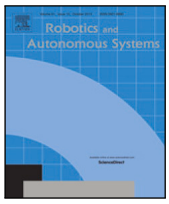
Robotics and Autonomous Systems 205 (2026) 105637

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Trajectory planning for 4WIS autonomous ground vehicles in RORO terminals using CBF-enhanced deep reinforcement learning

Runjiao Bao [a], Shoukun Wang [a], Yongkang Xu [b,*]

[a] National Key Lab of Autonomous Intelligent Unmanned Systems, School of Automation, Beijing Institute of Technology, Beijing, 100081, China
[b] Institute of Intelligence Technology and Robotic Systems, Shenzhen Research Institute of Nankai University, Shenzhen, 518000, China

ARTICLE INFO

Keywords:
Autonomous ground vehicles
Trajectory planning
Control barrier function
Deep reinforcement learning

ABSTRACT

In the operational environment of Roll-On/Roll-Off (RORO) terminals, intelligent autonomous ground vehicle (AGV) systems, which integrate perception and motion planning modules, are widely used for the transportation of vehicles and cargo. The motion planning module plays a crucial role in improving operational efficiency and reducing energy consumption. To meet the stringent requirements for trajectory smoothness, kinematic feasibility, and safety in transfer tasks, we propose a novel real-time trajectory planning framework. This framework integrates control barrier functions with proximal policy optimization, enabling direct control command outputs without the need for separate tracking modules typically required in traditional planning methods. We design a composite multi-objective reward function to optimize key performance indicators, including time efficiency, trajectory smoothness, and collision safety, while adhering to the AGV's kinematic constraints. Simulation experiments and physical tests validate the effectiveness of the proposed framework, with results demonstrating significant improvements in motion performance and operational efficiency. The method also exhibits real-time applicability, with an average computation time of less than 3 ms per control step, making it suitable for practical implementation in RORO terminals.

1. Introduction

In the context of accelerating the decarbonization of global terminals, the electrification and automation of roll-on/roll-off (RORO) logistics terminals have increasingly been recognized as key strategies for reducing carbon emissions in the transportation sector [1]. As critical hubs in automotive logistics networks, RORO terminals handle large volumes of vehicle loading, unloading, and transfer operations, thereby imposing stringent requirements on operational efficiency, precision, and sustainability. To replace conventional human-operated systems, autonomous electric automated guided vehicles (AGVs) have emerged as a promising solution, enabling efficient and environmentally friendly transportation of vehicles and cargo within terminal areas [2]. Fig. 1 illustrates a representative working scenario of an AGV operating in a terminal environment.

In practical RORO terminal operations, AGVs are required to repeatedly perform standard maneuvers, within structured and largely fixed environments. Due to this high level of repetition, deficiencies in trajectory quality are repeatedly executed and accumulated over long-term operation, resulting in increased energy consumption, mechanical wear, and reduced operational efficiency. Consequently, RORO terminals applications impose stricter requirements on trajectory smoothness, kinematic feasibility, and energy efficiency than one-off navigation tasks. By optimizing the movement and energy consumption of electric AGVs, this research not only supports the overarching goals of transport and power sector decarbonization but also demonstrates the transformative role of artificial intelligence in advancing green terminal infrastructures [3].

The motion planning module is a critical component of an AGV system, responsible for high-level decision-making and action execution [4]. It determines the vehicle's path and travel strategy to ensure safe navigation in complex traffic environments, including obstacle avoidance and route selection. In RORO terminals, where parking areas and drivable regions are fixed, the globally optimal path between a given start and destination is uniquely determined. Consequently, motion planning in such environments primarily shifts from global route discovery to trajectory quality optimization along repeatedly executed paths, which plays a decisive role in long-term operational performance.

Existing motion planning methods can be broadly categorized into search-based, sampling-based, optimization-based, and learning-based approaches. Search-based planners rely on discretized state spaces and provide deterministic convergence with low implementation complexity, but may generate trajectories that are difficult to execute when

* Corresponding author.
E-mail address: xuyk@nankai.edu.cn (Y. Xu).

https://doi.org/10.1016/j.robot.2026.105637
Received 2 December 2025; Received in revised form 23 June 2026; Accepted 15 July 2026
Available online 21 July 2026

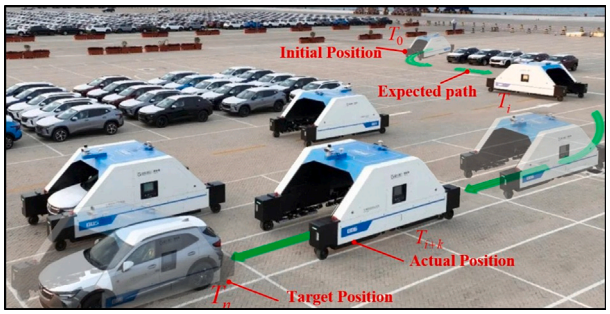


Fig. 1. AGV for vehicle transportation at RORO terminals.

vehicle kinematic constraints are not explicitly modeled [5]. Hybrid A* [6] improves executability by embedding kinematically feasible motion primitives into the search process. However, the effectiveness of these methods strongly depends on heuristic design and primitive quality, which is difficult to guarantee in complex environments. Sampling-based methods construct feasible paths via random sampling [7], offering good scalability in high-dimensional and constrained spaces, but their probabilistic convergence makes it difficult to consistently guarantee solution quality under limited computation time. Moreover, both approaches are primarily designed for path generation and therefore require additional velocity or time-parameterization modules to produce fully executable trajectories.

Optimization-based approaches formulate motion planning as a constrained optimization problem. Offline nonlinear optimization strategies [8] and real-time convex feasible set methods for nonconvex problems [9] have been proposed to improve tractability. Heuristic techniques such as genetic algorithms [10] and particle swarm optimization are therefore often used to provide initial solutions, followed by local refinement [11]. Nevertheless, balancing model fidelity, computational efficiency, and solution quality remains challenging.

In recent years, learning-based methods have attracted increasing attention in motion planning research. One major branch is supervised learning, which employs network architectures such as LSTM [12], graph neural networks [13], and residual dense networks [14] to learn perception-to-control mappings from offline datasets. However, these methods require optimal trajectories to be generated in advance and often exhibit limited generalization. Another major branch is reinforcement learning, which optimizes control policies through interaction with the environment. Although reinforcement learning typically involves substantial training time, the resulting policies often demonstrate improved generalization and real-time performance. Successful applications on unmanned aerial vehicles [15] and ground robots [16,17] further demonstrate the effectiveness of deep reinforcement learning for trajectory planning.

Another important background of this study is that the target AGV platform adopts a four-wheel independent steering (4WIS) configuration. In recent years, extensive research has been conducted on 4WIS platforms, which can generally be categorized into multi-modal and single-modal frameworks. Multi-modal approaches explicitly define multiple motion modes, each associated with distinct kinematic constraints, and have been investigated for local decision-making and trajectory tracking [18]. However, in highly structured scenarios such as port terminals, where operational routes are largely repetitive, the demand for frequent mode switching is limited. In such settings, introducing multiple motion modes often provides marginal benefits and may degrade trajectory continuity and smoothness, thereby reducing operational efficiency. Accordingly, this study focuses on the dual-Ackermann steering mode of the 4WIS platform [19]. Prior work under this mode has investigated kinematic and dynamic modeling [20], as well as trajectory tracking [21] and motion control methods, showing that this mode can effectively reduce the turning radius while maintaining a unified motion constraint structure. Due to the large physical size of transfer AGVs, this property can significantly enhance their maneuverability and passability in constrained spaces such as narrow lanes and intersections, thereby further improving operational efficiency.

Grid-based and search-based methods typically generate geometric paths that require additional post-processing to become executable trajectories. Optimization-based approaches often simplify environmental constraints to meet real-time requirements [22]. These methods usually output full trajectory sequences and rely on tracking controllers to produce control commands. In contrast, learning-based methods can directly output control actions from the current state, making them more suitable for scenarios requiring fast response. Moreover, supervised learning methods are inherently imitation-based and their performance strongly depends on the quality of the training dataset, whereas deep reinforcement learning enables AGVs to learn optimal policies through interaction and exploration.

