Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.083491

ARTICLE

SW-DWNS: A Single-Wave Autonomous Navigation System in Partially Observable, Highly Dynamic Warehouses

Xianhui Fan[1], **Zongwei Li**[1,*], **Yuxuan Zhai**[1] and **Zhenyu Li**[2]

[1]School of Economics and Management, Shanghai Institution of Technology, Shanghai, China
[2]School of Cultural Heritage and Information Management, Shanghai University, Shanghai, China

*Corresponding Author: Zongwei Li. Email: lzw0118@163.com

Received: 04 April 2026; Accepted: 11 June 2026; Published: 23 July 2026

ABSTRACT: Single-wave order picking in dynamic warehouses is a sequential multi-goal navigation problem. A robot must visit an ordered set of shelves and then a delivery station while avoiding moving obstacles under partial observability. Existing approaches either entangle long-horizon task logic with low-level obstacle avoidance or rely on static-environment assumptions that limit responsiveness in dynamic settings. This paper proposes the Single-Wave Dynamic Warehouse Navigation System (SW-DWNS), a lightweight scheduling framework that extends a pretrained ColorDynamic point-to-point local planner to ordered warehouse picking without retraining. The scheduler maintains a shelf queue, exposes only the active subgoal to the local policy, and advances the queue through an arrival-gated state machine with a geometrically derived pickup threshold. We evaluate SW-DWNS in a controlled dynamic warehouse simulation benchmark with three layouts, pickup horizons $K \in \{1, \ldots, 5\}$, stochastic obstacle motion, sensor noise, and obstacle-size randomization. Across the full layout–horizon grid, SW-DWNS achieves 87.3% average success, compared with 26.8% for A* search combined with the Dynamic Window Approach (A* + DWA) and 1.3% for ColorDynamic-only. The gain is not obtained by sacrificing safety: the average collision rate is reduced from 69.0% for A* + DWA to 7.5% for SW-DWNS. Factor-wise ablation shows uneven robustness across disturbances: sensing noise and obstacle-size randomization have limited impact (main effects of +1 and +4.5 percentage points), whereas disabling obstacle motion costs 33 percentage points and shifts failures from collision to timeout. A pickup-threshold sweep further shows that reported success rates become inflated outside the admissible geometric band $\varepsilon \in [14, 31]$ cm. These results show that a decoupled scheduling layer can make a pretrained dynamic local planner effective for sequential warehouse picking, while also revealing the conditions under which that extension remains valid.

KEYWORDS: Warehouse robot; single-wave order picking; navigation under partial observability; highly dynamic environment; autonomous navigation system

1 Introduction

Intelligent warehouse systems are core infrastructure for modern e-commerce supply chains, where Business-to-Consumer (B2C) fulfilment must process large volumes of fragmented orders under increasingly tight delivery-time constraints. Wave picking is widely used in this setting because it batches multiple customer orders into a coordinated retrieval process, reducing redundant travel and improving equipment utilization. For a mobile robot, however, a wave is not a single navigation query: it is a sequential task in which the robot must visit multiple target shelves in order and then return to a delivery station while sharing the workspace with workers, forklifts, or other robots.

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This requirement exposes a gap between task-level scheduling and local dynamic navigation. Classical warehouse scheduling methods, including analytical models, simulation-based methods, meta-heuristics, and hybrid optimization strategies [1], can assign and sequence work under simplified assumptions [2,3], but they often rely on static models, global information, or offline computation [4]. In contrast, recent deep reinforcement learning based local planners [5] can react to moving obstacles and noisy observations, but they are usually trained for point-to-point navigation. A policy that can reach one goal in a dynamic scene does not automatically know how to maintain an ordered shelf queue, when to switch to the next target, or when the wave is complete.

End-to-end or hierarchical reinforcement learning [6,7] could in principle learn the full wave-picking policy, but at substantial training cost, and must learn temporal task logic jointly with local navigation, making adaptation costly when the task definition changes. Yet a competent pretrained point-to-point local planner is often already available. The more immediate systems question is whether such a planner can be extended to sequential warehouse picking without retraining, by separating temporal task logic from local navigation execution.

This paper answers that question with the Single-Wave Dynamic Warehouse Navigation System (SW-DWNS). SW-DWNS places a lightweight scheduling layer above the pretrained ColorDynamic (CD) local planner [5], maintaining an ordered shelf queue and exposing only the current subgoal to it. When the robot enters a geometrically derived pickup threshold (Section 3.5) around the active shelf, the queue advances; after all shelves are visited, the delivery station becomes the final goal. The result is a decoupled architecture that turns a sequential warehouse-picking task into a series of point-to-point dynamic navigation subtasks, without modifying or fine-tuning the underlying policy.

