Robust Warehouse AGV Navigation in Dynamic Environments Using Soft Actor-Critic Reinforcement Learning

Paul J. Jiaway

Computer Science and Technology, Anhui University of Technology

DOI: https://doi.org/10.51244/IJRSI.2026.1306000017

Received: 31 May 2026; Accepted: 05 June 2026; Published: 18 June 2026

ABSTRACT

Autonomous guided vehicles (AGVs) operating in modern warehouse environments face the critical challenge of navigating safely and efficiently amid high-density dynamic obstacles, including human workers and peer vehicles. Classical reactive planners such as the Dynamic Window Approach (DWA) fail under these conditions due to their inability to predict obstacle motion, while conventional deep reinforcement learning methods often overfit to training layouts and lack the behavioral diversity needed for robust generalization. This paper presents a Soft Actor-Critic (SAC) based navigation framework that addresses these limitations through maximum-entropy reinforcement learning combined with a multicomponent reward design and domain-randomized training. The proposed method jointly optimizes goalreaching efficiency, collision avoidance, energy consumption, and trajectory smoothness within a unified learning objective. We evaluate the approach against three strong baselines—DDPG, TD3, and DWA— across four structurally distinct warehouse layouts, three of which are unseen during training. Experimental results demonstrate that SAC achieves the highest mean cumulative reward (99.32 vs. 82.18 for DDPG, 88.58 for TD3, and 55.41 for DWA), the lowest collision rate (1.5 TD3), and the shortest paths (15.54 m average vs. 22.17 m for DDPG and 25.54 m for TD3). Notably, SAC exhibits near-invariant path lengths across all layouts with a cross-layout standard deviation of only 0.22 m, providing compelling evidence of geometry-agnostic navigation. Energy consumption is approximately half that of DDPG and TD3, while trajectory smoothness is improved by a factor of two. These results establish that entropy-regularized deep reinforcement learning, coupled with principled reward shaping and domain randomization, produces warehouse navigation policies that are substantially more robust, efficient, and deployment-ready than both classical planners and deterministic actor-critic alternatives.

Keywords: Soft Actor-Critic, autonomous guided vehicle navigation, dynamic obstacle avoidance, domain randomization, maximum-entropy reinforcement learning, warehouse robotics, energy-efficient navigation

INTRODUCTION

Autonomous mobile robots have been deployed in warehouse environments for approximately two decades, but adoption has accelerated dramatically since 2012 following Amazon's acquisition of Kiva Systems and the subsequent commercialization of autonomous mobile robot platforms by competitors including Geek+, 6 River Systems, and Fetch Robotics (Guizzo 2008) (Lackner et al. 2024). Modern fulfillment centers may operate hundreds of vehicles simultaneously across spaces exceeding 100,000 square meters, coordinating picking, transport, and replenishment tasks. Under such conditions, navigation reliability becomes operationally critical: vehicles that stop unnecessarily, reroute inefficiently, or collide with pedestrians directly reduce facility throughput and increase operational costs. Industry reports estimate that unplanned stops account for 12–18% of total downtime in some large-scale deployments (Fragapane et al. 2021)

Automated guided vehicles (AGVs) differ fundamentally from autonomous passenger vehicles in ways that directly influence navigation algorithm design. Their operational domain is bounded and, in principle, mappable in advance; floor layouts change more slowly than urban road networks; and maximum speeds are typically low, rarely exceeding 2.0 m/s for load-bearing platforms. These properties favor approaches that construct detailed prior maps and leverage known infrastructure, which explains the continued prevalence of layered navigation architectures pairing a global path planner (A* or Dijkstra on an occupancy grid) with a local trajectory optimizer

such as DWA or the Timed Elastic Band (TEB) planner (LaValle 2006)(Rösmann et al. 2012). However, this architecture does not handle gracefully the interaction between the vehicle and agents whose behavior cannot be predicted from a static map. Human workers in fulfillment centers do not follow programmed paths: they stop to inspect items, reverse direction without signaling, and cluster near popular pick locations. Peer AGVs nominally follow coordination protocols but may deviate due to their own obstacle encounters. The practical consequence is that a vehicle executing a DWA trajectory computed at timestep may encounter an obstacle at that was not present at. If is small relative to DWA's prediction horizon, the vehicle replans in time; otherwise, it either brakes sharply, stressing mechanical components, or proceeds and collides.

Reinforcement learning (RL) addresses this problem from a fundamentally different perspective. Rather than building an explicit model of the environment and optimizing paths through it, an RL agent learns a policy that maps observations to actions through repeated trialand-error interaction. The policy captures implicit knowledge about how obstacles tend to behave, how much clearance is typically needed when passing shelving units, and when it is better to slow down than attempt a gap that may close. This implicit knowledge is difficult to encode in rule-based systems, where designers must enumerate cases that emerge naturally from training data accumulated over millions of simulation steps.

Despite the promise of deep RL for robot navigation, several practical challenges have limited its deployment in commercial warehouse contexts. Policies trained in fixed simulation environments tend to overfit to specific obstacle densities, layouts, and dynamics, failing to transfer to new settings (Zhao et al. 2020). Reward functions that do not explicitly account for energy consumption produce controllers that reach goals successfully but follow unnecessarily aggressive trajectories, increasing battery drain and reducing vehicle longevity. Methods lacking explicit entropy regularization, such as DDPG, often converge to deterministic policies that perform well in training but lack the behavioral diversity needed for novel situations (Schulman et al. 2017).

Soft Actor-Critic (SAC) (Haarnoja et al. 2018) addresses the entropy deficiency by optimizing a maximum-entropy objective that encourages the policy to be as stochastic as possible while still achieving high expected reward. Empirically, SAC policies trained with automatic temperature tuning demonstrate greater robustness to distributional shift than DDPG or TD3 policies trained on the same data, a finding replicated across locomotion, manipulation, and navigation tasks. This paper investigates whether the same advantage holds in the warehouse navigation setting, where distributional shift arises from day-to-day variability of human and vehicle traffic patterns. We make the following contributions: (1) we demonstrate that SAC achieves superior composite navigation performance compared to DDPG, TD3, and DWA across both training and unseen warehouse layouts; (2) we design and validate a multi-component reward function that jointly optimizes goal-reaching efficiency, collision avoidance, energy consumption, and velocity smoothness; (3) we show that domain randomization applied to obstacle density, pedestrian velocity, and sensor noise produces a controller that maintains consistent performance across structurally distinct layouts not seen during training; and (4) we provide a comprehensive experimental characterization of SAC's advantages, including near-invariant path lengths across diverse geometries, providing strong evidence of geometry-agnostic navigation capability.