Motivated by these considerations, this paper proposes a trajectory planning method that integrates control barrier functions with proximal policy optimization (PPO) [23]. The proposed method directly outputs instantaneous control commands and provides the underlying control law for the AGV's wheels based on the dual-Ackermann steering, eliminating the need for additional tracking modules compared to traditional trajectory planning approaches. The main contributions of this work are summarized as follows:

1. A real-time trajectory planning framework is proposed by integrating control barrier functions with proximal policy optimization (CBF-PPO), which directly outputs control commands online, enabling unified optimization of planning and execution without reference paths or separate tracking modules.
2. A composite multi-objective reward formulation is designed to jointly account for time efficiency, trajectory smoothness, terminal accuracy, and collision safety, enabling simultaneous optimization of multiple performance objectives under AGV kinematic constraints.
3. Detailed comparative experiments were conducted to demonstrate the effectiveness of the proposed planning method. Extensive experiments with random perturbations demonstrate an average computation time below 3 ms per control step, validating the real-time capability and practical applicability of the proposed approach.

2. Preliminaries and problem formulation

2.1. Environment and planning challenges

To accommodate the large-scale storage and transportation requirements of commercial vehicles, RORO terminals commonly utilize standardized yard layouts. These layouts involve densely packed vehicles arranged in rectangular berths, with interconnecting road networks that follow a structured grid-like design. The road network is characterized by regular geometric patterns and numerous four-way and T-junctions, as illustrated in Fig. 2. In daily operations, AGVs are tasked with navigating through both straight road segments and crossroads to transport vehicles between different yard locations. Given the fixed nature of the terminal's road topology, the network is often abstracted as a directed graph with nodes and edges, where global path planning is dictated by higher-level scheduling systems.

A key challenge for onboard real-time path planning is executing efficient and safe turning maneuvers within predefined traffic corridors. In practice, AGVs often encounter misalignment with the road centerline, which can be attributed to factors such as localization inaccuracies, tire slippage, or delays in actuator responses. As a result, the planning and control systems must exhibit sufficient robustness to ensure reliable navigation through these crossroads. Additionally,

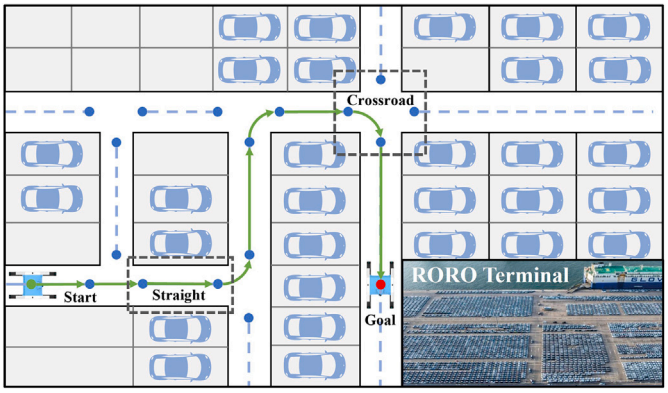


Fig. 2. Vehicle yard and road network layout in RORO terminals.

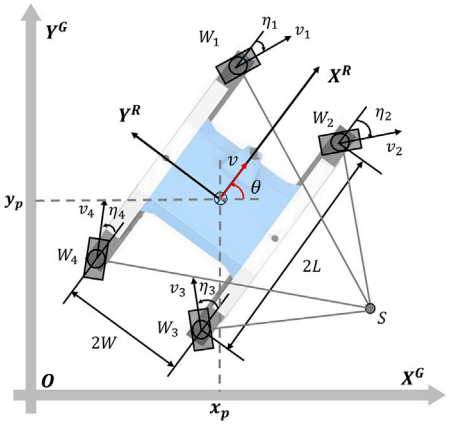


Fig. 3. Configuration of the 4WIS AGV.

AGVs must have the inherent capability to detect and avoid unexpected obstacles, such as cargo, personnel, or other vehicles, in real-time. This requires the AGV's algorithms to process and respond to dynamic changes in the environment with high responsiveness.

2.2. 4WIS AGV kinematic model

The AGV considered in this work adopts a 4WIS configuration, in which all four wheels are actively involved in steering. This configuration significantly enhances vehicle maneuverability and motion flexibility, particularly in constrained environments. The configuration of the 4WIS AGV is illustrated in Fig. 3. In the global coordinate frame, (x, y, θ) denote the position and orientation of the vehicle, and v represents the forward velocity. v_i and η_i indicate the linear velocities and steering angle of wheel i. The parameters L and W denote the longitudinal and lateral distances from the vehicle center to each wheel, respectively.

In practical applications, the Ackermann steering principle ensures that all wheels point to a common instantaneous center of rotation during steering, thereby maintaining stable steering characteristics and effectively reducing tire wear. Compared to traditional single Ackermann steering, the double Ackermann steering corresponding to 4WIS optimizes the vehicle's steering performance by coordinating the steering angles of the front and rear wheels, enabling it to navigate narrower environments more efficiently.

Similar to Ackermann steering, double Ackermann steering adjusts the rear wheels of the AGV to ensure the desired center of rotation characteristics [24]. To ensure steering symmetry and speed coordination, the following constraints are introduced:

$$\begin{cases} \eta_1 = -\eta_3, \quad \eta_2 = -\eta_4 \\ v_1 = v_3, \quad v_2 = v_4 \\ \cot\eta_2 - \cot\eta_1 = 2W/L. \end{cases} \tag{1}$$

When the AGV's control variables, forward velocity v and equivalent steering angle η, are provided, the actual execution quantities for each wheel can be calculated as follows:

$$\begin{cases} \cot\eta_1 = \cot\eta - W/L, & \cot\eta_2 = \cot\eta + W/L \\ v_1 = v\tan\eta/\sin\eta_1, & v_2 = v\tan\eta/\sin\eta_2. \end{cases} \tag{2}$$

Through the above calculation method, the equivalent control quantities can be uniquely mapped to the actual steering angles and wheel speeds via the Ackermann constraints. However, the current double Ackermann model typically simplifies the entire system to a bicycle model after the computation, which negatively impacts the accuracy of the AGV's kinematic model [19,25]. Therefore, to improve the model's accuracy, it is necessary to perform an independent kinematic analysis of the four wheels of the AGV and develop a complete model. Assuming that the vehicle center coincides with its geometric center, the local coordinates of the four wheels (x_i, y_i) in the body-fixed frame are given by:

$$x_i = (-1)^{\lfloor (i-1)/2 \rfloor} L, \;\; y_i = (-1)^{i-1} W. \tag{3}$$

Building on the findings in [26], and assuming that the vehicle's center coincides with its geometric center, that all wheels have the same radius and mass, and that sliding friction is neglected, the vehicle's kinematic model can be described as:

$$\begin{cases} \dot{x} = \frac{1}{4}\sum_{i=1}^{4} v_i \cos(\eta_i + \theta) = v\cos\theta \\ \dot{y} = \frac{1}{4}\sum_{i=1}^{4} v_i \sin(\eta_i + \theta) = v\sin\theta \\ \dot{\theta} = \sum_{i=1}^{4} \frac{v_i(-y_i\cos\eta_i + x_i\sin\eta_i)}{4{x_i}^2 + 4{y_i}^2} = \frac{v\tan\eta}{L}. \end{cases} \tag{4}$$

By combining the double Ackermann model with the aforementioned kinematic model, a more accurate representation of the AGV's motion model is achieved, enabling the development of planning strategies with improved performance. We opted for a kinematic model over a more complex dynamic model, as the AGV operates in a low-to-medium speed environment, where the kinematic model suffices. Furthermore, the decoupled modeling approach, which separates the motion mode constraints from the AGV's 4WIS kinematic model, facilitates future expansion of motion modes.

2.3. Collision avoidance constraints

In practical applications, on-road agents such as AGVs cannot be reasonably approximated as point masses [27]. While collision checking can be simplified in scenarios where the drivable region boundaries admit explicit functional representations, such assumptions rarely hold in complex environments with intersections or dense obstacles. As a result, point-mass abstractions and boundary-function-based methods lack general applicability in realistic AGV operation scenarios. Consequently, the geometric extent of both the vehicle and surrounding obstacles must be explicitly considered to ensure reliable collision avoidance.