The central question is empirical: is this lightweight layer a mere convenience or a load-bearing component? We evaluate SW-DWNS on a controlled dynamic warehouse simulation benchmark across three layouts, pickup horizons $K \in \{1, \ldots, 5\}$, stochastic obstacle motion, obstacle-size randomization, and sensor noise, against a classical A* search combined with the Dynamic Window Approach (A* + DWA) pipeline and a ColorDynamic-only (CD-only) ablation that removes the scheduling layer. Our contributions are threefold:

1. We formulate single-wave order picking as a sequential dynamic navigation problem and propose SW-DWNS, a lightweight scheduling framework that extends a pretrained point-to-point dynamic navigator to ordered multi-shelf tasks without retraining.
2. We provide system-level comparisons and ablations showing that the scheduling layer is necessary: SW-DWNS outperforms both A* + DWA and CD-only, demonstrating that the gain depends on the scheduling layer rather than the local planner alone.
3. We analyze the robustness boundary of the system through layout-scale evaluation, factor-wise disturbance ablation, and pickup-threshold sensitivity. The results identify obstacle motion as a load-bearing deployment assumption and show that geometrically justified arrival thresholds are necessary for fair success-rate evaluation.

2 Problem Setting and Experimental Environments

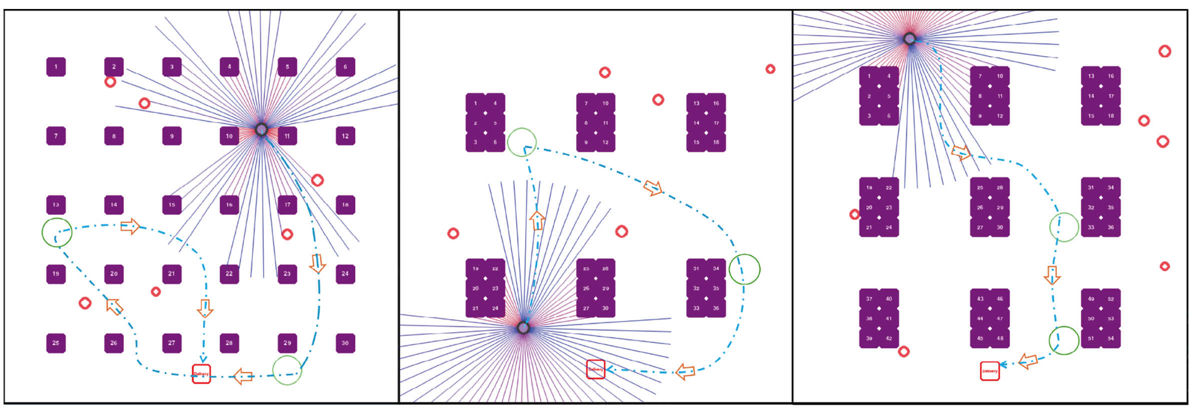
We address the single-wave order picking problem in dynamic warehouses: an autonomous mobile robot (AMR) departs from a designated start node, sequentially visits K target shelving units in a prescribed order, and transports the collected items to a delivery station. A mission is deemed successful only if the entire target sequence is completed without collision and within a prescribed step budget; otherwise the failure is recorded as either COLLISION or TIMEOUT. The robot operates under partial observability (it relies solely on local LiDAR (Light Detection and Ranging) readings, neither receiving a global map nor anticipating future obstacle trajectories) and must contend with stochastically moving dynamic obstacles such as pedestrians, forklifts, and peer AMRs. A formal Markov Decision Process treatment of this setting is deferred to Section 3.1.

To probe SW-DWNS across diverse spatial configurations, we constructed three warehouse layouts spanning a deliberate hardness gradient in shelf density, aisle geometry, and obstacle-routing freedom. All three share the same global footprint (800 cm × 800 cm) and dynamic-obstacle configuration (Section 3.2), so that any performance variation across them is attributable to layout structure rather than environment scale. We use $K = 2$ as a recurring illustrative scale for the trajectory visualizations in Fig. 1; the full sweep $K \in \{1, \ldots, 5\}$ is reported in Section 4.

(1) Dispersed Layout (Fig. 1a). Thirty shelving units form a 5 × 6 grid spanning the workspace. The open lattice between rows gives the AMR ample freedom for global routing, but the aisles between shelves are narrow, leaving little clearance during inter-shelf transit, a macro–micro asymmetry: open routing space, precise micro-maneuvers near each target.

(2) Compact-Simple Layout (Fig. 1b). Thirty-six shelving units cluster into six zones, each a 2 × 3 sub-array, separated by wide inter-zone corridors. Intra-zone aisles are tighter than in the dispersed layout, demanding more of local obstacle avoidance, while the wide corridors leave ample room to cruise and re-route between zones. Difficulty thus concentrates in the local navigation layer, with global routing still open.