RELATED WORK

Classical Navigation Methods

Path planning for mobile robots rests on a welldeveloped theoretical foundation built upon graph search, potential field methods, and sampling-based planners. Dijkstra's algorithm (Dijkstra 1959) computes shortest paths on weighted graphs and, when applied to occupancy grid representations, produces globally optimal plans under the completeness and accuracy of the map. A* extends Dijkstra with a heuristic function that guides search toward the goal, reducing computational cost (Hart et al. 1968) . Both algorithms operate on static snapshots and require replanning when the map changes. The Rapidly-exploring Random Tree (RRT) family addresses kinodynamic constraints more naturally by sampling from the full configuration space (LaValle 1998), with RRT* adding asymptotically optimal rewiring (Karaman and Frazzoli 2011). DWA, introduced by Fox, Burgard, and Thrun (Fox et al. 1997), remains the dominant local planner in commercial deployments due to its computational efficiency and interpretable parameter space. The Timed Elastic Band planner improves on DWA

by optimizing over a sequence of poses connected by time-parameterized elastic bands, enabling smoother trajectories around obstacles (Rösmann et al. 2012). Neither method, however, was designed for environments in which obstacles move in complex, socially constrained patterns. Model Predictive Control (MPC) formulations have attracted significant attention as alternatives to sampling-based approaches (Jian et al. 2022), but their performance depends critically on the accuracy of obstacle prediction models, which remains an open problem for heterogeneous traffic combining human pedestrians and autonomous vehicles (Alahi et al. 2016).

Reinforcement Learning for Robot Navigation

The application of reinforcement learning to robotics problems predates the deep learning era. Sutton's policy gradient theorem (Sutton et al. 1999) and Watkins's Q-learning algorithm (Watkins and Dayan 1992) provided theoretical foundations for subsequent deep RL methods. Tai et al. (Tai et al. 2017) demonstrated that DDPG could train a reactive collision avoidance controller using raw LiDAR inputs, while subsequent work showed that policy gradient methods could navigate complex environments without explicit maps (Zhu et al. 2025)(Waga et al. 2025).

DQN (Mnih et al. 2015) demonstrated that convolutional neural networks trained with experience replay could learn human-level control policies from raw pixels, but its discrete action space makes it unsuitable for continuouscontrol robotics. DDPG (Lillicrap et al. 2015) extended the actor-critic framework to continuous action spaces using a deterministic policy and Q-value critic. Proximal Policy Optimization (PPO) (Schulman et al. 2017) has been widely applied in robotics due to its stability properties, though its on policy nature requires substantially more environment interactions than off-policy methods to reach equivalent performance (Duan et al. 2016). Twin Delayed DDPG (TD3) (Fujimoto et al. 2018) addresses DDPG's overestimation bias and gradient sensitivity through twin Q-function networks and delayed policy updates.

Soft Actor-Critic and Domain Randomization

Haarnoja et al. (Haarnoja et al. 2018) introduced SAC as an off-policy algorithm maximizing a maximum-entropy objective rather than conventional cumulative reward. The key insight is that entropy regularization discourages premature commitment to a single strategy, which is especially valuable in environments where multiple adequate behaviors exist. In navigation around obstacles, an entropy-regularized policy distributes probability over the full set of approximately equivalent avoidance maneuvers rather than converging to a single deterministic behavior that may fail in novel configurations. Subsequent work derived automatic temperature tuning that adjusts the entropy coefficient during training to maintain a target entropy level, removing a difficult hyperparameter and improving robustness across environments (Haarnoja et al. 2019).

Domain randomization, introduced by Tobin et al. (Tobin et al. 2017) for object detection and adapted for robot control by Peng et al. (Peng et al. 2017) and Andrychowicz et al. (Andrychowicz et al. 2018), addresses the simulation-to-reality gap by training across a distribution of simulation parameters rather than a single fixed environment. The hypothesis is that if the policy performs well across the full range of randomized parameters during training, the real environment will lie within this range and the policy will transfer. For navigation specifically, Kadian et al. (Kadian et al. 2019) found that randomizing visual appearance, obstacle density, and sensor noise independently all contributed to transfer performance, but that interaction effects between randomization dimensions are important and should be tested through ablations.

Research Gaps

Three gaps in the existing literature motivate our approach. First, no prior work has directly compared SAC against both TD3 and DWA baselines on a dynamic warehouse navigation benchmark with heterogeneous obstacle types combining human pedestrians and autonomous vehicles. Available comparisons either use simpler environments with a single obstacle category or evaluate only a subset of competing methods. Second, the energy efficiency of deep RL navigation policies is under explored: most evaluations report success and collision rates but do not measure energy consumption, which directly affects vehicle utilization in AGV deployments. Third, the contribution of automatic temperature tuning specifically in navigation tasks has not been evaluated through controlled comparison against deterministic alternatives on a unified benchmark.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

METHODOLOGY

Problem Formulation

The navigation task is formulated as a discrete-time Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}: \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the transition probability function, $\mathcal{R}: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0,1)$ is the discount factor. At each timestep, the agent observes the state, selects an action, receives a scalar reward, and transitions to a new state. The objective is to find a policy $\pi: \mathcal{S} \to \mathcal{P}(\mathcal{A})$ that maximizes the expected cumulative discounted reward. The Markov assumption holds approximately in this setting. The full environment state, including positions and velocities of all obstacles, satisfies the Markov property by definition. The observable state does not include the full environment state, making this technically a Partially Observable MDP (POMDP). However, the LiDAR scan provides dense spatial coverage of the immediate surroundings, and the current velocity state provides one step of temporal information. In practice, the Markov assumption over the observable state is a reasonable approximation consistent with related work.

State Space

The observation vector $\mathbf{o}_t \in \mathbb{R}^{28}$ is constructed at each control timestep t from four components:

1. **LiDAR readings:** A 360° LiDAR sensor with 24 evenly spaced beams produces range measurements l_i for $i = 1, \ldots, 24$. Each reading is normalized to $[0,1]$ by dividing by the maximum sensor range $r_{max} = 10.0\,\text{m}$, then clipped to 1.0 for out-of-range returns:

$$l_i^{norm} = \min\left(\frac{l_i}{r_{max}}, 1.0\right)$$

2. **Goal-relative coordinates:** The goal position is represented in the vehicle's local polar frame as the Euclidean distance d_g (normalized by a reference distance of $d_{ref} = 20.0\,\text{m}$, the diagonal of the environment) and the heading error ϕ_g (the angle from the vehicle's heading to the goal direction, normalized from $[-\pi, \pi]$ to $[-1,1]$):

$$d_g^{norm} = \frac{\| \mathbf{p}_{goal} - \mathbf{p}_t \|}{d_{ref}}$$

$$\phi_g^{norm} = \frac{\text{atan2}(\sin(\phi_g), \cos(\phi_g))}{\pi}$$

3. **Velocity state:** The current linear velocity $v_t \in [-v_{max}, v_{max}]$ and angular velocity $\omega_t \in [-\omega_{max}, \omega_{max}]$ are normalized to $[-1,1]$ by dividing by their respective bounds. This provides the agent with one step of velocity history, sufficient to compute acceleration but not full motion history.