The region occupied by the vehicle at time t can be denoted as $\mathcal{S}(t)$. Let $\mathcal{O}$ represent the total region occupied by obstacles. $\mathcal{N}$ is the region occupied by the AGV at its initial position. $\mathbf{R}_t$ represents the rotation matrix at time t, and $\mathbf{P}_t$ represents the translation matrix at time t. Therefore:

$$\mathcal{S}(t) = \mathbf{R}_t\mathcal{N} + \mathbf{P}_t, \quad \mathcal{S}(t) \cap \mathcal{O} = \emptyset. \tag{5}$$

In practical applications, directly using the above equations may encounter issues with non-differentiable and non-convex functions. Collision avoidance is typically achieved by constructing feasible regions through optimization conditions or performing real-time collision detection in dynamic environments. When constructing optimization

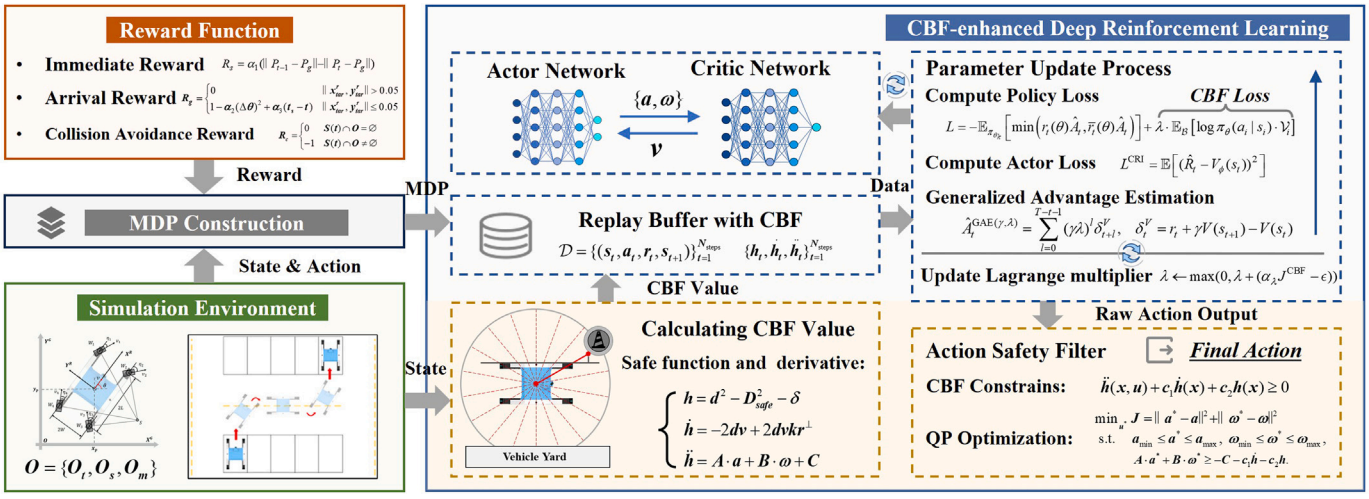


Fig. 4. Proposed CBF-enhanced DRL trajectory planning framework (CBF-PPO).

conditions, convex constraints are usually formed by creating safety corridors [28] or utilizing convex hulls and geometric relations [29]. Real-time collision detection is generally implemented by densely sampling boundary points and performing position validation. Additionally, LiDAR sensors can be used to detect obstacles. Our implementation detects whether 2D LiDAR points are too close to the vehicle and sets a safety threshold to effectively avoid collision risks.

2.4. Objective and constraints formulation

To clearly describe the objective task, we model it as a standard constrained optimal control problem. The full system state and control inputs are defined as: $\mathbf{x} = [x, y, v, \theta, \eta]^{\mathsf{T}}, \mathbf{u} = [a, \omega]^{\mathsf{T}}$, where a and ω represent acceleration and steering rate, respectively. The primary objective is to complete the transferring operation in the shortest time, so we minimize the terminal time t_f as the main objective function. In addition, to prevent excessive steering and speed changes, we introduce a control cost term. Inspired by [22], the AGV planning problem can be modeled as follows:

$$
\begin{aligned}
\min_{\mathbf{u}} \quad & J = \int_0^{t_f} \left(1 + \alpha a(t)^2 + \beta v(t)^2 \omega(t)^2\right) dt \\
\text{s.t.} \quad & \mathbf{x}_0 = \mathbf{x}_{\text{init}},\ \mathbf{x}_{t_f} = \mathbf{x}_{\text{goal}}, \\
& \mathbf{x}_{\min} \le \mathbf{x}_k \le \mathbf{x}_{\max},\ \mathbf{u}_{\min} \le \mathbf{u}_k \le \mathbf{u}_{\max}, \\
& S(t) = \mathbf{R}_t \mathcal{N} + \mathbf{P}_t,\ S(t) \cap \mathcal{O} = \emptyset, \\
& \text{4WIS kinematic constraints (Eq. (4)).}
\end{aligned} \tag{6}
$$

It is worth noting that the high computational cost of numerical optimization often hinders real-time planning. Additionally, traditional optimizers are highly sensitive to initial guesses, which can lead to convergence to infeasible or poor local optima in challenging initial conditions or complex scenarios. These limitations have motivated the search for alternative solution strategies.

3. Methodology

Building upon these considerations, we propose a trajectory planner based on control barrier functions combined with deep reinforcement learning. The planner takes vehicle state parameters and LiDAR perception data as input states, and generates control actions and trajectories that approach optimal solutions in real time. The overall framework is illustrated in Fig. 4.

3.1. Control barrier functions

The Control Barrier Function (CBF) is used to provide a guarantee method for safety constraints in control systems. The safe set of the CBF is defined as $\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n : h(\mathbf{x}) \ge 0\}$. When the CBF condition is satisfied, the set $\mathcal{C}$ is forward invariant, meaning that trajectories starting inside $\mathcal{C}$ will always remain inside $\mathcal{C}$.

To accurately consider the physical size of the vehicle, the vehicle is modeled as a rectangle, and the minimum distance from the obstacle to the vehicle's boundary is calculated. Let l and w denote the overall length and width of the AGV body, the global coordinates of the four corner points $\mathbf{p}_i$ can then be obtained through the following rotational transformation:

$$
\mathbf{p}_i = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} (-1)^{\lfloor (i-1)/2 \rfloor} l/2 \\ (-1)^{i-1} w/2 \end{bmatrix}. \tag{7}
$$

The closest obstacle point in the radar distance sequence is denoted as $\mathbf{p}_{obs} = [x_{obs}, y_{obs}]^T$. The nearest AGV edge to the point obstacle must first be detected, and this edge will be used for constructing the CBF. Consider the front edge (connecting $\mathbf{p}_1$ and $\mathbf{p}_2$) as an example. The construction method for CBF on other edges is the same, simply replacing the variables according to Eq. (7).

Let the relative position vector be $\Delta\mathbf{p} = \mathbf{p}_{obs} - \mathbf{p}_1$, and the directed distance from the point obstacle to its nearest edge is expressed as:

$$
d = \Delta x \cos\theta + \Delta y \sin\theta, \tag{8}
$$

where $\Delta x = x_{obs} - x_1$, $\Delta y = y_{obs} - y_1$, and x_1, y_1 are the global coordinates of $\mathbf{p}_1$. To avoid absolute value operations and simplify differentiation, the square of the distance is used as the metric. The CBF corresponding to the front edge is defined as:

$$
h = d^2 - D_{safe}^2 - \delta, \tag{9}
$$

where D_{safe} is the safety distance threshold, and δ is a small constant that affects the scaling of the distance. The smaller δ is, the closer the safety set will be to the prescribed safety distance.

Since the AGV is a system where the higher-order derivatives of h are influenced by the control inputs, a second-order CBF is used. The constraint condition is:

$$
\ddot{h}(\mathbf{x}, \mathbf{u}) + c_1 \dot{h}(\mathbf{x}) + c_2 h(\mathbf{x}) \ge 0. \tag{10}
$$

The first derivative of $h(\mathbf{x})$ can be obtained using the chain rule and the vehicle's kinematic equations:

$$
\begin{aligned}
& \dot{h} = -2dv + 2dvkr^{\perp},\ k = \frac{\tan\eta}{L}, \\
& r^{\perp} = -\Delta x \sin\theta + \Delta y \cos\theta + w/2,
\end{aligned} \tag{11}
$$

where $r^{\perp}$ represents the perpendicular geometric quantity of the obstacle relative to the vehicle's front edge.

By differentiating $\dot{h}$ again and substituting $\dot{v} = a$ and $\dot{\eta} = \omega$, the second derivative of $h(\mathbf{x})$ is obtained as:

$$\ddot{h} = A \cdot a + B \cdot \omega + C, \tag{12}$$

where the coefficients are:

$$A = -2d\left[1 - kr^{\perp}\right], \quad B = 2dvr^{\perp}\sec^2\eta/L,$$
$$C = 2v^2\left[1 - 2kr^{\perp} + k^2\left[(r^{\perp})^2 - d(d + l/2)\right]\right]. \tag{13}$$

This ensures that the safety constraints are effectively enforced, providing a robust guarantee for obstacle avoidance and safe operation of the AGV in dynamic environments.