(3) Compact-Complex Layout (Fig. 1c). Fifty-four shelving units fill nine zones of the same 2 × 3 pattern, raising both shelf density and zone count by 50% over Compact-Simple. The denser packing narrows the inter-zone corridors and lengthens intra-zone traversal, eroding both global and local degrees of freedom, making this layout the hardness upper bound of our evaluation suite.



(a) (b) (c)

Figure 1: Representative successful K = 2 task trajectories in the three warehouse layouts: (**a**) Dispersed; (**b**) Compact-Simple; (**c**) Compact-Complex. Green circle: start; red square: delivery station; dash-dot line: robot trajectory; rings: dynamic obstacles; fan: LiDAR scan.

3 Method

3.1 Problem Formalization

We cast the single-wave order picking problem as a sequential goal-conditioned Markov Decision Process (MDP).

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \mathcal{G} \rangle \quad (1)$$

State space $\mathcal{S}$. The state $\boldsymbol{s}_t \in \mathcal{S}$ at time t comprises three elements:

$$\boldsymbol{s}_t = [\boldsymbol{K}(t), \boldsymbol{L}(t), \boldsymbol{g}_t] \tag{2}$$

where the robot kinematic vector $\boldsymbol{K}(t)$ and the temporal LiDAR window $\boldsymbol{L}(t)$ are, respectively,

$$\boldsymbol{K}(t) = [V_l^{ct}, V_a^{ct}, V_l^{rt}, V_a^{rt}, D_{2T}, \alpha, V_l, V_a] \in \mathbb{R}^8 \tag{3}$$

$$\boldsymbol{L}(t) = (\boldsymbol{l}_{t-T+1}, \ldots, \boldsymbol{l}_t), \boldsymbol{l}_\tau \in \mathbb{R}^{n_g}, T = 10, n_g = 24 \tag{4}$$

Here, $D_{2\mathrm{T}} = \|\boldsymbol{p}_t - \boldsymbol{g}_t\|_2$ is the Euclidean distance from the robot position $\boldsymbol{p}_t$ to the active subgoal $\boldsymbol{g}_t$, and α is the bearing of $\boldsymbol{g}_t$ relative to the robot's heading. The active subgoal $\boldsymbol{g}_t$ is maintained by the scheduling layer (Section 3.4). Implementation details of state normalization and observation noise are deferred to Section 3.3 and Supplementary Section S1.

Action space $\mathcal{A}$. A 7-dimensional discrete action set is adopted:

$$\mathcal{A} = \{a^{(0)}, a^{(1)}, \ldots, a^{(6)}\} \tag{5}$$

Each $a^{(i)}$ corresponds to a predefined pair of linear and angular velocity commands (V_l, V_a); exact values are listed in Section 3.3.

Reward function $\mathcal{R}$ combines sparse terminal rewards with a dense shaping reward:

$$r_t = \begin{cases} +200, & if \parallel \boldsymbol{p}_t - \boldsymbol{g}_t \parallel_2 < \varepsilon \\ -200, & if collision or D_{2T} > D \\ r_o \cdot r_a + 0.5 r_p + 0.5 r_d - 0.5, & otherwise \end{cases} \tag{6}$$

where r_o, r_a, r_p, and r_d are the orientation reward, forward-motion reward, backward-movement penalty, and distance-potential term (closed forms follow CD [5]); D is the maximum local-planning distance, and ε = 30 cm is the goal-activation threshold.

Goal sequence $\mathcal{G}$. In contrast to single-goal local-planning formulations, this work explicitly models an ordered sequence of subgoals:

$$\mathcal{G} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_K, \boldsymbol{g}_{K+1}) \tag{7}$$

where $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_K \in \mathbb{R}^2$ are the K target picking shelves visited in the prescribed order, and $\boldsymbol{g}_{K+1} \in \mathbb{R}^2$ is the delivery station. $\mathcal{G}$ is fixed throughout an episode. A trial is considered successful if and only if Eq. (8) holds:

$$\exists t_1 < t_2 < \cdots < t_{K+1}: \parallel \boldsymbol{p}_{t_i} - \boldsymbol{g}_i \parallel_2 < \varepsilon, \forall i \in \{1, \ldots, K+1\} \tag{8}$$

with threshold ε = 30 cm, provided that no collision occurs and the step counter does not exceed $t_{\max}$ = 2000. Failure modes are categorized as either COLLISION or TIMEOUT.

