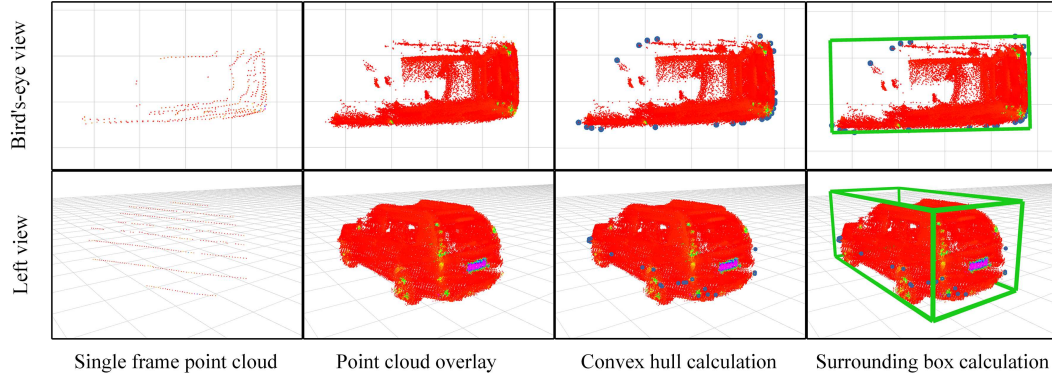


Figure 6 (Color online) Visualization of the LiDAR point cloud and the resulting optimal bounding box for target vehicle localization.

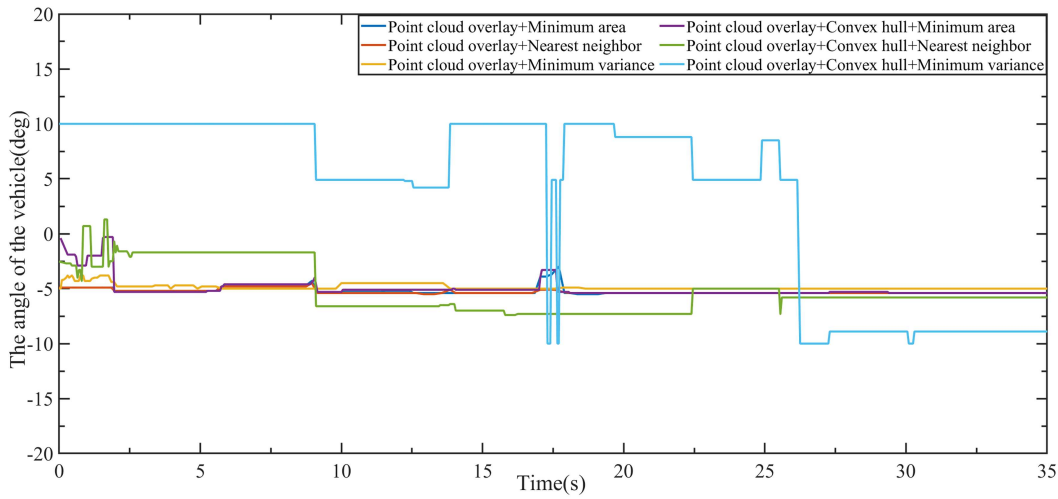


Figure 7 (Color online) Orientation angle results of lateral positioning of target finished vehicle parking position.

from the convex hull point cloud to obtain accurate parking pose data for the target vehicle, visualized as the green box in Figure 6.

Analyzing various algorithmic outputs as the robot approaches the target vehicle enables a detailed evaluation of their effectiveness, as shown in Figures 7 and 8. Additionally, the search processes of three evaluation criteria, minimum area, nearest neighbor, and minimum variance, are depicted in Figure 9. Among these, the minimum area and nearest neighbor criteria consistently demonstrate high performance, whereas the minimum variance criterion depends heavily on extensive statistical data. Since convex hull calculations typically retain only 15–30 points, the minimum variance criterion loses accuracy, making it unsuitable for these algorithmic processes.