3.2. Markov decision process modeling

3.2.1. State space and action space

To ensure accurate perception of the agent's state and the surrounding environment, the observation space is defined as $O = \{O_t, O_s, O_m\}$. The sensor perception component O_t originates from 3D LiDAR point clouds. The raw data is Z-axis cropped, downsampled, and transformed into a 360° omnidirectional distance sequence that mimics a 2D LiDAR scan, yielding a 180-dimensional vector. It is worth noting that in real-world scenarios, the boundary of stack areas typically does not contain explicit obstacles and cannot be directly scanned by the LiDAR. However, the global coordinates of the stack boundaries and the AGV's position are known, so the LiDAR distance sequence needs to be cropped based on these known infeasible regions.

AGV state is defined as $O_s = \{\eta, v, t\}$, where t represents the current time, used to mark the vehicle's temporal state. To mitigate policy overfitting and poor generalization often caused by using global coordinates, the task information $O_m = \{x^r_{tar}, y^r_{tar}, \theta^r_{tar}\}$ is parameterized as the local sub-goal's pose within the AGV's coordinate frame. This design preserves task semantics while effectively decoupling the policy from the absolute coordinate system, thereby improving cross-scene transferability and robustness.

The action space A defines the agent's control commands. In practical work scenarios, considering the vehicle's mechanical structure and steering efficiency, kinematic constraints must be incorporated into the action space design. To avoid violating the vehicle's dynamic limitations during turning or acceleration, we define the action space using acceleration and angular acceleration. This better aligns with the AGV's kinematic model, ensuring control commands adhere to motion constraints. We define $A = \{a, \Delta\eta\}$, where $\Delta\eta \in [\Delta\eta_{\min}, \Delta\eta_{\max}]$ represents the wheel steering control.

3.2.2. Reward function design

The reward function is designed to guide the agent's actions toward an optimal strategy. To motivate the AGV to move forward and avoid idle behavior, an immediate reward R_s is granted for each successful step without collision. Specifically, R_s is formulated to incentivize the vehicle to minimize its distance to the target parking space. The immediate reward function is defined as:

$$R_s = \alpha_1\left(\|\mathbf{P}_{t-1} - \mathbf{P}_g\| - \|\mathbf{P}_t - \mathbf{P}_g\|\right), \tag{14}$$

where $\mathbf{P}_t$ and $\mathbf{P}_{t-1}$ represent the position vectors of the AGV at time t and $t-1$ respectively, $\mathbf{P}_g$ denotes the coordinate vector of the target goal, $\|\cdot\|$ signifies the L2 norm used to calculate the Euclidean distance.

To prevent collisions between the vehicle and obstacles, or when the vehicle enters the yard, a significant penalty R_c is applied to steps where collisions occur. Once the collision penalty function R_c takes effect, the current trajectory is immediately terminated, and the final reward is calculated. The collision penalty function R_c is defined as:

$$R_c = \begin{cases} 0 & S(t) \cap O = \emptyset \\ -1 & S(t) \cap O \neq \emptyset. \end{cases} \tag{15}$$

To encourage the AGV to reach the target state quickly and aligned with the target's orientation, a fixed reward is given when the vehicle's center is within 5 cm of the target. The system also applies penalties for directional deviations and rewards based on the time required to reach the target. Similar to collision penalties, the trajectory terminates and a return value is calculated once the arrival reward function R_g is triggered:

$$R_g = \begin{cases} 0 & \|x^r_{tar}, y^r_{tar}\| > 0.05 \\ 1 - \alpha_2(\Delta\theta)^2 + \alpha_3(t_s - t) & \|x^r_{tar}, y^r_{tar}\| \le 0.05, \end{cases} \tag{16}$$

where $\Delta\theta$ represents the difference in orientation angles between the AGV and the target state. t_s denotes the reference standard time, which can be either roughly estimated or computed using conventional planning algorithms. During training, the reference path is defined as the total length of the road centerline, and the corresponding time is calculated based on a predefined reference speed.

The final composite reward function R is defined as:

$$R = R_s + R_c + R_g. \tag{17}$$

3.3. CBF-PPO algorithm

Algorithm 1 CBF-PPO Algorithm Procedure

Initialize policy network π_θ, value network V_ϕ, Lagrange multiplier λ
Set hyperparameters: α_λ, ϵ, α_1, α_2, D_{safe}, k, N_{steps}
for Iteration $n = 1, 2, \ldots, N$ **do**
  Collect trajectory $D = \{(s_t, a_t, r_t, s_{t+1})\}_{t=1}^{N_{\text{steps}}}$
  Calculate CBF values $\{h_t, \dot{h}_t, \ddot{h}_t\}_{t=1}^{N_{\text{steps}}}$
  Save D and $\{h_t, \dot{h}_t, \ddot{h}_t\}_{t=1}^{N_{\text{steps}}}$ in the rollout buffer
  Compute GAE Value $\hat{A}_t$ using V_ϕ
  Compute reward-to-go $\hat{R}_t$
  for Training epochs $e = 1, \ldots, E$ **do**
    Sample mini-batches $\mathcal{B}$ from the rollout buffer D
    for Mini-batches $\mathcal{B} \subset D$ **do**
      Compute PPO objective L^{PPO} via Eq. (22)
      Compute violation degree: $\mathcal{V}_t$ via Eq. (20)
      CBF loss: $L^{\text{CBF}} = \mathbb{E}_{\mathcal{B}}\left[\log \pi_\theta(a_t|s_t) \cdot \mathcal{V}_t\right]$
      Total loss: $L = -L^{\text{PPO}} + \lambda \cdot L^{\text{CBF}}$
      Update policy parameters: $\theta \leftarrow \theta - \alpha_\theta \nabla_\theta L$
      Compute critic loss: $L^{\text{CRI}} = \mathbb{E}\left[(\hat{R}_t - V_\phi(s_t))^2\right]$
      Update value parameters: $\phi \leftarrow \phi - \alpha_\phi \nabla_\phi L^{\text{CRI}}$
    end for
  end for
  Evaluate violation degree: $J^{\text{CBF}} = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathcal{V}_i$
  Update Lagrange multiplier: $\lambda \leftarrow \max(0, \lambda + \alpha_\lambda(J^{\text{CBF}} - \epsilon))$
end for
Return the trained policy π_θ

To provide safety guarantees during the policy learning process, we model the safe autonomous navigation problem as a constrained Markov decision process. The proposed CBF-PPO framework directly integrates safety awareness into the policy gradient optimization process, allowing the agent to learn inherently safe behavior policies. The optimization objective is formulated as:

$$\max_\theta \quad \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] \tag{18}$$

$$\text{s.t.} \quad \mathbb{E}_{s\sim\rho^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\mathcal{V}(s,a)\right] \le \epsilon, \tag{19}$$

where θ represents the policy network parameters, τ denotes the trajectory, γ is the discount factor, and ρ^{π_θ} is the state distribution induced by policy π_θ. The CBF violation degree $\mathcal{V}(s,a)$ in the constraint is defined as:

$$\mathcal{V}(s,a) = \max\left(0, -\left(\ddot{h}(s,a) + c_1\dot{h}(s) + c_2 h(s)\right)\right), \tag{20}$$

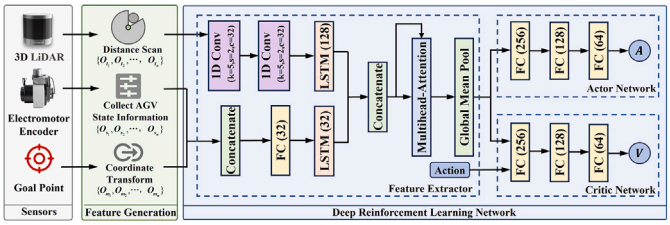


Fig. 5. Deep reinforcement learning network architecture.