Transition kernel and discount factor. The control interval is Δt = 0.1 s, and the discount factor is γ = 0.99. The detailed kinematic equations of the robot and the disturbance model of dynamic obstacles are presented in Section 3.2.

3.2 Robot-Obstacle Dynamics

The transition kernel $\mathcal{T}$ is fully specified by the following equations on robot pose, obstacle motion, and LiDAR observation.

Robot kinematics. The robot pose $\boldsymbol{q}_t = [x_t, y_t, \theta_t]^T$ evolves under a differential-drive model:

$$\boldsymbol{q}_{t+1} = \boldsymbol{q}_t + \Delta t \cdot [V_l \cos\theta_t, V_l \sin\theta_t, V_a]^\top \tag{9}$$

(V_l, V_a) denote the linear and angular velocities sent to the chassis after low-pass filtering.

The target velocity (V_l^{ct}, V_a^{ct}) emitted by the policy is fused with the previous applied velocity (V_l^{rt}, V_a^{rt}) via a first-order low-pass filter:

$$V_l \leftarrow \kappa_l V_l^{ct} + (1-\kappa_l) V_l^{rt}, V_a \leftarrow \kappa_a V_a^{ct} + (1-\kappa_a) V_a^{rt} \tag{10}$$

Filter coefficients are $\kappa_l = 0.55$, $\kappa_a = 0.6$; velocity limits $V_l^{max} = 50$ cm/s, $V_a^{max} = 2$ rad/s; robot radius $r_c = 9$ cm.

Dynamic obstacles. Pedestrians, forklifts, and peer AMRs in real warehouses move unpredictably. We capture this with $N_o = 6$ obstacles $\{o^{(i)}\}_{i=1}^{N_o}$ whose positions and velocities evolve as:

$$\boldsymbol{p}_{t+1}^{(i)} = \boldsymbol{p}_t^{(i)} + \Delta t \cdot \boldsymbol{v}_t^{(i)} \tag{11}$$

$$\boldsymbol{v}_{t+1}^{(i)} = \Pi_{V_o^{max}}(\boldsymbol{v}_t^{(i)} + \delta\boldsymbol{v}_t^{(i)}), \delta\boldsymbol{v}_t^{(i)} \sim Unif(\{-1, 0, +1\}^2) \text{ cm/step} \tag{12}$$

The clipping operator $\Pi_{V_o^{max}}$ projects velocity onto the disk of radius $V_o^{max} = 30$ cm/s; the perturbation $\delta v_t^{(i)}$ is drawn independently each step from $\{-1, 0, +1\}^2$ (cm/step), equivalent to a per-axis acceleration in $[-10, +10]$ cm/s^2. Each obstacle's radius $r_o^{(i)}$ is resampled from [10, 14] cm at the start of every episode.

LiDAR observation. The 360° LiDAR scans $n_{raw} = 72$ beams up to $d_{max} = 300$ cm. Adjacent beams are min-pooled in groups of $n_{GN} = 3$, yielding the depth vector $\boldsymbol{l}_t \in \mathbb{R}^{n_g}$ with $n_g = 24$. Each component is then perturbed by independent ±1 cm uniform noise and clipped to the sensor's valid range:

$$\tilde{\boldsymbol{l}}_t = \Pi_{[0, d_{max}]}(\boldsymbol{l}_t + \boldsymbol{\eta}_t), \boldsymbol{\eta}_t \sim \text{Unif}([-1, +1])^{n_g} \text{ cm} \tag{13}$$

3.3 ColorDynamic Foundation

The dense traffic of moving obstacles demands a local planner that stays responsive and robust to disturbance, where DRL (Deep Reinforcement Learning)-based methods show a clear advantage [8,9]. In particular, CD [5], which extends the authors' earlier published planner Color [10] to dynamic obstacle settings, surpasses classical approaches (Artificial Potential Field, DWA) and learning-based competitors (Color, Deep Reinforcement Learning-based Velocity Obstacle [11]) on standard point-to-point navigation benchmarks. Following the "local perception with online decision-making" paradigm [7], it needs no global map or explicit obstacle-trajectory prediction, matching our warehouse constraints; we adopt it as the foundational local planner.

CD's Transqer network encodes the LiDAR temporal window $L(t)$ through a Transformer whose multi-head attention attends to several moving obstacles across frames simultaneously. This design maps directly onto our 6-obstacle concurrent setting. End-to-end inference over the 7-action discrete space completes in 1.2–1.3 ms/step, within the 100 ms control cycle. Before entering Transqer, $\boldsymbol{s}$t is normalized component-wise to [−1, 1] [5].