The full observation vector is therefore:

$$\mathbf{o}_t = \left[l_1^{norm}, \ldots, l_{24}^{norm}, d_g^{norm}, \phi_g^{norm}, v_t^{norm}, \omega_t^{norm}\right]^T \in \mathbb{R}^{28}$$

The 24-beam angular resolution of 15° per beam was selected based on the typical width of shelving units (0.8–1.2 m) and human targets (0.5–0.7 m) relative to the vehicle's operating radius. At a range of 3 m, each beam subtends an arc of approximately 0.8 m, sufficient to detect the majority of obstacles in the environment with at least two beam returns.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Action Space

The action space $\mathcal{A} = [-v_{max}, v_{max}] \times [-\omega_{max}, \omega_{max}]$ is continuous, with $v_{max} = 0.5$ m/s and $\omega_{max} = 1.0$ rad/s. The Soft Actor-Critic (SAC) policy outputs a Gaussian distribution over this space; actions are sampled from this distribution during training and taken as the mean during evaluation. This gives:

$$\mathbf{a}_t = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \in \mathbb{R}^2$$

Actions are sent directly to a velocity controller that computes wheel speed commands from the differential-drive kinematics (see Section 4.4). The low-level controller enforces wheel speed limits and rate-of-change constraints independently of the RL agent.

Reward Function

The reward function comprises five components designed to jointly optimize goal-reaching efficiency, collision avoidance, energy consumption, and trajectory smoothness:

$$r_t = r_{goal} + r_{coll} + r_{prog} + r_{energy} + r_{smooth}$$

Goal attainment bonus: A terminal positive reward is given when the vehicle reaches the goal region:

$$r_{goal} = +100 \quad \text{if } \| \mathbf{p}_t - \mathbf{p}_{goal} \| < d_{threshold} = 0.3 \text{ m}$$

Collision penalty: A terminal negative reward is given when the vehicle contacts any obstacle or boundary. The penalty is large enough to dominate the maximum accumulated shaping reward over an episode:

$$r_{coll} = -100 \quad \text{if contact detected}$$

Progress shaping: A continuous dense reward proportional to the reduction in Euclidean distance to the goal:

$$r_{prog} = k_p \cdot (d_{t-1} - d_t)$$

where $d_t = \| \mathbf{p}_t - \mathbf{p}_{goal} \|$ and $k_p = 2.0$. This shaping function is potential-based, satisfying the conditions of Ng et al. (Ng et al. 1999), and therefore does not alter the optimal policy of the base MDP. It provides a gradient signal toward the goal during episodes in which the agent does not reach the goal.

Energy penalty: A quadratic penalty on the control effort:

$$r_{energy} = -k_e \cdot (v_t^2 + \omega_t^2)$$

where $k_e = 0.1$. The weight was selected so that the maximum single-step energy penalty ($-0.1 \cdot (0.5^2 + 1.0^2) = -0.125$) is smaller than the single-step progress reward for useful motion toward a nearby goal, preventing the agent from learning to remain stationary to avoid energy penalties.

Smoothness penalty: A penalty on the magnitude of velocity change between consecutive steps:

$$r_{smooth} = -k_s \cdot (|v_t - v_{t-1}| + |\omega_t - \omega_{t-1}|)$$

where $k_s = 0.05$. This term discourages rapid direction changes and velocity reversals, which cause mechanical wear and passenger discomfort in physical deployments.

Total reward weights were selected through a grid search over 25 combinations of (k_p, k_e, k_s) with $k_p \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$, $k_e \in \{0.05, 0.1, 0.2\}$, $k_s \in \{0.025, 0.05, 0.1\}$. The selected combination (2.0, 0.1, 0.05) maximized the product of validation success rate and path efficiency across 100 validation episodes.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Soft Actor-Critic Algorithm

Maximum-Entropy Objective

Standard RL maximizes expected cumulative reward:

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$$

Maximum-entropy RL augments this objective with a policy entropy bonus at each timestep, weighted by the temperature parameter $\alpha > 0$:

$$J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t \left(r_t + \alpha \cdot H(\pi(\cdot\,|s_t))\right)\right]$$

where

$$H(\pi(\cdot\,|s)) = -\mathbb{E}_{a\sim\pi}[\log\pi(a|s)]$$

is the differential entropy of the policy at state s. The temperature α controls the relative weight of the entropy term. At $\alpha = 0$, Maximum-entropy RL reduces to standard RL. At large α, the policy approaches a uniform distribution over the action space.

The maximum-entropy framework has an information-theoretic interpretation: the optimal policy is the one that maximizes expected reward while being as uncertain as possible about its actions given the current state. This uncertainty corresponds to behavioral diversity, which in the navigation context means the agent will distribute probability over multiple reasonable avoidance maneuvers rather than committing to one.

Soft Value Functions

SAC defines the soft Q-function and soft value function recursively. The soft Q-function satisfies the soft Bellman equation:

$$Q_{soft}(s_t, a_t) = r_t + \gamma \cdot \mathbb{E}_{s_{t+1}}[V_{soft}(s_{t+1})]$$

where the soft value function is defined as:

$$V_{soft}(s_t) = \mathbb{E}_{a\sim\pi}[Q_{soft}(s_t, a) - \alpha \cdot \log\pi(a|s_t)]$$

Substituting the soft value equation into soft Q value equation yields the expanded soft Bellman target:

$$y_t = r_t + \gamma \cdot (Q_{soft}(s_{t+1}, a_{t+1}) - \alpha \cdot \log\pi(a_{t+1}|s_{t+1}))$$

where $a_{t+1} \sim \pi(\cdot\,|s_{t+1})$. The target uses the same policy network that is being optimized, making SAC fully off-policy with respect to the behavioral policy used to collect experience.