where the constraint threshold ϵ specifies the maximum allowable violation value. To implement this, we use the Lagrangian relaxation method to convert the constrained optimization into an unconstrained saddle point problem:

$$\mathcal{L}(\theta,\lambda)=\underset{\tau\sim\pi_\theta}{\mathbb{E}}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right]-\lambda\left(J^{\mathrm{CBF}}(\theta)-\epsilon\right), \tag{21}$$

where $\lambda \geq 0$ is the Lagrange multiplier that penalizes constraint violations, and $J^{\mathrm{CBF}}(\theta)=\mathbb{E}_{s,a\sim\pi_\theta}[\mathcal{V}(s,a)]$ represents the expected CBF violation degree. The optimization process is carried out via alternating updates between the primal and dual variables. Initially, we fix λ and optimize θ using the PPO objective combined with the safety penalty term:

$$\begin{aligned}\theta_{k+1}&=\arg\max_\theta\left\{L^{\mathrm{PPO}}(\theta)-\lambda J^{\mathrm{CBF}}(\theta)\right\},\\ L^{\mathrm{PPO}}(\theta)&=\mathbb{E}_{\pi_{\theta_k}}\left[\min\left(r_t(\theta)\hat{A}_t,\bar{r}_t(\theta)\hat{A}_t\right)\right],\end{aligned} \tag{22}$$

where $r_t(\theta)=\pi_\theta(a_t|s_t)/\pi_{\theta_k}(a_t|s_t)$ is the probability ratio, $\hat{A}_t$ is the advantage function estimate obtained via generalized advantage estimation (GAE) [30], and $\bar{r}_t(\theta)=\mathrm{clip}(r_t(\theta),1-\epsilon_c,1+\epsilon_c)$ is the clipped probability ratio. Based on the updated policy θ_{k+1}, we adjust λ using gradient ascent:

$$\lambda_{k+1}=\max\left(0,\lambda_k+\alpha_\lambda\left(J^{\mathrm{CBF}}(\theta_{k+1})-\epsilon\right)\right), \tag{23}$$

where $\alpha_\lambda>0$ is the dual learning rate. This update mechanism increases λ when the safety constraint is violated to penalize unsafe behavior, and decreases λ when the policy becomes overly conservative, thus dynamically balancing task performance and safety. Furthermore, the parameter update process of the value network remains consistent with the original PPO throughout the training process. The Algorithm 1 summarizes the complete training process.

3.4. DRL network framework

To integrate heterogeneous multi-source information in the observation space, we enhanced the neural network architecture. Traditional multi-layer perceptrons struggle with modeling the coupling between sensor and state data. Thus, we introduce a 1D convolutional layer to capture local sequential patterns, specifically processing omnidirectional distance sequences to extract relevant features.

To enhance the model's ability to capture spatial and temporal dependencies, we further introduced Long Short-Term Memory (LSTM) [31] and a multi-head attention mechanism [32]. The LSTM mitigates issues of vanishing and exploding gradients, enhancing the model's ability to handle long-range temporal dependencies. The attention module focuses on key spatial information by performing parallel representation learning across subspaces. The full sequence of LSTM hidden states is fed into the attention module, preserving temporal information and preventing the loss of long-term dependencies. The multi-head design helps model interactions across different spatial locations and semantic patterns. Fig. 5 illustrates the deep reinforcement learning architecture used for trajectory planning.

3.5. Action safety filter

Although a CBF loss term is introduced during training to guide the policy learning, the outputs of the trained neural network still do not provide strict safety guarantees. Considering the potential significant economic losses and safety risks to personnel that could result from collisions in AGV transfer scenarios, we designed an online safety filtering mechanism to adjust the control commands output by the policy network in real-time, ensuring that the AGV's state is always constrained within a predefined safe set.

Using the CBF constraint defined in Eq. (10), the forms of the barrier function and its time derivative are substituted, transforming the constraint into a linear inequality in terms of the corrected control inputs $\mathbf{u}^*=[a^*,\omega^*]^T$:

$$A\cdot a^*+B\cdot\omega^*\geq -C-c_1\dot{h}-c_2h. \tag{24}$$

The goal is to minimize the correction to the original control inputs $\mathbf{u}=[a,\omega]^T$ while satisfying the constraint:

$$\begin{aligned}\min_{\mathbf{u}^*}\quad & J=\|a^*-a\|^2+\|\omega^*-\omega\|^2\\ \text{s.t.}\quad & a_{\min}\leq a^*\leq a_{\max},\ \omega_{\min}\leq\omega^*\leq\omega_{\max},\\ & A\cdot a^*+B\cdot\omega^*\geq -C-c_1\dot{h}-c_2h.\end{aligned} \tag{25}$$

Since the objective function is convex and quadratic, and the constraint is a linear inequality, this problem is a quadratic programming problem with a global optimal solution, which can be efficiently solved in real-time within each control cycle using numerical algorithms. If the quadratic program is infeasible or cannot be solved within the prescribed time limit, the AGV applies maximum braking until it comes to a complete stop. It is important to note that this safety filtering mechanism is only activated during the testing and deployment phases, and is not activated during the training phase.

4. Experiments

4.1. Experimental setup

To verify the feasibility and effectiveness of the proposed trajectory planning method, we conducted multi-scenario tests in a simulation environment and verified the effect in actual scenarios.

4.1.1. Experimental platform

To validate the proposed trajectory planning framework, simulation experiments were conducted on a computing platform equipped with an Intel i7-13700H CPU and an NVIDIA RTX 4060 GPU. The simulation environment was built upon CoppeliaSim [33] for both model training and performance evaluation. The physical platform parameters are consistent with those of the simulation model, which is shown in Fig. 6. The AGV adopts a 4WIS configuration, with its key geometric parameters, kinematic parameters, and planning algorithm settings summarized in Table 1. The action safety filter was solved using the IPOPT [34] solver.

4.1.2. Scenario setup

Two categories of structured road scenarios were designed for evaluation: straight-road scenarios and intersection scenarios, denoted as Env1 and Env2, respectively. The schematic layouts and corresponding representative CoppeliaSim simulation scenes of the two environments are presented in Fig. 7. These scenarios can be generalized to a wide range of real-world operating conditions through affine transformations. The road width was set to 8 m, a standard value in a yard environment, to evaluate the algorithm's navigation and obstacle avoidance capabilities. In each scenario, the AGV's initial pose was fully randomized, and 0 to 3 obstacles with radii randomly sampled from 0.3 m to 1 m were placed between the start region and the goal region to increase the diversity of training conditions. After each randomization, the Hybrid A* algorithm was employed to quickly verify the existence of a feasible path. If no feasible path was found, the process was re-initialized. During training, the two environments were sampled with equal probability.

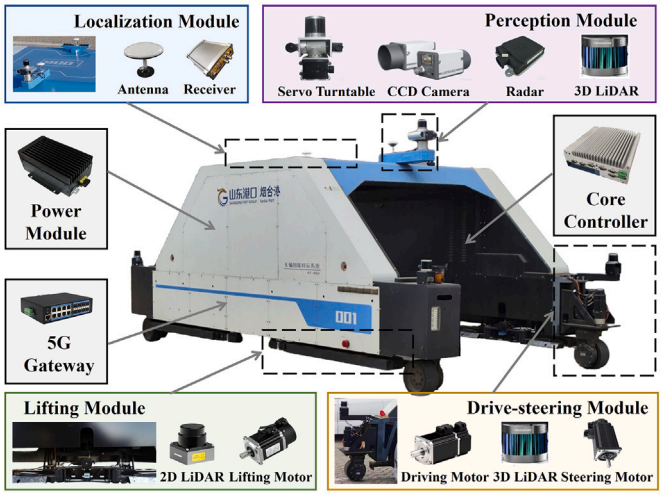


Fig. 6. Illustration of the AGV system physical platform.

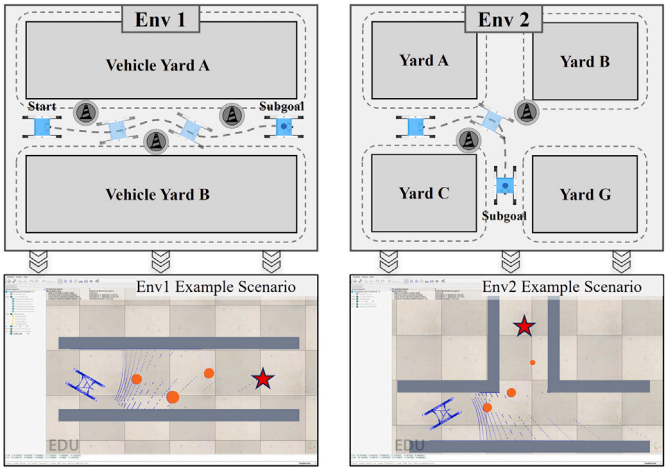


Fig. 7. Trajectory planning evaluation scenarios and corresponding CoppeliaSim scenes.

Table 1
AGV parameters and algorithm parameters settings.