Action space. At each control step, Transqer selects one action from the discrete set $\mathcal{A} = \{a^{(0)}, \ldots, a^{(6)}\}$, mapped to a (V_l, V_a) velocity command (Supplementary Table S1). The seven actions span four linear-speed levels and angular commands of ±2 or 0 rad/s, and fall into three functional groups: in-place turning for posture adjustment in narrow aisles, cruise-speed forward and compound maneuvers for routine travel, and reverse or slow-down for dynamic avoidance and precise docking.

Reward function. The four dense shaping terms in Eq. (6) (an orientation reward r_o that rewards facing the goal, an action reward r_a favoring forward motion, a penalty r_p on backward maneuvers, and a distance-potential term r_d that rewards closing the distance to the subgoal) drive the policy toward fast, smooth, and collision-free arrival. Their closed forms and relative weighting follow CD [5] without modification.

Integration. The SW-DWNS scheduling layer (detailed in Section 3.4) interacts with CD solely through the active subgoal g_t in the state vector, leaving π_θ's internal parameters untouched ($T = 10$, $H = 8$, $L = 3$, linear-layer width 64, discount $\gamma = 0.99$). This interface convention decouples the scheduling logic from the underlying planner, directly enabling the controlled comparison in Section 4.2 that substitutes A* + DWA for π_θ.

3.4 Lightweight Scheduling Layer

Although the wave picking task comprises $K + 1$ ordered subgoals (Eq. (7)), at any single instant the robot still faces a standard point-to-point problem: the wave is sequential rather than joint. End-to-end alternatives that fold the entire workflow into one policy through memory modules, hierarchical controllers, or phase-conditioned rewards inflate the policy class and burden training without exploiting this structure. We therefore retain CD as the per-instant planner and wrap π_θ with a lightweight scheduling layer that, at each control step, exposes only the active subgoal g_t. The layer consists of a Goal Queue (GQ) and an Automatic Goal Switching (AGS) rule; the overall architecture is illustrated in Fig. 2.

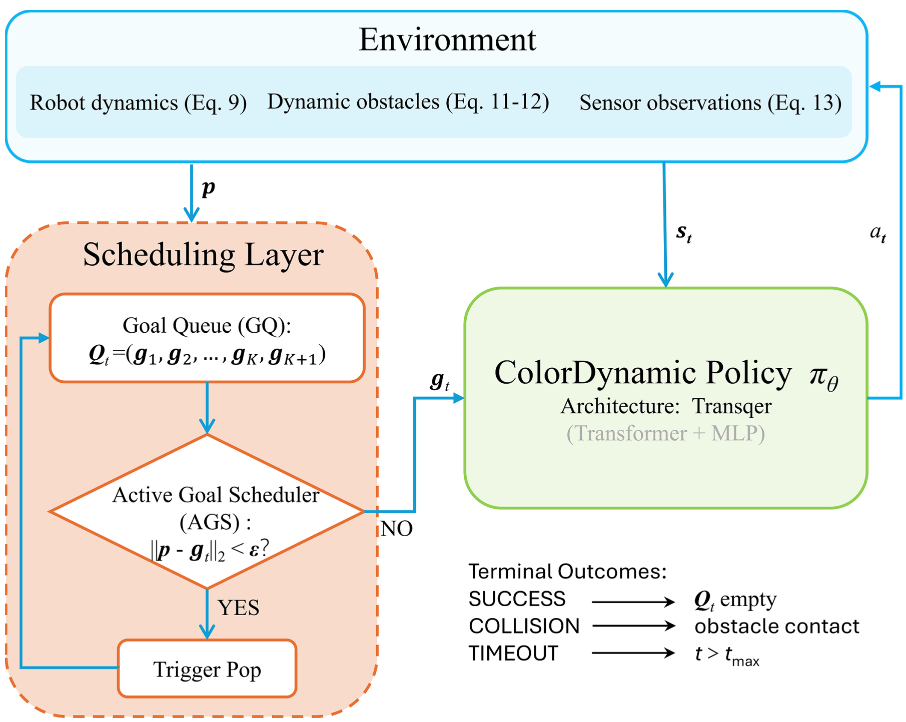


Figure 2: SW-DWNS architecture. The scheduling layer (GQ + AGS) and the CD policy π_θ operate as decoupled modules atop the environment, interacting solely through the active subgoal g_t.

GQ and switching rule. At episode start, GQ initializes Q_0 from $\mathcal{G}$ as a First-In-First-Out whose head is the active subgoal:

$$Q_0 = (g_1, g_2, \ldots, g_K, g_{K+1}), g_t = head(Q_t) \tag{14}$$

After every step, AGS checks whether the robot has entered the ε-neighborhood of the active subgoal; if so, the head is popped and the next goal becomes active:

$$Q_{t+1} = \begin{cases} pop\left(Q_t\right), & \| \boldsymbol{p}_{t+1} - \boldsymbol{g}_t \|_2 < \varepsilon \\ Q_t, & otherwise \end{cases} \tag{15}$$

Three termination outcomes follow: SUCCESS when Q_t becomes empty ($\boldsymbol{g}_{K+1}$ reached), COLLISION upon contact with an obstacle, and TIMEOUT upon exceeding $t_{\max}$ steps. The full procedure is given as Algorithm S1 in the Supplementary Materials.