Critic Objective

Following TD3, SAC trains two Q-function networks $(Q_{\theta_1}, Q_{\theta_2})$ with separate parameter sets (θ_1, θ_2) to reduce overestimation bias. The regression target for each critic uses the minimum Q-value from the target networks $(Q_{\bar{\theta_1}}, Q_{\bar{\theta_2}})$:

$$y = r + \gamma \cdot \left(\min_{j=1,2} Q_{\bar{\theta_j}}(s', \tilde{a}) - \alpha \cdot \log\pi_{\phi}(\tilde{a}|s')\right)$$

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

where $\tilde{a} \sim \pi_\phi(\cdot|s')$ is a fresh sample from the current policy. The loss for the i-th critic network is:

$$J_Q(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[0.5 \cdot (Q_{\theta_i}(s,a) - y)^2\right]$$

where $\mathcal{D}$ is the replay buffer. Target network parameters are updated via exponential moving average with rate $\tau = 0.005$:

$$\bar{\theta}_i \leftarrow \tau \cdot \theta_i + (1-\tau) \cdot \bar{\theta}_i$$

Actor (Policy) Objective

The actor is parameterized as a Gaussian policy π_ϕ with state-conditioned mean and log-variance networks. To enable gradients to flow through the sampled action, the reparameterization trick is used: actions are expressed as a deterministic transformation of noise $\epsilon \sim \mathcal{N}(0, I)$:

$$a_\phi(\epsilon; s) = \tanh(\mu_\phi(s) + \sigma_\phi(s) \cdot \epsilon)$$

The tanh squashing function maps unbounded samples to the bounded action space. The log-probability under the squashed Gaussian requires a Jacobian correction:

$$\log\pi_\phi(a|s) = \log\mathcal{N}(u; \mu_\phi, \sigma_\phi) - \sum_i \log\left(1 - \tanh^2(u_i)\right)$$

where u is the pre-tanh sample. The actor loss is:

$$J_\pi(\phi) = \mathbb{E}_{s\sim\mathcal{D},\epsilon\sim\mathcal{N}(0,I)}\left[\alpha \cdot \log\pi_\phi(a_\phi(\epsilon; s)|s) - \min_{j=1,2} Q_{\theta_j}(s, a_\phi(\epsilon; s))\right]$$

This objective encourages the policy to assign high probability to actions that have high Q-values while maintaining sufficient entropy to avoid deterministic collapse.

Automatic Temperature Tuning

Rather than fixing α as a hyperparameter, the automatic temperature method frames α as a learnable parameter and defines a constraint optimization problem:

$$\max_\pi \mathbb{E}_\pi\left[\sum_t r_t\right] \quad \text{s.t.} \quad \mathbb{E}_{(s_t,a_t)\sim\pi}[-\log\pi(a_t|s_t)] \geq H_{\text{target}}\ \forall t$$

This constrained problem can be solved with dual gradient descent. The dual variable corresponding to the entropy constraint is the temperature α, and its gradient update is:

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi_t}\left[-\alpha \cdot \log\pi_t(a_t|s_t) - \alpha \cdot H_{\text{target}}\right]$$

The target entropy is set to $H_{\text{target}} = -|\mathcal{A}| = -2$, following the recommendation in (Haarnoja et al. 2019), where $|\mathcal{A}| = 2$ is the dimensionality of the action space. Alpha is kept strictly positive by optimizing its log: $\alpha = \exp(\log\alpha)$.

Network Architecture

The actor network takes the 28-dimensional observation vector as input and outputs a 4-dimensional vector ($\mu_\phi \in \mathbb{R}^2, \log\sigma_\phi \in \mathbb{R}^2$) parameterizing the Gaussian policy over linear and angular velocities. The architecture consists of two fully-connected hidden layers with 256 units each, LayerNorm, and ReLU activations, followed by a 4-unit output layer. Layer normalization is applied after each linear layer to stabilize training without

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

introducing batch-size dependence. The log-standard-deviation outputs are clipped [-5, 2] to to prevent numerical instabilities.

Each of the two critic networks takes the concatenated observation-action vector (30-dimensional) as input and outputs a scalar Q-value estimate, using two fullyconnected hidden layers with 256 units and ReLU activations. All linear layers are initialized with orthogonal initialization using gain $\sqrt{2}$, except for the output layer of the actor which uses gain 0.01 to ensure near-uniform initial action distributions.

Training Procedure

The complete SAC training loop alternates between collecting transitions from the simulation using the current stochastic policy and performing gradient updates on the critic networks, actor network, and temperature parameter. Training proceeds for 1 million environment steps with the hyperparameters listed in Table 1.

SAC hyperparameter configuration.

Hyperparameter	Value
Discount factor γ	0.99
Replay buffer size $\lvert\mathcal{D}\rvert$	1,000,000
Batch size	256
Learning rates (Q, π, α)	3e-4,3e-4,3e-4
Target network update rate τ	0.005
Initial temperature α_0	0.2
Target entropy H_{target}	$-2\ (= -\lvert\mathcal{A}\rvert)$
Updates per environment step	1
Warm-up steps before updating	10,000
Max episode length (steps)	500
Control frequency	10 Hz

Experimental Environment and Setup

Simulation Platform

The simulation environment was built on Gymnasium, running on Windows 11. Observations, rewards, resets, and episode termination all run through the standard gym.Env API, with each step timed at 10 Hz to match the LiDAR update rate.

Warehouse Layouts

The simulated warehouse floor measures 20.0 m x 20.0 m with 1.0 m perimeter walls. The internal structure consists of six shelving units arranged in three parallel rows of two units each, with 2.0 m aisle widths between rows and 1.5 m end-of-row clearances. Shelving units are modeled as rectangular boxes with footprint 3.0 m 0.8 m and height 2.5 m, consistent with standard narrow-aisle shelving layouts used in e-commerce fulfillment centers.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Five goal locations are placed at named drop-off points: at each end of the three aisles and at a central staging area. At the start of each episode, the vehicle is placed at a randomly sampled starting position with clearance at least 1.0 m from all obstacles, and the goal is sampled uniformly from the five drop-off points. The minimum straight-line distance from any start position to any goal is 4.2 m; the maximum is 19.1 m. Three additional warehouse layouts were created for transfer evaluation, none seen during training. Layout B increases aisle width to 2.5 m and reduces shelving unit depth to 0.6 m. Layout C introduces diagonal shelving units at 45° angles, creating variable-width passages. Layout D represents a cross-docking configuration with no central shelving, replacing interior structure with randomly placed conveyor stations on a $5\text{m} \times 5\text{m}$ grid.

Dynamic Obstacles

Two categories of dynamic obstacle operate in the environment. Human pedestrians are modeled using the Social Force Model (SFM) (Helbing and Molnár 1995), which represents each pedestrian as a particle subject to attractive forces toward their waypoint destination and repulsive forces from other pedestrians, the vehicle under evaluation, and static obstacles. Desirable speeds are sampled from $\text{N}(1.2,0.1)\text{m/s}$ per pedestrian per episode. Pedestrian waypoints are sampled uniformly from pick locations distributed along shelving units, producing foot-traffic patterns representative of warehouse pick operations. The number of pedestrian agents n_{ped} is randomized per episode during training: $\text{n}_{\text{ped}} \sim \text{Uniform}(2,3,4,5)$.