Parameter	Value	Parameter	Value
1. AGV physical parameters			
AGV width (w)	3000 mm	AGV length (l)	6000 mm
Track width ($2W$)	2570 mm	Wheelbase ($2L$)	5438 mm
Max velocity ($v_{\max}$)	2 m/s	Max steering angle ($\eta_{\max}$)	45°
Max acceleration ($a_{\max}$)	5 m/s^2	Max steering rate ($\dot{\eta}_{\max}$)	360°/s
2. Algorithm related parameters			
Distance coefficient (α_1)	0.1	Angular error coefficient (α_2)	1
Arrival time coefficient (α_3)	1	CBF coefficient (c_1)	10
CBF coefficient (c_2)	10	Safe distance (D_{safe})	0.1 m
3. DRL training parameters			
Network learning rate	1e−4	Multiplier learning rate (α_λ)	2e−6
Mini batch size	256	Max episode steps	1024
GAE parameter	0.98	Discount factor (γ)	0.99

4.1.3. Baselines and metrics

In the training analysis, we compare the proposed CBF-PPO with PPO [23], SAC [35] and TD3 [36]. These methods are representative deep reinforcement learning algorithms for online continuous control tasks. To ensure a fair comparison, all learning-based methods share the same observation space, action space, reward design, network architecture, training scenarios, and termination conditions.

In addition, in the overall comparison experiments, we further introduce three traditional methods: A* [5], Hybrid A* [6], and safe corridor method [22]. A* and Hybrid A* are essentially path planning methods that output discrete waypoints. Therefore, we use a pure pursuit controller to handle the steering control and accelerate the vehicle to its maximum speed as quickly as possible for speed control. For the safe corridor method, since it directly generates a reference trajectory, we construct an MPC tracker to track this trajectory. In the overall comparison experiments, we use success rate (SR), collision rate (CR), travel time, terminal heading error, average trajectory curvature, and maximum trajectory curvature as evaluation metrics. These metrics provide a comprehensive and thorough assessment of the quality of the generated trajectories.

4.2. Training analysis

Fig. 8 presents the episodic cumulative reward curves of the CBF-PPO algorithm and three baseline RL algorithms evaluated under identical mixed scenarios. All experiments were conducted with a uniform training duration of 10 h. In the figure, bold solid lines represent

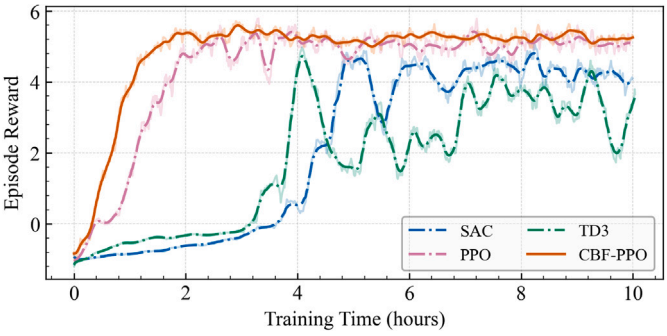


Fig. 8. Training reward curves of different RL algorithms.

the smoothed reward trends, while semi-transparent shaded regions indicate the raw data observed throughout the training process.

The experimental results demonstrate that incorporating control barrier function constraints significantly enhances policy learning efficiency. CBF-PPO exhibits a notably rapid performance improvement in the early stages of training, converging from an initially negative reward range to a high-reward regime within approximately 1.7 h, a convergence rate substantially superior to all baseline methods. By contrast, the standard PPO algorithm achieves stable convergence but with a comparatively slower ascent rate. The off-policy methods, SAC and TD3, exhibit markedly sluggish progress in the early training phase, revealing pronounced exploration bottlenecks.

Regarding final convergence performance, both CBF-PPO and PPO attain high asymptotic reward levels, with CBF-PPO marginally outperforming standard PPO. Notably, while SAC and TD3 eventually undergo rapid performance surges after an extended low-performance period, these gains are accompanied by severe reward oscillations following the peak, indicating poor training robustness. These observations suggest that the integration of CBF constraints provides effective safety-aware guidance during policy exploration, enabling the agent to avoid hazardous states early in training and concentrate exploration on high-return regions, thereby substantially accelerating the accumulation of informative experience. Concurrently, the CBF constraints ensure the stability of the learned policy under prolonged training conditions.

4.3. Comparison experiment

To comprehensively evaluate the performance of the proposed method, we conducted systematic comparative experiments between

Table 2
Performance comparison of different methods across scenarios.

Scenario	Method	Performance metrics					
		SR (%)	CR (%)	Travel time (s)	Heading error (°)	Avg. curvature (m^{-1})	Max. curvature (m^{-1})
Env1	A*	73	27	15.35	14.32	0.2212	0.3673
	Hybrid A*	90	10	15.92	4.344	0.0951	**0.2122**
	Safe Corridor	96	0	16.72	**1.784**	0.1329	0.3159
	CBF-PPO (Ours)	**100**	**0**	**15.05**	2.562	**0.0809**	0.2782
Env2	A*	68	32	14.95	31.88	0.2638	0.3673
	Hybrid A*	86	14	15.44	8.789	0.1268	**0.2822**
	Safe Corridor	92	0	16.32	**2.082**	0.1427	0.3463
	CBF-PPO (Ours)	**100**	**0**	**14.75**	3.264	**0.1241**	0.3258

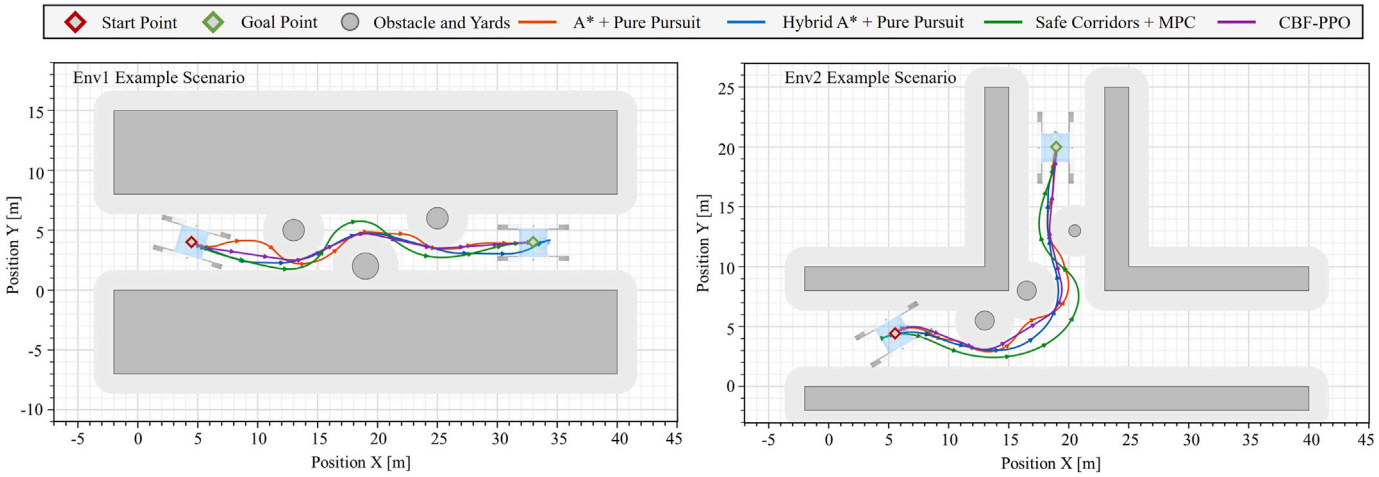


Fig. 9. Visualization of trajectories from various algorithms across example experimental scenarios.

several baseline algorithms and the CBF-PPO framework in highly randomized scenarios. For each scenario category, 100 randomized test cases were generated, all of which were pre-validated to ensure the existence of feasible solutions. The mean values of key performance indicators across these diverse environments are summarized in Table 2.

Quantitative results indicate that the conventional A* algorithm exhibits pronounced limitations across all primary metrics. Due to its inherent 2D grid-based search nature, A* faces an irreconcilable trade-off between geometric obstacle avoidance and search completeness. Experiments show that using the overall circumscribed circle expansion leads to no solution in most restricted scenarios. Meanwhile, the compromise of using a double-circle model with local radius expansion results in the highest collision rate due to underestimating the physical space occupied. Furthermore, the absence of explicit kinematic modeling results in high path curvature and significant terminal heading errors, making it ill-suited for high-precision execution tasks.