Complexity. Each control step performs one $O(1)$ distance check and at most one $O(1)$ pop; the cumulative number of pops per episode is bounded by $K + 1$. Relative to the 1.2–1.3 ms/step Transqer inference, the scheduling overhead is negligible.

3.5 Pickup Threshold Selection

The pickup threshold ε in Eq. (15) is not a tunable hyper-parameter but a physically constrained quantity. It admits a lower bound, set by the agent radius and per-step sensing uncertainty, below which the robot cannot reliably register arrival; and an upper bound, set by half the minimum inter-shelf clearance, above which the trigger may fire from a neighboring aisle and miss the intended shelf. These two requirements give

$$r_c + \delta_\sigma \le \varepsilon \le \frac{1}{2} d_{\min} - r_c. \tag{16}$$

With $r_c = 9$ cm, $\delta_\sigma \approx 5$ cm and $d_{\min} \approx 80$ cm, Eq. (16) admits $\varepsilon \in [14, 31]$ cm, from which we adopt $\varepsilon = 30$ cm. The sweep $\varepsilon \in \{15, 30, 50, 80, 120\}$ cm in Section 4.7 verifies stability inside this band and breakdown outside it.

4 Results

4.1 Experimental Setup

We evaluate SW-DWNS on the three layouts of Section 2. Each layout is realized in an 800 cm × 800 cm workspace populated by six dynamic obstacles, whose kinematics, LiDAR observation model, and episode-level radius randomization over [10, 14] cm follow Eqs. (9)–(13). Every (Layout, K) cell is run for 100 independent episodes with $K \in \{1, \ldots, 5\}$, a step budget of $t_{\max} = 2000$, and a control interval of $\Delta t = 0.1$ s (Supplementary Table S2). Obstacle randomization enables velocity and radius randomization while disabling velocity scaling and type randomization.

4.2 Evaluation Metrics

Performance on every (Layout, K) cell is summarized by four metrics. Task Success Rate (TSR) is the fraction of episodes that complete the full ordered sequence of K target shelves and the delivery station within the step budget; Collision Rate (CR) and Task Timeout Rate (TOR) are the fractions terminated by contact with a dynamic obstacle and by exhausting $t_{\max}$, respectively; and Average Steps per Successful Episode (ASSE) is the mean step count over successful episodes, interpreted only together with TSR. TSR + CR + TOR = 1 in every cell, with each rate reported as rate ± h, h the half-width of the 95% Wilson confidence interval (n = 100); Wilson rather than normal-approximation intervals are used because the latter degrade near the 0% and 100% rates in our data. Each simulation step corresponds to 0.1 s of simulated time (10 Hz control).

4.3 Main Results: Superiority over Baselines

We compare SW-DWNS against two reference systems. A* + DWA pairs global A* search with reactive Dynamic-Window avoidance, the *de facto* classical baseline in warehouse navigation. CD-only ablates the scheduling layer entirely: the policy is queried with the delivery station as its sole goal, with no access to the ordered shelf queue. It is included as a necessity check, not a competitor: any success it scores reflects incidental proximity to a shelf, not intentional visitation. All three run on the same 100 episodes per (Layout, K) cell, $K \in \{1, \ldots, 5\}$, across the three layouts of Section 2; Table 1 reports success rates, Table 2 the failure-mode decomposition.

Table 1: Success-rate comparison against two baselines across three layouts ($K \in \{1, \ldots, 5\}$).