The experimental results, presented in Table 2, demonstrate the performance of different algorithms for target vehicle localization and orientation. Using the minimum area, point cloud superposition, and convex hull algorithm, the target vehicle was successfully located and oriented with an angular error of 1.4° , a positional error of 1.2 cm, and a processing time of 11.37 ms. In contrast, the minimum variance approach, coupled with point cloud superposition and the convex hull algorithm, yielded a detection angle of 1.3° , a position error of 1.7 cm, and a detection time of 12.51 ms. Finally, the minimum edge method, integrated with point cloud superposition and convex hull processing, produced the most precise results, achieving a detection angle of 0.8° , a position error of only 0.9 cm, and an average detection time of 11.89 ms. These results highlight the comparative advantages of the different algorithms in terms of detection accuracy and processing efficiency, illustrating their potential for real-time vehicle localization applications. Overall, the nearest neighbor criterion has the best comprehensive performance.

To verify the practical applicability of the model trained in a simulated environment, it was tested in an authentic operational scenario involving a real autonomous intelligent transportation robot. The experimental environment closely mirrored the simulation conditions in terms of scene layout and param-

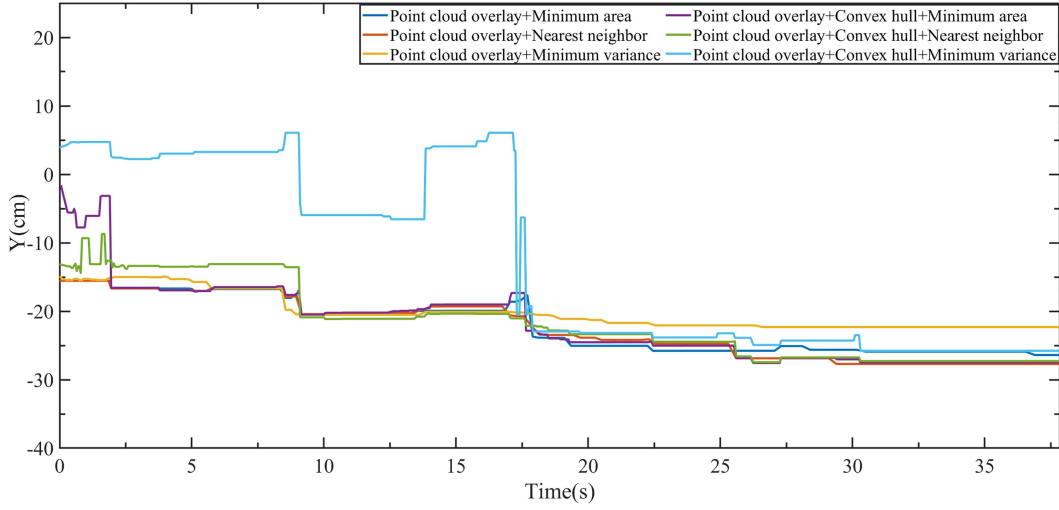


Figure 8 (Color online) Lateral deviation results in the localization of the target vehicle's parking position.