While Hybrid A* achieves a qualitative leap in path smoothness by integrating kinematic feasibility into the search process, its discrete graph-search nature still yields a degree of motion redundancy. Crucially, the collisions observed during the execution phase suggest that its detection mechanism, which relies on geometric boundaries, lacks sufficient safety margins when subjected to execution perturbations. In contrast, the safe corridor method ensures high safety via explicitly constructed convex free spaces. However, this defensive planning strategy sacrifices spatio-temporal efficiency, resulting in longer travel times. CBF-PPO achieves optimal motion efficiency and a 100% success rate while maintaining a zero-collision record, demonstrating superior comprehensive performance.

The trajectory visualizations shown in Fig. 9 further reveal the mechanistic disparities in the underlying modeling logic of each algorithm. The failure of A* stems from its low-dimensional state space and topological discreteness, which causes motion redundancy and terminal pose misalignment due to the neglect of non-holonomic constraints. While Hybrid A* produces more structured trajectories, the inherent nature of discrete control primitives limits its ability to achieve global optimality within a continuous state space.

The trajectories generated by the Safe Corridor method, though smooth and continuous, tend to maintain excessive clearance from obstacles, reflecting how the conservative spatial margins configured during corridor construction constrain path optimality. Conversely, the CBF-PPO trajectories strike a sophisticated balance between fluid obstacle avoidance and high-efficiency directivity. These results indicate that CBF-PPO can effectively safeguard safety boundaries without compromising planning degrees of freedom, realizing a synergistic improvement in both operational efficiency and execution precision.

4.4. Parameter sensitivity analysis

To further evaluate the performance and robustness of the proposed framework, parameter sensitivity analyses were conducted on the distance reward coefficient α_1 and the arrival reward coefficient α_3. These evaluations were performed in the more challenging Env2 scenario. Given that the absolute values of reward functions in reinforcement learning are less significant than the reward margins between different states in driving the policy update, we fixed $\alpha_2 = 1$ as a baseline. A control variable method was employed to assess the impact of individual parameter variations on system performance, and the results are shown in Fig. 10.

Theoretically, the distance reward associated with α_1 functions as a dense reward, primarily designed to provide continuous guidance signals during the early stages of training, thereby accelerating the agent's progress toward the target. Consequently, increasing α_1 is expected to enhance the convergence rate. However, since α_1 acts as a positive incentive for path length, it inherently competes with the objective of mission time minimization. Our results indicate that a moderate increase in α_1 strengthens the dense progress guidance,

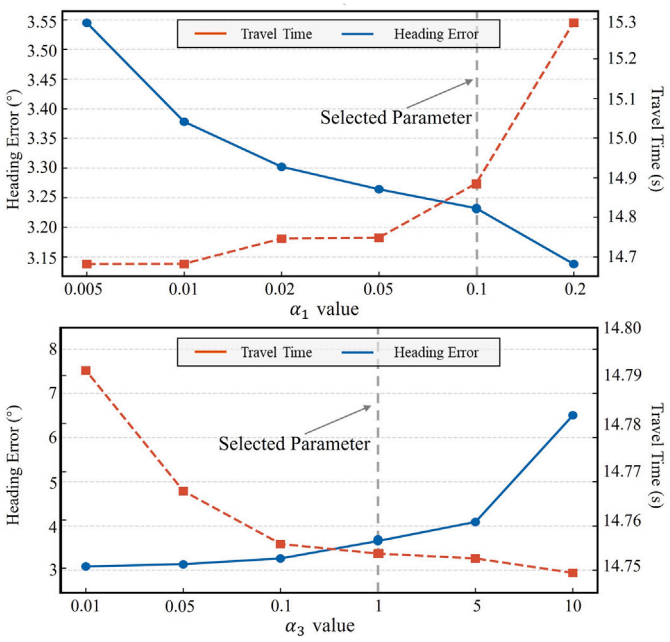


Fig. 10. Sensitivity analysis of key parameters.

Table 3
Ablation study results in typical scenarios.

Method	Performance metrics			
	SR (%)	CR (%)	Travel time (s)	Conv. time (h)
Vanilla PPO	94	6	14.83	2.8
w/o CBF-training	100	0	14.94	2.7
w/o Filter	96	4	**14.68**	1.7
CBF-PPO	**100**	**0**	14.75	**1.7**

indirectly promoting terminal orientation alignment and thereby reducing terminal pose error. Nevertheless, when α_1 exceeds a critical threshold, the temporal performance of the system degrades significantly. When the magnitude of the cumulative distance reward overwhelms the time penalty term (e.g., at $\alpha_1 = 0.5$), the algorithm suffers from severe convergence difficulties and performance collapse, characterized by the vehicle oscillating or spinning in place. Therefore, the selection criterion for α_1 is to maximize training efficiency while ensuring it does not undermine the dominant role of the terminal reward components.

As α_3 increases from 0.01 to 10, the terminal pose error exhibits a clear monotonic increasing trend. This confirms that a higher arrival reward effectively directs the agent's optimization priority toward precise pose alignment at the end of the trajectory. However, it is important to note that the time term and orientation term within the terminal reward are inherently mutually exclusive, necessitating a strategic trade-off. The sensitivity analysis demonstrates that the current parameter configuration achieves a robust balance between these conflicting objectives, maintaining high operational efficiency while ensuring superior precision.

4.5. Ablation experiment

To assess the individual contribution of each component within the CBF-PPO framework, we conducted an ablation study. The results presented in Table 3 systematically evaluate the role of each core module under typical navigation scenarios. We also select Env2 as the test environment and compare four configurations: the PPO model, the full CBF-PPO model, a variant without CBF-based action filtering (w/o Filter), and a variant without CBF-incorporated training (w/o CBF-Training).

Vanilla PPO, which serves as the baseline, exhibits the highest collision rate and the longest convergence time, reflecting the inherent limitations of unconstrained policy optimization in safety-critical environments. Since the action filter is only activated during the testing phase, the w/o CBF-Training variant is equivalent to vanilla PPO during training, and the marginal difference in convergence time between the two falls within the stochastic variation intrinsic to reinforcement learning. At test time, however, integrating the CBF-based action filter eliminates all collisions, confirming that the filter acts as a deterministic safety barrier capable of guaranteeing safe deployment regardless of the underlying policy quality. Incorporating CBF constraints into the policy gradient optimization substantially accelerates convergence and marginally improves travel time, as the CBF gradient term guides the agent to perceive safety boundaries and avoid futile exploration in high-risk regions. Nevertheless, the w/o Filter variant demonstrates that soft constraints applied solely during training still yield a residual collision rate of 4%, failing to cover all edge cases in complex environments. The complete CBF-PPO framework achieves strict safety guarantees at the cost of only a slight increase in travel time, demonstrating that safety and motion efficiency are not mutually exclusive and that both components are jointly essential to the framework's effectiveness.

Furthermore, the computational efficiency of the proposed framework was evaluated and analyzed across the 200 simulation trials. The average inference time per step for the CBF-PPO policy network was 0.8785 ms, with a recorded maximum of 2.6440 ms. The subsequent action filtering stage, encompassing optimization problem formulation and quadratic programming solving, required an average execution time of 0.0624 ms (with a peak of 0.1673 ms). Consequently, the total computational latency per control step remains consistently below 3 ms. This performance underscores the superior real-time responsiveness of the proposed method.

4.6. Application deployment

To evaluate the practical performance of the proposed algorithm, field experiments were conducted in a representative vehicle yard at Yantai Port. The experiments encompassed two typical tasks: vehicle pickup and dropoff. Initially, the AGV approached the operational area via the peripheral road and executed a zero-turn maneuver, enabled by its 4WIS system, to enter the working zone. After completing the pickup task, the robot exited the yard and proceeded to the interaction area for unloading. While the AGV tracked pre-planned trajectories in clear zones, two regions with unknown obstacles were introduced to test collision avoidance. Notably, a stationary AGV with identical parameters was positioned in area 2 to simulate a vehicle breakdown scenario during operations. The overall experimental layout and executed trajectories are illustrated in Fig. 11. Field experiments demonstrate that the proposed algorithm successfully avoids all placed obstacles and completes the full operational procedure safely.