Success Rate (%)							
Layout	**Method**	**K = 1**	**K = 2**	**K = 3**	**K = 4**	**K = 5**	**Δ@ K = 5**
Dispersed	A* + DWA	40.0 ± 9.4	26.0 ± 8.5	21.0 ± 7.9	13.0 ± 6.6	4.0 ± 4.1	—
	CD-only	5.0 ± 4.5	0.0 ± 1.9	0.0 ± 1.9	0.0 ± 1.9	0.0 ± 1.9	—
	SW-DWNS	98.0 ± 3.2	97.0 ± 3.7	90.0 ± 6.0	86.0 ± 6.8	83.0 ± 7.3	+79.0
Compact-Simple	A* + DWA	58.0 ± 9.5	37.0 ± 9.3	26.0 ± 8.5	20.0 ± 7.8	7.0 ± 5.2	—
	CD-only	5.0 ± 4.5	0.0 ± 1.9	0.0 ± 1.9	0.0 ± 1.9	0.0 ± 1.9	—
	SW-DWNS	97.0 ± 3.7	95.0 ± 4.5	90.0 ± 6.0	87.0 ± 6.6	80.0 ± 7.8	+73.0
Compact-Complex	A* + DWA	55.0 ± 9.6	44.0 ± 9.6	29.0 ± 8.8	16.0 ± 7.2	6.0 ± 4.9	—
	CD-only	10.0 ± 6.0	0.0 ± 1.9	0.0 ± 1.9	0.0 ± 1.9	0.0 ± 1.9	—
	SW-DWNS	92.0 ± 5.4	85.0 ± 7.0	84.0 ± 7.2	79.0 ± 7.9	67.0 ± 9.1	+61.0
Average	A* + DWA			26.8			—
	CD-only			1.3			—
	SW-DWNS			87.3			+60.5

Note: Δ@ K = 5 reports the SW-DWNS advantage over A* + DWA at K = 5, in percentage points (pp).

Table 2: Failure-mode decomposition across three layouts ($K \in \{1, \ldots, 5\}$).

Collision Rate (%)						
Layout	**Method**	**K = 1**	**K = 2**	**K = 3**	**K = 4**	**K = 5**
Dispersed	A* + DWA	60.0 ± 9.4	74.0 ± 8.5	79.0 ± 7.9	86.0 ± 6.8	95.0 ± 4.5
	SW-DWNS	2.0 ± 3.2	3.0 ± 3.7	10.0 ± 6.0	11.0 ± 6.2	16.0 ± 7.2
Compact-Simple	A* + DWA	40.0 ± 9.4	58.0 ± 9.5	69.0 ± 8.9	73.0 ± 8.6	81.0 ± 7.6
	SW-DWNS	3.0 ± 3.7	5.0 ± 4.5	8.0 ± 5.4	9.0 ± 5.7	12.0 ± 6.4
Compact-Complex	A* + DWA	41.0 ± 9.5	52.0 ± 9.6	68.0 ± 9.0	73.0 ± 8.6	86.0 ± 6.8
	SW-DWNS	1.0 ± 2.6	5.0 ± 4.5	7.0 ± 5.2	11.0 ± 6.2	10.0 ± 6.0

(Continued)

Table 2 (continued)

Layout	Method	K = 1	K = 2	K = 3	K = 4	K = 5
Collision Rate (%)						
Average	A* + DWA			69.0		
	SW-DWNS			7.5		
Timeout Rate (%)						
Dispersed	A* + DWA	0.0 ± 1.8	0.0 ± 1.8	0.0 ± 1.8	1.0 ± 2.6	1.0 ± 2.6
	SW-DWNS	0.0 ± 1.8	0.0 ± 1.8	0.0 ± 1.8	3.0 ± 3.7	1.0 ± 2.6
Compact-Simple	A* + DWA	2.0 ± 3.2	5.0 ± 4.5	5.0 ± 4.5	7.0 ± 5.2	12.0 ± 6.4
	SW-DWNS	0.0 ± 1.8	0.0 ± 1.8	2.0 ± 3.2	4.0 ± 4.1	8.0 ± 5.4
Compact-Complex	A* + DWA	4.0 ± 4.1	4.0 ± 4.1	3.0 ± 3.7	11.0 ± 6.2	8.0 ± 5.4
	SW-DWNS	7.0 ± 5.2	10.0 ± 6.0	9.0 ± 5.7	10.0 ± 6.0	23.0 ± 8.2
Average	A* + DWA			4.2		
	SW-DWNS			5.1		

Note: CD-only is omitted from the failure breakdown: it does not attempt the picking task, so its collision and timeout rates are not comparable.

Across the full 3 × 5 grid (Table 1), SW-DWNS averages 87.3% success, against 26.8% for A* + DWA and 1.3% for CD-only, a full-grid advantage of +60.5 pp over A* + DWA, at least +61 pp in every layout, peaking at +79 pp in the Dispersed layout, where SW-DWNS preserves high success over five targets while A* + DWA accumulates reactive-avoidance failures. Even in the hardest cell (Compact-Complex, $K = 5$), SW-DWNS still reaches 67% vs. 6%. A* + DWA degrades sharply as K grows, every layout falling below 7% by $K = 5$: a static global route with reactive avoidance cannot maintain feasibility through sequential targets under stochastic traffic. CD-only flatlines at ≤1% for every $K \geq 2$; the slightly positive $K = 1$ row (5%–10%) comes from incidental proximity to a shelf along the direct route to the delivery station, not intentional visitation. The Wilson 95% intervals of SW-DWNS and A* + DWA do not overlap in any (Layout, K) cell. The two baselines bracket SW-DWNS from complementary sides: A* + DWA shows a competent classical planner cannot scale with K, and CD-only shows the DRL planner alone cannot execute the task at all. The scheduling layer is thus not a refinement but a necessity. A natural follow-up is whether this gain is purchased at the cost of safety; Table 2 decomposes the failure modes into collision and timeout.