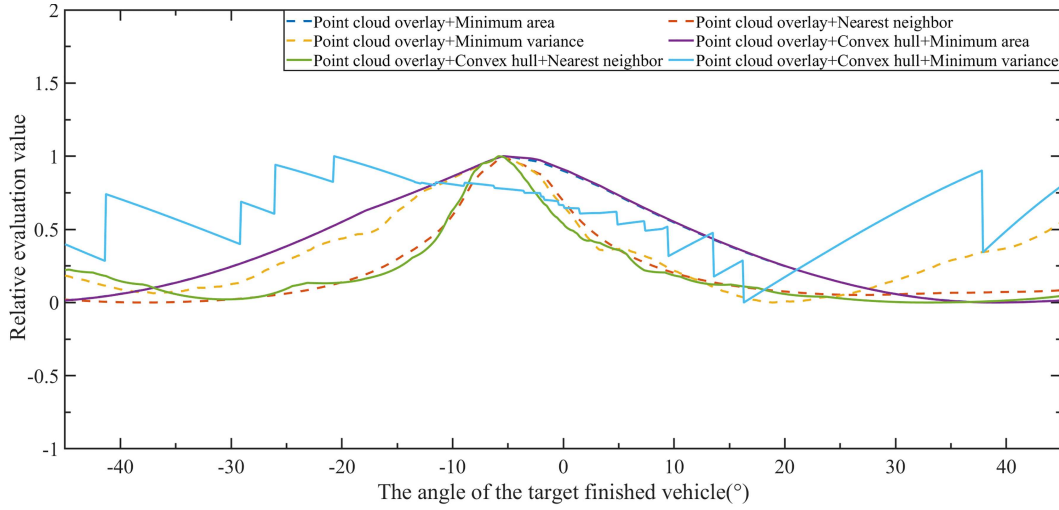


Figure 9 (Color online) Comparison of algorithmic evaluation values among different criteria for target vehicle localization.

Table 2 Performance comparison of different target vehicle localization algorithms. The values in bold represent the best performance achieved through the combination of point cloud overlay and convex hull-based optimization.

Rule	Point cloud overlay	Convex hull	Angle (°)	Position (cm)	Processing time (ms)
Minimum area	—	—	3.9	16.0	9.97
Minimum area	—	✓	3.9	16.0	1.08
Minimum area	✓	—	1.4	1.2	933.46
Minimum area	✓	✓	1.4	1.2	11.37
Minimum variance	—	—	8.4	14.7	15.07
Minimum variance	—	✓	8.9	16.1	1.95
Minimum variance	✓	—	1.2	1.3	1355.95
Minimum variance	✓	✓	1.3	1.7	12.51
Nearest neighbor	—	—	1.6	11.7	13.48
Nearest neighbor	—	✓	2.6	12.1	1.61
Nearest neighbor	✓	—	1.1	1.2	1243.60
Nearest neighbor	✓	✓	0.8	0.9	11.89

eters. Figures 10(a) and 11(a) illustrate the schematic representation of experimental outcomes in two distinct scenarios. Figure 10 presents the robot's planning results obtained from three different strategies in Scenario 1, whereas Figure 11 depicts corresponding results in Scenario 2. It is evident that the

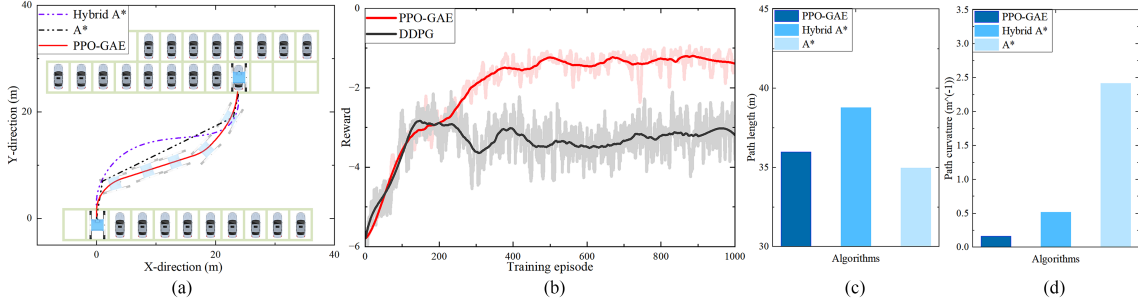


Figure 10 (Color online) Robot autonomous planning results based on actual experiments. (a) Experimental scenario 1; (b) comparison among different RL algorithms; (c) experimental path lengths of different algorithms; (d) experimental path curvature of different algorithms.