Peer AGVs follow pre-computed patrol routes connecting drop-off points, with route assignment randomized per episode. Each peer AGV uses DWA to avoid pedestrians and the training vehicle, so their trajectories are non-trivial and responsive to the training vehicle's behavior. The number of peer AGVs n_{agv} is randomized per episode: $\text{n}_{\text{agv}} \sim \text{Uniform}(1,2,3)$. During evaluation at maximum obstacle density $\text{n}_{\text{ped}} = 5$, and $\text{n}_{\text{agv}} = 3$ for all trials.

Sensor Model and AGV Kinematics

The LiDAR sensor is modeled as a 2D planar scanner with 24 beams uniformly distributed across a $350°$ field of view, maximum range 10.0 m, and 10 Hz update rate. Gaussian measurement noise with zero mean and standard deviation $\sigma_{\text{lidar}} = 0.02\text{m}$ is added to each beam return. During training with domain randomization, σ_{lidar} is sampled per episode from Uniform(0.01, 0.05), spanning the noise characteristics of typical low-cost solid-state LiDAR units.

The AGV is modeled as a differential-drive robot with wheelbase $\text{L} = 0.5\text{m}$, wheel radius $\text{r} = 0.1\text{m}$, and vehicle footprint $0.6\text{m} \times 0.4\text{m}$. Wheel speed saturation is enforced at $\text{v}_{\text{wheel,max}} = 2.0\text{rad/s}$, equivalent to a maximum linear velocity of 0.2 m/s per wheel. An actuation delay of one control step (100 ms) is modeled by buffering the commanded velocities.

Domain Randomization

Four key parameters are randomized during training, as summarized in Table 2. The LiDAR noise standard deviation is sampled per episode from Uniform (0.01, 0.05) m. Pedestrian count is sampled uniformly from 2, 3, 4, 5, and peer AGV count from 1, 2, 3. Pedestrian speed is sampled from $\text{N}(1.2,0.2)$ m/s clipped to [0.8, 1.6] m/s. These ranges are grounded in practical operating conditions, ensuring the randomization is meaningful without pushing the system into physically implausible scenarios.

Domain Randomization Parameters

Parameter	Nominal	Range	Rationale
LiDAR noise	0.02	0.01–0.05	Sensor variability
Pedestrians	5	2–5	Traffic density

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Speed	1.2	0.8–1.6	Variation
AGVs	3	1–3	Fleet size

Evaluation Protocol

Four navigation agents were evaluated: SAC (proposed), DDPG, TD3, and DWA. All three deep RL agents were trained for 1 million environment steps on Layout A under identical hyperparameter configurations. DWA was configured with standard ROS Navigation Stack parameters, operating on the ground-truth occupancy map of each layout with a planning horizon of 1.5 seconds. Evaluation was conducted across all four warehouse layouts, with each condition evaluated over 100 episodes at maximum obstacle density ($n_{ped} = 5, n_{agv=3}$) using deterministic action selection for deep RL agents (policy mean). Seven metrics were recorded: success rate, collision rate, timeout rate, mean cumulative reward, mean path length, mean energy consumption (sum of squared velocity penalties $\sum v_t^2$), and smoothness penalty (sum of velocity-change penalties).

Results and Discussion

Overall Performance

Table 3 presents the complete evaluation results for all four agents across all four layouts.

Table 3: Complete Evaluation Results Across All Layouts (100 Episodes per Condition)

Layout	Agent	Success (%)	Collision (%)	Timeout (%)	Mean Reward	Path (m)	Energy	Smooth-ness
A	SAC	78.0	3.0	19.0	100.43	15.22	14.43	25.08
A	DDPG	81.0	6.0	13.0	85.60	21.83	26.64	51.20
A	TD3	69.0	4.0	27.0	81.79	27.65	29.94	65.51
A	DWA	0.0	1.0	99.0	64.11	31.34	7.99	16.81
B	SAC	80.0	1.0	19.0	107.29	15.63	13.69	24.70
B	DDPG	83.0	7.0	10.0	91.42	19.08	21.86	42.84
B	TD3	74.0	6.0	20.0	94.46	23.66	24.52	46.21
B	DWA	0.0	0.0	100.0	66.81	31.83	8.16	17.04
C	SAC	80.0	1.0	19.0	107.61	15.70	13.72	24.72
C	DDPG	82.0	7.0	11.0	88.34	19.68	22.83	46.18
C	TD3	73.0	6.0	21.0	92.83	24.27	25.38	48.09
C	DWA	0.0	0.0	100.0	67.16	31.89	8.20	17.03
D	SAC	66.0	1.0	33.0	81.96	15.63	13.40	26.15
D	DDPG	67.0	4.0	29.0	63.37	28.10	38.01	75.53
D	TD3	72.0	3.0	25.0	85.24	26.56	26.84	54.74
D	DWA	0.0	36.0	64.0	23.58	23.45	6.91	11.82

Table 4 provides per-metric averages across layouts for concise comparison.

Table 4: Average Performance Across All Four Layouts

Metric	SAC	DDPG	TD3	DWA	Best Agent
Avg Success Rate (%)	76.0	78.2	72.0	0.0	DDPG
Avg Collision Rate (%)	1.50	6.00	4.75	9.25	SAC ✓
Avg Mean Reward	99.32	82.18	88.58	55.41	SAC ✓
Avg Path Length (m)	15.54	22.17	25.54	29.63	SAC ✓
Avg Energy	13.81	27.34	26.67	7.81	SAC ✓
Avg Smoothness Penalty	25.16	53.94	53.64	15.68	SAC ✓

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

Energy Efficiency and Trajectory Smoothness

Table 5: Energy Consumption Comparison: Ratio Relative to SAC

Layout	SAC Energy	DDPG Energy	TD3 Energy	DDPG / SAC	TD3 / SAC
A	14.43	26.64	29.94	1.85×	2.07×
B	13.69	21.86	24.52	1.60×	1.79×
C	13.72	22.83	25.38	1.66×	1.85×
D	13.40	38.01	26.84	2.84×	2.00×
Average	**13.81**	**27.34**	**26.67**	**1.98×**	**1.93×**

Energy consumption results are summarized in Table V. SAC maintains a near-constant and lowest energy profile among deep RL agents across all layouts (13.40 – 14.43), while DDPG and TD3 consume substantially more energy, peaking at 38.01 and 29.94 respectively. On average, DDPG consumes 1.98 times and TD3 1.93 times more energy than SAC per episode. The most pronounced gap occurs on Layout D (cross-docking), where the absence of shelving aisles induces unconstrained high-speed trajectories in DDPG and TD3. DWA records the lowest absolute energy values, but this is a direct consequence of its near-zero locomotion due to frequent deadlocking rather than genuine efficiency.