5. Conclusion

This paper presents a real-time AGV trajectory planning framework that combines control barrier functions with proximal policy optimization, aiming to ensure the safe and efficient operation of transfer AGVs in RORO terminal tasks. The framework does not require the independent tracking modules typically used in traditional planning methods and can directly output control commands. By constructing a control barrier function based on the relative position information of the nearest obstacle point to the AGV and incorporating it into the gradient optimization process of PPO, the agent is guided to quickly move away from danger zones, accelerating policy convergence. After the network outputs the initial control commands, a control barrier function is further used to construct a command filter to ensure operational safety. Experimental results show that the framework exhibits high planning performance and real-time responsiveness, demonstrating its potential

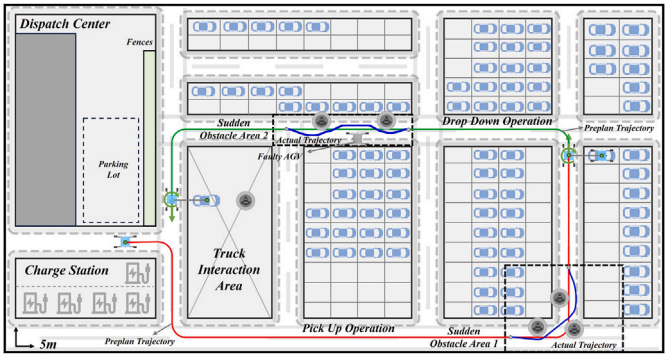


Fig. 11. Schematic of the field experiment layout and operations.

to replace traditional two-stage planning methods in structured port environments.

Future work will focus on two directions. The current framework is primarily designed for static obstacles. However, it is capable of being extended to dynamic scenarios. Future plans include incorporating a trajectory prediction module or reciprocal velocity obstacle methods to enable the system to predict and avoid moving obstacles. Additionally, the current framework focuses on single-agent planning, but future extensions will involve multi-agent reinforcement learning techniques and the integration of graph neural networks to model the state interactions between vehicles, supporting efficient collective decision-making in multi-AGV collaborative scheduling scenarios.

CRediT authorship contribution statement

Runjiao Bao: Writing – original draft. **Shoukun Wang:** Project administration. **Yongkang Xu:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62473044.

Data availability

Data will be made available on request.

References

[1] S. Deniz, Y. Wu, Y. Shi, Z. Wang, A reinforcement learning approach to vehicle coordination for structured advanced air mobility, Green Energy Intell. Transp. 3 (2) (2024) 100157.
[2] Z. Zong, T. Feng, J. Wang, T. Xia, Y. Li, Deep reinforcement learning for demand-driven services in logistics and transportation systems: A survey, ACM Trans. Knowl. Discov. from Data 19 (4) (2025) 1–42.
[3] M. Chen, Z. Liang, Z. Cheng, J. Zhao, Z. Tian, Optimal scheduling of FTPSS with PV and HESS considering the online degradation of battery capacity, IEEE Trans. Transp. Electrification 8 (1) (2022) 936–947.
[4] Y. Xu, R. Bao, L. Zhang, J. Wang, S. Wang, Embodied intelligence in RO/RO logistic terminal: Autonomous intelligent transportation robot architecture, Sci. China Inf. Sci. 68 (5) (2025) 1–17.
[5] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (2) (1968) 100–107.
[6] D. Dolgov, S. Thrun, M. Montemerlo, J. Diebel, Practical search techniques in path planning for autonomous driving, AAAI Work. - Tech. Rep. (2008).
[7] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robot. Res. 30 (7) (2011) 846–894.
[8] B. Li, K. Wang, Z. Shao, Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach, IEEE Trans. Intell. Transp. Syst. 17 (11) (2016) 3263–3274.
[9] J. Chen, C. Liu, M. Tomizuka, FOAD: Fast optimization-based autonomous driving motion planner, in: 2018 Annual American Control Conference, ACC, 2018, pp. 4725–4732.
[10] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.
[11] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, Two-stage trajectory optimization for autonomous ground vehicles parking maneuver, IEEE Trans. Ind. Informatics 15 (7) (2019) 3899–3909.
[12] Z. Xing, R. Chai, K. Chen, Y. Xia, S. Chai, Online trajectory planning method for autonomous ground vehicles confronting sudden and moving obstacles based on LSTM-attention network, IEEE Trans. Cybern. 55 (1) (2025) 421–435.
[13] R. Bao, Y. Xu, C. Wang, T. Niu, J. Wang, S. Wang, STGN: A spatio-temporal graph network for real-time and generalizable trajectory planning, IEEE Trans. Autom. Sci. Eng. (2025).
[14] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, S. Chai, Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver, IEEE Trans. Autom. Sci. Eng. 20 (3) (2023) 1633–1647.
[15] L. Gao, T. Ma, K. Liu, Y. Zhang, Application of improved Q-learning algorithm in path planning, J. Jilin Univ. (Information Sci. Edition) 36 (4) (2018) 439.
[16] Z. Xie, P. Dames, DRL-VO: Learning to navigate through crowded dynamic scenes using velocity obstacles, IEEE Trans. Robot. 39 (4) (2023) 2700–2719.
[17] H. Yang, C. Yao, C. Liu, Q. Chen, RMRL: Robot navigation in crowd environments with risk map-based deep reinforcement learning, IEEE Robot. Autom. Lett. 8 (12) (2023) 7930–7937.
[18] R. Bao, Y. Xu, L. Zhang, H. Yuan, J. Si, S. Wang, T. Niu, Deep reinforcement learning-based trajectory tracking framework for 4WS robots considering switch of steering modes, in: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2025, pp. 3792–3799.
[19] R. Bao, J. Wang, S. Wang, Geodesic-based path planning for port transfer robots on Riemannian manifolds, Expert Syst. Appl. (2025) 129706.
[20] V. Arvind, Optimizing the turning radius of a vehicle using symmetric four wheel steering system, Int. J. Sci. Eng. Res. 4 (12) (2013) 2177–2184.
[21] X. Liu, W. Wang, X. Li, F. Liu, Z. He, Y. Yao, H. Ruan, T. Zhang, MPC-based high-speed trajectory tracking for 4WIS robot, ISA Trans. 123 (2022) 413–424.
[22] B. Li, T. Acarman, Y. Zhang, Y. Ouyang, C. Yaman, Q. Kong, X. Zhong, X. Peng, Optimization-based trajectory planning for autonomous parking with irregularly placed obstacles: A lightweight iterative framework, IEEE Trans. Intell. Transp. Syst. 23 (8) (2021) 11970–11981.
[23] Y. Gu, Y. Cheng, C.L.P. Chen, X. Wang, Proximal policy optimization with policy feedback, IEEE Trans. Syst. Man, Cybern.: Syst. 52 (7) (2022) 4600–4610.
[24] W. Paszkowiak, T. Bartkowiak, M. Pelic, Kinematic model of a logistic train with a double ackermann steering system, Int. J. Simul. Model. 20 (2) (2021) 243–254.
[25] N.T. Nguyen, P. Tej Gangavarapu, N. Mandel, R. Bruder, F. Ernst, Motion planning for 4WS vehicle with autonomous selection of steering modes via an MIQP-MPC controller, in: 2024 IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 9765–9771.
[26] M.-h. Lee, T.-H.S. Li, Kinematics, dynamics and control design of 4WIS4WID mobile robots, J. Eng. 2015 (1) (2015) 6–16.
[27] X. Zhang, A. Liniger, F. Borrelli, Optimization-based collision avoidance, IEEE Trans. Control Syst. Technol. 29 (3) (2021) 972–983.
[28] J. Lian, W. Ren, D. Yang, L. Li, F. Yu, Trajectory planning for autonomous valet parking in narrow environments with enhanced hybrid a* search and nonlinear optimization, IEEE Trans. Intell. Veh. 8 (6) (2023) 3723–3734.
[29] Z. Han, Y. Wu, T. Li, L. Zhang, L. Pei, L. Xu, C. Li, C. Ma, C. Xu, S. Shen, et al., An efficient spatial-temporal trajectory planner for autonomous vehicles in unstructured environments, IEEE Trans. Intell. Transp. Syst. 25 (2) (2023) 1797–1814.
[30] X. Zhang, Y. Ding, H. Zhao, L. Yi, T. Guo, A. Li, Y. Zou, Mixed skewness probability modeling and extreme value predicting for physical system input–output based on full Bayesian generalized maximum-likelihood estimation, IEEE Trans. Instrum. Meas. 73 (2024) 1–16.
[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[33] E. Rohmer, S.P.N. Singh, M. Freese, V-REP: A versatile and scalable robot simulation framework, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 1321–1326.
[34] A. Wächter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program. 106 (2006) 25–57.
[35] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International Conference on Machine Learning, PMLR, 2018, pp. 1861–1870.
[36] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International Conference on Machine Learning, PMLR, 2018, pp. 1587–1596.