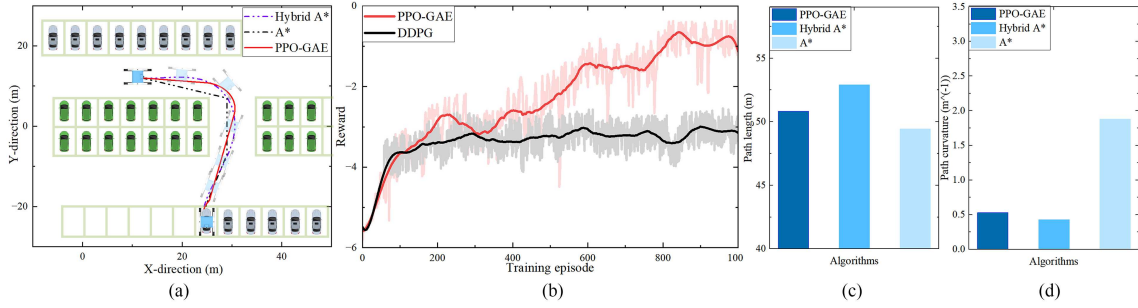


Figure 11 (Color online) Robot autonomous planning results based on actual experiments. (a) Experimental scenario 2; (b) comparison among different RL algorithms; (c) experimental path lengths of different algorithms; (d) experimental path curvature of different algorithms.

PPO-based planner consistently outperformed the other two methods.

The experimental results for scenario 1, as shown in the analysis, indicate the performance of three different path-planning algorithms. When using the A* planning algorithm, the total time consumption was 10.34 s, with a final target position planning error of 0.85° and a maximum path curvature of 2.42 m⁻¹. In comparison, the hybrid A* planning approach required 10.57 s, achieving a reduced target position error of 0.34° and a significantly lower maximum curvature of 0.74 m. The PPO-GAE algorithm outperformed both, requiring only 8.75 s, with a final target position error of 0.23° and a maximum curvature of 0.52 m⁻¹, demonstrating superior efficiency and precision.

Similarly, the experimental results from Scenario 2 indicate that when the A* planning algorithm is employed, the total execution time is 11.52 s, the final target position error is 1.78°, and the maximum path curvature reaches 1.88 m⁻¹. In comparison, the hybrid A* algorithm completes the task in 9.75 s, with a reduced target position error of 1.48° and a significantly lower maximum curvature of 0.43 m⁻¹. The PPO-GAE algorithm demonstrates the best performance, requiring only 9.42 s, with a final target position error of 0.46° and a maximum curvature of 0.53 m⁻¹.

Figure 12 illustrates the results of 2D LiDAR localization for the vehicle's wheels. The stationary mechanical arm, equipped with a 2D LiDAR, identifies the rear wheels between 0–2.4 s, prompting the vehicle to initiate posture refinement. By 5.9 s, the center of the front wheels is detected, indicating the vehicle has entered a parked state. Subsequently, the mobile vehicle arm utilizes *X*-direction data from the 2D LiDAR to adjust the wheelbase, while refining the pose based on *Y*-direction data. Wheelbase adjustment is completed by 16.5 s, enabling the mechanical arm to successfully grip and lift the finished vehicle.

5 Conclusion and future work

An embodied intelligent mobile transportation robot endowed with visual perception, cognitive processing, and gripping capabilities is proposed for handling complex tasks. Such robots, which autonomously acquire and analyze environmental information and independently perform decision-making, planning, and motion control, closely resemble advanced life forms. Firstly, we designed a novel transportation robot