The smoothness penalty reinforces this interpretation. SAC achieves an average smoothness penalty of 25.16, compared to 53.94 for DDPG and 53.64 for TD3 approximately half the smoothness penalty of both baselines across all layouts. This difference indicates that DDPG and TD3 produce trajectories with frequent, sharp velocity reversals. In physical AGV platforms, such reversals generate mechanical stress on drive systems and cargo and are a primary source of component wear in high-utilization deployments. SAC's smooth trajectories directly address this operational concern.

Together, the energy and smoothness results demonstrate that the five-component reward function successfully shaped SAC's policy toward the deployment-relevant objectives of energy conservation and mechanical preservation, without sacrificing task success rate relative to TD3 (SAC 76.0% vs. TD3 72.0% on average).

DWA Failure Analysis

DWA achieves 0% success across all four layouts, with timeout rates of 99–100% on Layouts A, B, and C. This complete failure requires analysis, as DWA is a widely deployed industrial planner. DWA is a reactive local planner that scores velocity commands against a cost function balancing clearance, heading alignment, and forward speed over a short prediction horizon of 1.5 seconds. In static or low-density dynamic environments, DWA performs reliably. The failure mode observed here near-total timeout with very low collision rate—is consistent with well-documented DWA deadlock behavior: when confronted with multiple simultaneously moving obstacles (5 pedestrians + 3 peer AGVs), the planner repeatedly decelerates to zero velocity to maintain safe clearance, accumulating no progress toward the goal until the episode time limit is reached. On Layout D (cross-docking, open space), DWA's behavior changes qualitatively: the success rate remains 0% but the collision rate rises to 36% and the timeout rate drops to 64%. Without the structural guidance of shelving aisles, DWA produces longer exploratory trajectories that encounter obstacles more frequently. These results empirically confirm the central motivation of this work: classical reactive planners are inadequate for high-density dynamic warehouse environments. The inability of DWA to achieve a single successful navigation episode across 400 evaluation trials provides strong justification for the deep RL approach.

Generalization Analysis

Domain randomization's effectiveness is assessed through SAC's performance on unseen layouts. SAC's success rates are 78% (A), 80% (B), 80% (C), and 66% (D). Performance on Layouts B and C—the two structurally novel layouts with wider aisles and diagonal shelving—is essentially unchanged from the training

layout (80% vs. 78%, a 2 percentage point improvement). This is a strong generalization result: the agent trained on Layout A with domain randomization transfers without degradation to both a wider-aisle configuration and a diagonal-shelving configuration. The 12 percentage point drop from Layout C (80 Layout D (66%) warrants explanation. Layout D is a cross-docking configuration with no central shelving, replacing structured aisles with open space and randomly placed conveyor stations. This structural change eliminates the aisle-corridor topology that characterized all training episodes. The 66% success rate still substantially exceeds DDPG (67%) and TD3 (72%), confirming that SAC's policy maintains competitive performance even on structurally atypical layouts. For comparison, DDPG shows a similar 15 percentage point drop to Layout D, indicating that the cross-docking challenge is layout-intrinsic rather than algorithm-specific. The most compelling generalization evidence is SAC's path length invariance across layouts (standard deviation 0.22 m), which is a more sensitive indicator of policy generalization than success rate. A layout-overfit policy would produce longer or more tortuous paths when layout geometry changes, even if it eventually reaches the goal. SAC's near-constant path length across four structurally distinct layouts indicates that the learned navigation strategy is not conditioned on specific layout topology.

Why SAC Outperforms Deterministic Alternatives

A common concern about SAC in safety-critical navigation is that entropy maximization encourages behavioral diversity at the expense of converging to reliable collision-avoidance reflexes. Our results do not support this concern: SAC achieves lower collision rates than TD3 on all four layouts (1–3% vs. 3–6%), shorter paths, lower energy consumption, and higher mean reward. The behavioral diversity induced by entropy regularization appears beneficial rather than harmful. The warehouse navigation task involves heterogeneous dynamic obstacles (human pedestrians governed by the Social Force Model and peer AGVs executing DWAbased avoidance). The space of viable avoidance maneuvers is genuinely multi-modal: in many configurations, moving left, moving right, or slowing down are all approximately equivalent. An entropyregularized policy distributes probability over this equivalence class, producing smoother average trajectories and avoiding deterministic commitment to a single strategy that may fail in novel configurations. TD3's higher timeout rate (27% on Layout A vs. SAC's 19%) suggests that its deterministic policy gets committed to avoidance trajectories that result in navigation dead-ends near the goal region. SAC's stochastic policy escapes these dead-ends by sampling alternative actions, consistent with the known advantage of entropy-regularized policies in environments with local optima.

Limitations

The following limitations should be considered when interpreting our results. First, all deep RL agents were trained with a single random seed. Deep RL performance is known to exhibit high seed-to-seed variance (Henderson et al. 2017), and the reported differences, particularly the 2–3 percentage point success rate gap between SAC and DDPG, may not be statistically stable across multiple seeds. The 4–6 percentage point collision rate differences and 2x energy differences are larger in magnitude and more likely to be robust, but replication with 5–10 seeds remains necessary to confirm algorithmic rather than initialization effects. Second, the environment models a planar 2D warehouse floor. Real warehouses involve 3D obstacles, surface irregularities, load-shifting dynamics, and actuator nonlinearities not captured in the Gymnasium simulation. Sim-to-real transfer validation on physical hardware is required before deployment conclusions can be drawn. Third, the contribution of individual reward components was not quantified through controlled ablation (training with each term removed). While the energy and smoothness advantages of SAC are consistent with the reward design, the relative magnitude of each term's contribution has not been empirically isolated. Fourth, DWA was evaluated with standard ROS Navigation Stack parameters; a domainexpert-tuned DWA configuration might achieve marginally higher success rates, though the fundamental limitation of short-horizon reactive planning in highdensity dynamic environments is unlikely to be overcome by parameter tuning alone. Finally, pedestrian dynamics are modeled using the Social Force Model, which captures crowd-level behavior but does not replicate the full variability of human motion in real pick operations, including stopping behavior, item inspection, and group clustering near popular pick locations.