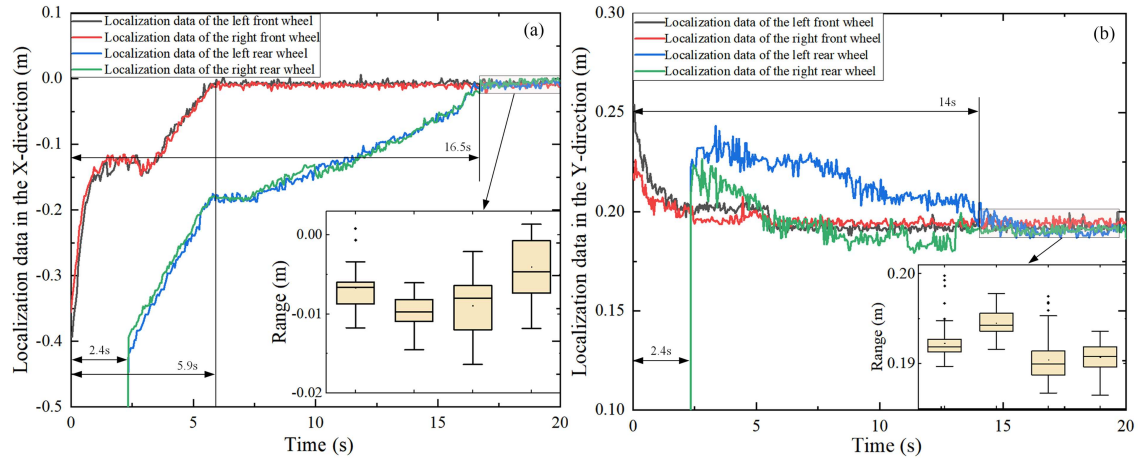


Figure 12 (Color online) Experimental results of static target docking in docking phase. (a) Localization data in the X-direction; (b) localization data in the Y-direction.

that integrates perception, autonomous planning, and motion control. The robot controller achieves seamless integration of sensing, computing, and control. Secondly, the effectiveness of the proposed method was validated through target localization experiments, demonstrating the robot controller's capability to autonomously determine the pose of the target vehicle. Furthermore, the application of PPO was extended through a new method employing deep reinforcement learning to approximate optimal trajectories. This method utilizes GAE to balance bias and variance, thereby closely approximating the optimal strategy. Finally, based on the proposed framework, a novel conceptual design was developed, overcoming traditional limitations of robots in RO/RO terminal operations, particularly addressing the challenges posed by limited flexibility and autonomy in dynamically changing industrial environments. The present research lays a foundation for future studies initially focused on transportation robots, with the scope soon expanding to broader application scenarios. Additionally, by enhancing large-scale models, generative artificial intelligence can enable robotic autonomous operations, equipping robots with greater versatility to handle complex automation tasks, thus advancing their general intelligence capabilities.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 62473044) and Fundamental Research Funds for the Central Universities of China (Grant No. 2024CX06030).

References

- Li Z, Wu W R, Wang J L, et al. Continuous advantage learning for minimum-time trajectory planning of autonomous vehicles. *Sci China Inf Sci*, 2024, 67: 172206
- Sun J, Lei H, Zhang J, et al. Adaptive consensus control of multiagent systems with an unstable high-dimensional leader and switching topologies. *IEEE Trans Ind Inf*, 2024, 20: 10946–10953
- Ma M, Cheng L. A human-robot collaboration controller utilizing confidence for disagreement adjustment. *IEEE Trans Robot*, 2024, 40: 2081–2097
- Li G, Cheng L, Zhang C. Design and modeling of a bioinspired flexible finger exoskeleton for strength augmentation. *IEEE ASME Trans Mechatron*, 2024, 29: 4389–4400
- He W, Zhang Y, Zhou W. Observerless output feedback control of DC-DC converters feeding a class of unknown nonlinear loads via power shaping. *IEEE Trans Circuits Syst I*, 2024, 71: 2951–2963
- Huang H, He W, Zou Y, et al. USTButterfly: a servo-driven biomimetic robotic butterfly. *IEEE Trans Ind Electron*, 2024, 71: 1758–1767
- Wu X T, Tang Y, Mao S, et al. Stability analysis and stabilization of semi-Markov jump linear systems with unavailable sojourn-time information. *Sci China Inf Sci*, 2024, 67: 172201
- Qian F, Du W L, Zhong W M, et al. Artificial intelligence-assisted design of new chemical materials: a perspective. *Sci China Inf Sci*, 2024, 67: 186201
- Slanina Z, Pergl I, Kedron P. Automated guided vehicle control system for automated parking purposes. *IFAC-PapersOnLine*, 2022, 55: 362–367
- Pan T, Yan J, Zhou S, et al. Spin-type forward motion mode based on double steering wheel parking AGV. *Ind Robot*, 2020, 47: 777–787
- Qu S, Xu Z, Lu F, et al. ITDP-robot: design of an intelligent transport dispatch parking robot. *Smart City Appl*, 2019, 2: 24–32
- Ning Z, Wang X, Wang J, et al. Vehicle pose estimation algorithm for parking automated guided vehicle. *Int J Adv Robotic Syst*, 2020, 17: 1729881419891335
- Polack P, Dallen L, Cord A. Strategy for automated dense parking: how to navigate in narrow lanes. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 9196–9202
- Endo M, Hirose K, Hirata Y, et al. A car transportation system by multiple mobile robots-cart. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008. 2795–2801
- Kashiwazaki K, Yonezawa N, Endo M, et al. A car transportation system using multiple mobile robots: iCART II. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011. 4593–4600

- 16 Amanatiadis A, Henschel C, Birkicht B, et al. AVERT: an autonomous multi-robot system for vehicle extraction and transportation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2015. 1662–1669
- 17 Wu E Q, Gao Y, Tong W, et al. Cognitive state detection in task context based on graph attention network during flight. *IEEE Trans Syst Man Cybern Syst*, 2024, 54: 5224–5236
- 18 Tong W, Guan X, Zhang M, et al. Edge-assisted epipolar transformer for industrial scene reconstruction. *IEEE Trans Automat Sci Eng*, 2025, 22: 701–711
- 19 Tong W, Zhang M, Zhu G Y, et al. Robust depth estimation based on parallax attention for aerial scene perception. *IEEE Trans Ind Inf*, 2024, 20: 10761–10769
- 20 Zhu Y, Mottaghi R, Kolve E, et al. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2017. 3357–3364
- 21 Chen T, Murali A, Gupta A. Hardware conditioned policies for multi-robot transfer learning. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS), 2018. 9355–9366
- 22 Luck K, Amor H, Calandra R. Data-efficient co-adaptation of morphology and behavior with deep reinforcement learning. In: Proceedings of the Conference on Robot Learning (CoRL), 2020. 854–869
- 23 Ha D. Reinforcement learning for improving agent design. *Artif Life*, 2019, 25: 352–365
- 24 Fager P, Calzavara M, Sgarbossa F. Modelling time efficiency of cobot-supported kit preparation. *Int J Adv Manuf Technol*, 2020, 106: 2227–2241
- 25 Zhao H, Pan F, Ping H, et al. Agent as cerebrum, controller as cerebellum: implementing an embodied LMM-based agent on drones. 2023. ArXiv:2311.15033
- 26 Ding N, Ming R, Wang B. Efficient convex-hull-based vehicle pose estimation method for 3D LiDAR. *Transp Res Record-J Transp Res Board*, 2024, 2678: 1172–1182
- 27 Dyllong E, Luther W. Verified convex hull and distance computation for octree-encoded objects. *J Comput Appl Math*, 2007, 199: 358–364
- 28 Rao R M, Emenonye D R. Iterative RNDOP-optimal anchor placement for beyond convex hull ToA-based localization: performance bounds and heuristic algorithms. *IEEE Trans Veh Technol*, 2024, 73: 7287–7303
- 29 Garloff J, Jansson C, Smith A P. Inclusion isotonicity of convex-concave extensions for polynomials based on Bernstein expansion. *Computing*, 2003, 70: 111–119
- 30 Heng P, Sun Y. Algorithms for convex hull finding in undirected graphical models. *Appl Math Comput*, 2023, 445: 127852
- 31 Xu H, Yan Z, Xuan J, et al. Improving proximal policy optimization with alpha divergence. *Neurocomputing*, 2023, 534: 94–105
- 32 Li Z, Sun J, Marques A G, et al. Pontryagin’s minimum principle-guided RL for minimum-time exploration of spatiotemporal fields. *IEEE Trans Neural Netw Learn Syst*, 2025, 36: 5375–5387
- 33 Peng B, Keskin M F, Kulcsár B, et al. Connected autonomous vehicles for improving mixed traffic efficiency in unsignalized intersections with deep reinforcement learning. *Commun Transp Res*, 2021, 1: 100017
- 34 Chen Y, Zhang F, Liu Z. Adaptive bias-variance trade-off in advantage estimator for actor-critic algorithms. *Neural Netws*, 2024, 169: 764–777