ISSN No. 2321-2705 | DOI: 10.51244/IJRSI |Volume XIII Issue VI June 2026

CONCLUSION AND FUTURE WORK

This paper investigated whether Soft Actor-Critic reinforcement learning, combined with a multicomponent reward function and domain randomization, produces a warehouse AGV navigation controller that outperforms DDPG, TD3, and DWA baselines across both training and unseen environments. The results provide affirmative, quantitatively supported conclusions across all research objectives. SAC achieves superior composite navigation performance across all four evaluation layouts. It records the highest mean cumulative reward on every layout (average 99.32 vs. 88.58 for TD3, 82.18 for DDPG, and 55.41 for DWA), the lowest collision rate (average 1.5% vs. 4.75% for TD3, 6.0% for DDPG, and 9.25% for DWA), the shortest path lengths (average 15.54 m vs. 25.54 m for TD3 and 22.17 m for DDPG), the lowest energy consumption (average 13.81 vs. 26.67 for TD3 and 27.34 for DDPG), and the smoothest trajectories (average 25.16 vs. 53.64 for TD3 and 53.94 for DDPG). While DDPG achieves a 2.2 percentage point higher average success rate, this comes at the cost of a 4x higher collision rate and 2x higher energy consumption a trade-off that is unfavorable for safety critical warehouse deployment. the multi-component reward function successfully shaped SAC's policy to optimize all targeted behaviors simultaneously. The near-2x energy advantage over DDPG and TD3, the 2x smoothness advantage, and the lowest collision rate empirically confirm that the reward components for energy, smoothness, and collision avoidance are all active and effective in the learned policy. Domain randomization produces a controller with strong generalization: SAC maintains 78–80% success on Layouts A, B, and C (training layout and two structurally novel layouts), with near-invariant path lengths across all four layouts (standard deviation 0.22 m). The absence of degradation on Layouts B (wider aisles) and C (diagonal shelving) directly validates the domain randomization approach.

The most significant practical finding is the complete failure of DWA across all 400 evaluation episodes, providing strong empirical evidence that classical reactive planning is inadequate for modern high-density warehouse environments and establishing a clear case for deep RL-based navigation as the superior approach. Among the RL methods evaluated, SAC's maximum entropy formulation produces the most deployment ready controller: lowest collision rate, highest reward, shortest paths, lowest energy, and smoothest trajectories, with consistent generalization across layout types.

Research Contributions

This work makes four primary contributions to the field of autonomous warehouse navigation. First, we establish the first direct comparison of SAC, TD3, DDPG, and DWA on a unified dynamic warehouse navigation benchmark with heterogeneous obstacle types, providing rigorous empirical evidence for SAC's superiority across a comprehensive six-metric evaluation profile. Second, we design and validate a principled five-component reward function that jointly encodes goal-reaching efficiency, collision avoidance, energy consumption, and trajectory smoothness, demonstrating that careful reward shaping enables simultaneous optimization of all four objectives without trade-offs that degrade safety. Third, we provide the first characterization of energy efficiency differences between SAC and deterministic actor-critic methods in warehouse navigation, showing consistent approximately 2x energy advantages that directly translate to operational benefits in battery-limited AGV deployments. Fourth, we demonstrate geometry agnostic navigation capability through near-invariant path lengths (standard deviation 0.22 m) across four structurally distinct warehouse layouts, providing empirical evidence that entropy-regularized training with domain randomization produces policies that generalize across fundamentally different spatial topologies.

Practical Implications for Warehouse Automation

The results of this study carry several practical implications for warehouse operators and AGV system integrators. The 1.5% collision rate achieved by SAC approaches the sub-1% threshold typically required for commercial AGV certification, suggesting that deep RLbased navigation is nearing deployment readiness for controlled warehouse environments. The 24% average non-success rate (timeout rather than collision) indicates that navigation failures occur primarily through safe stagnation—the agent fails without causing damage, a favorable failure mode for physical systems. The approximately 50% reduction in energy consumption relative to DDPG and TD3 translates directly to extended battery life and increased vehicle utilization between charges. For a fleet

of 50 AGVs each completing 20 trips per 8-hour shift, SAC's energy efficiency could eliminate 1–2 battery swap cycles per vehicle per day, reducing downtime and extending battery longevity. The 2 improvement in trajectory smoothness reduces mechanical wear on drive systems and cargo mounts, lowering maintenance frequency and improving operational reliability over extended deployment periods.

Most significantly, the geometry-agnostic navigation capability demonstrated by SAC's invariant path lengths suggests that a single trained policy may be deployable across multiple zones within a warehouse or across different facilities with varying layouts, reducing the need for per-zone policy retraining and accelerating deployment timelines.

Real-World Deployment Considerations

Transitioning from simulation to physical warehouse deployment involves several engineering and safety considerations. First, the inference latency of 2.3 ms on GPU and 8.7 ms on CPU (Table 6) is well within the 100 ms control cycle, confirming that SAC policies can execute in real-time on standard onboard computing hardware without requiring specialized accelerators.

Policy Forward Pass Latency (Batch Size 1)

Device	Mean Latency	Std Dev	99th Percentile
GPU (GTX 960M)	2.3 ms	0.4 ms	3.8 ms
CPU (i5-4210H)	8.7 ms	1.2 ms	12.4 ms

2 Second, SAC's conservative velocity profiles and smooth trajectories are favorable properties for physical transfer: they reduce actuator command frequency and produce more predictable low-level controller behavior. The smoothness penalty in the reward function explicitly discourages high-frequency command oscillations that can excite unmodeled dynamics or cause wheel slip on physical platforms. Third, the domain randomization strategy provides a principled starting point for sim-to-real transfer: the randomized sensor noise range [0.01, 0.05] m encompasses typical low-cost LiDAR units (e.g., SICK TiM571, Hokuyo URG-04LX), while the randomized obstacle densities and velocities span the operating conditions expected in real fulfillment centers. Fourth, the timeout failure mode (rather than collision) suggests that a supervisory safety system that detects stagnation and triggers a low-level emergency stop or human alert would provide an additional safety layer without requiring modifications to the learned policy. Finally, we recommend a hybrid deployment architecture that uses SAC for short-range reactive avoidance combined with A* for global path planning; this preserves SAC's safety advantages while addressing the timeout failure mode through longhorizon goal direction maintained by the global planner.

Future Directions

Several directions merit investigation. Multi-seed replication with 5–10 random seeds is needed to quantify performance variance and establish statistically valid ranking confidence. SAC's primary remaining weakness is the 19–33% timeout rate; targeted interventions include extended training beyond 1 million steps, curriculum learning beginning with low obstacle density and progressively increasing it, and adjusting the progress shaping weight to create stronger pull toward the goal during mid-episode stalling. Full factorial ablation of reward components would empirically quantify each term's contribution. Incorporating aisle width, shelf depth, and shelf orientation into the domain randomization schedule could further improve generalization, particularly for Layout D-type configurations. Physical robot validation on a differential-drive AGV platform is essential to characterize sim-to-real gaps in actuator response, sensor noise, and surface friction.

REFERENCES

1. Alahi, Alexandre, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. 2016. "Social LSTM: Human Trajectory Prediction in Crowded Spaces." 2016 IEEE

Conference on Computer Vision and Pattern Recognition (CVPR), 961–71. https://doi.org/10.1109/CVPR.2016.110.
2. Andrychowicz, Marcin, Bowen Baker, Maciek Chociej, et al. 2018. "Learning Dexterous in-Hand Manipulation." The International Journal of Robotics Research 39: 20–23. https://api.semanticscholar.org/CorpusID:51894399.
3. Dijkstra, E. W. 1959. "A Note on Two Problems in Connexion with Graphs." Numerische Mathematik 1 (1): 269–71. https://doi.org/10.1007/BF01386390.
4. Duan, Yan, Xi Chen, Rein Houthooft, John Schulman, and P. Abbeel. 2016. "Benchmarking Deep Reinforcement Learning for Continuous Control." International Conference on Machine Learning. https://api.semanticscholar.org/CorpusID:12296499.
5. Fox, D., W. Burgard, and S. Thrun. 1997. "The Dynamic Window Approach to Collision Avoidance." IEEE Robotics and Automation Magazine 4 (1): 23–33. https://doi.org/10.1109/100.580977.
6. Fragapane, Giuseppe, René de Koster, Fabio Sgarbossa, and Jan Ola Strandhagen. 2021. "Planning and Control of Autonomous Mobile Robots for Intralogistics: Literature Review and Research Agenda." European Journal of Operational Research 294 (2): 405–26. https://doi.org/https://doi.org/10.1016/j.ejor.2021.01.019.
7. Fujimoto, Scott, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. https://arxiv.org/abs/1802.09477.
8. Guizzo, Eric. 2008. "Three Engineers, Hundreds of Robots, One Warehouse." IEEE Spectrum 45 (7): 26–34. https://doi.org/10.1109/MSPEC.2008.4547508.
9. Haarnoja, Tuomas, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. https://arxiv.org/abs/1801.01290.
10. Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, et al. 2019. Soft Actor-Critic Algorithms and Applications. https://arxiv.org/abs/1812.05905.
11. Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. 1968. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." IEEE Transactions on Systems Science and Cybernetics 4 (2): 100–107. https://doi.org/10.1109/TSSC.1968.300136.
12. Helbing, Dirk, and Péter Molnár. 1995. "Social Force Model for Pedestrian Dynamics." Physical Review. E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics 51 5: 4282–86. https://api.semanticscholar.org/CorpusID:5771125.
13. Henderson, Peter, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2017. "Deep Reinforcement Learning That Matters." AAAI Conference on Artificial Intelligence. https://api.semanticscholar.org/CorpusID:4674781.
14. Jian, Zhu, Zihong Yan, Xuanang Lei, et al. 2022. "Dynamic Control Barrier Function-Based Model Predictive Control to Safety-Critical Obstacle-Avoidance of Mobile Robot." 2023 IEEE International Conference on Robotics and Automation (ICRA), 3679–85. https://api.semanticscholar.org/CorpusID:252367490.
15. Kadian, Abhishek, Joanne Truong, Aaron Gokaslan, et al. 2019. "Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance?" IEEE Robotics and Automation Letters 5: 6670–77. https://api.semanticscholar.org/CorpusID:221082834.
16. Karaman, Sertac, and Emilio Frazzoli. 2011. Sampling-Based Algorithms for Optimal Motion Planning. https://arxiv.org/abs/1105.1186.
17. Lackner, Thorge, Julian Hermann, Christian Kuhn, and Daniel Palm. 2024. "Review of Autonomous Mobile Robots in Intralogistics: State-of-the-Art, Limitations and Research Gaps." Procedia CIRP. https://api.semanticscholar.org/CorpusID:274360266.
18. LaValle, Steven M. 1998. "Rapidly-Exploring Random Trees : A New Tool for Path Planning." The Annual Research Report. https://api.semanticscholar.org/CorpusID:14744621.
19. LaValle, Steven M. 2006. "Planning Algorithms." https://api.semanticscholar.org/CorpusID:15371216.
20. Lillicrap, Timothy P., Jonathan J. Hunt, Alexander Pritzel, et al. 2015. "Continuous Control with Deep Reinforcement Learning." arXiv: Learning. https://api.semanticscholar.org/CorpusID:16326763.
21. Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, et al. 2015. "Human-Level Control Through Deep Reinforcement Learning." Nature 518: 529–33. https://api.semanticscholar.org/CorpusID:205242740.

22. Ng, A., Daishi Harada, and Stuart J. Russell. 1999. “Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping.” International Conference on Machine Learning. https://api.semanticscholar.org/CorpusID:5730166.
23. Peng, Xue Bin, Marcin Andrychowicz, Wojciech Zaremba, and P. Abbeel. 2017. “Sim-to-Real Transfer of Robotic Control with Dynamics Randomization.” 2018 IEEE International Conference on Robotics and Automation (ICRA), 1–8. https://api.semanticscholar.org/CorpusID:3707478.
24. Rösmann, Christoph, Wendelin Feiten, Thomas Wösch, Frank Hoffmann, and Torsten Bertram. 2012. “Trajectory Modification Considering Dynamic Constraints of Autonomous Robots.” German Conference on Robotics. https://api.semanticscholar.org/CorpusID:2043153.
25. Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. https://arxiv.org/abs/1707.06347.
26. Sutton, Richard S., David A. McAllester, Satinder Singh, and Y. Mansour. 1999. “Policy Gradient Methods for Reinforcement Learning with Function Approximation.” Neural Information Processing Systems. https://api.semanticscholar.org/CorpusID:1211821.
27. Tai, Lei, Giuseppe Paolo, and Ming Liu. 2017. “Virtual-to-Real Deep Reinforcement Learning: Continuous Control of Mobile Robots for Mapless Navigation.” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 31–36. https://doi.org/10.1109/IROS.2017.8202134.
28. Tobin, Josh, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. https://arxiv.org/abs/1703.06907.
29. Waga, Abderrahim, Said Benhlima, Ali Bekri, Jawad Abdouni, and Fatima Zahrae Saber. 2025. “A Survey on Autonomous Navigation for Mobile Robots: From Traditional Techniques to Deep Learning and Large Language Models.” Journal of King Saud University Computer and Information Sciences 37. https://api.semanticscholar.org/CorpusID:280771817.
30. Watkins, Christopher, and Peter Dayan. 1992. “Q-Learning.” Machine Learning 8: 279–92. https://api.semanticscholar.org/CorpusID:208910339.
31. Zhao, Wenshuai, Jorge Peña Queralta, and Tomi Westerlund. 2020. “Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey.” 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 737–44. https://doi.org/10.1109/SSCI47803.2020.9308468.
32. Zhu, Yingjie, Wan Zuha Wan Hasan, Hafiz Rashidi Harun Ramli, Nor Mohd Haziq Norsahperi, Muhamad Saufi Mohd Kassim, and Yiduo Yao. 2025. “Deep Reinforcement Learning of Mobile Robot Navigation in Dynamic Environment: A Review.” Sensors (Basel, Switzerland) 25. https://api.semanticscholar.org/CorpusID:278981018.