Collision dominates A* + DWA's failures: 69.0% of its episodes end in collision on average (Table 2), exceeding 86% in its two worst cells (Dispersed $K = 5$; Compact-Complex $K = 5$). SW-DWNS averages 7.5% and never exceeds 16% in any cell, a nearly tenfold reduction in collision failures. The success advantage of SW-DWNS is therefore not purchased through risk; rather, it reflects a planner that has learned to interact with moving obstacles instead of merely avoiding them. Timeouts stay low for both A* + DWA and SW-DWNS on average (4.2% and 5.1%); the rise for SW-DWNS at high K in Compact-Complex (23.0 ± 8.2% at $K = 5$) reflects episode lengths approaching the step budget, not a different failure mode. Supplementary Fig. S1 shows representative $K = 2$ successes across the three layouts.

4.4 Path-Length Efficiency

How much additional travel does SW-DWNS spend to achieve it? We analyze the success-conditional step counts from the main grid (Supplementary Table S3); at 10 Hz (0.1 s per step), these translate to simulated time (e.g., 600 steps ≈ 60 s). CD-only is omitted because its strict success rate is too low (1.3% full-grid average) for the success-conditional statistic to be meaningful.

Read at face value, A* + DWA appears more efficient: its full-grid average is 420.9 steps (≈42.1 s) per successful episode vs. 598.4 steps (≈59.8 s) for SW-DWNS, an apparent overhead of roughly 18 s. The gap widens with K: at $K = 5$, SW-DWNS averages 818–897 steps across layouts while A* + DWA averages 556–788. This is the honest cost: SW-DWNS does not produce shorter paths than a classical static planner when both succeed.

However, the two rows are not drawn from the same population: A* + DWA's average is conditioned on the 26.8% of episodes where its static plan held, SW-DWNS's on 87.3%, including the harder cases A* + DWA never reaches. In the sparsest cells (Compact-Simple and Dispersed at $K = 5$), A* + DWA's mean rests on only 7 and 4 successful episodes, the lucky-traffic tail. The conditional comparison therefore understates A* + DWA's true overhead and overstates SW-DWNS's.

SW-DWNS's longer paths fit the mechanism of Section 4.3: a planner that interacts with moving obstacles (detouring around incoming traffic, waiting briefly when a collision course develops) rather than committing to a precomputed route. Under the 2000-step (200 s) budget, its full-grid average of 59.8 s leaves ample headroom: it trades modest extra travel time for a more than threefold success rate over A* + DWA.

4.5 Robustness across Layout Difficulty and Pickup Horizon

We now test whether this performance pattern matches the difficulty signature predicted in Section 2. We use a separate trace run of 100 episodes per cell for $K \in \{3, 5\}$, recording success rate, realized path length, and simulated task time. Its success rates differ from the main grid (Table 1) by at most 7 pp and stay within overlapping Wilson intervals, so the analysis below focuses on patterns rather than point estimates.

Three patterns in Table 3 fit this difficulty signature. K-sensitivity grows with layout difficulty: success drops from $K = 3$ to $K = 5$ by −5 pp (Dispersed), −11 pp (Compact-Simple), and −18 pp (Compact-Complex). Each extra pickup target erodes completion more in harder layouts. K is a difficulty amplifier, not a fixed-cost addition, and the predicted layout ordering holds empirically.

Table 3: Trace-run performance summary.

Layout	K	Success Rate (%)	Path Length (cm)	Simulated Time (s)	Effective Speed (cm/s)
Dispersed	3	91.0 [83.8, 95.2]	1928	55.6	34.7
	5	86.0 [77.9, 91.5]	3149	84.6	37.2
Compact-Simple	3	90.0 [82.6, 94.5]	2156	54.2	39.8
	5	79.0 [70.0, 85.8]	2987	85.8	34.8
Compact-Complex	3	78.0 [68.9, 85.0]	2421	74.0	32.7
	5	60.0 [50.2, 69.1]	3371	109.8	30.7

Note: Path length, simulated time, and effective speed are over successful episodes only; effective speed = path length ÷ simulated time. Brackets: 95% Wilson intervals.

Layout difficulty asserts itself most clearly at high K: the three layouts span 13 points at $K = 3$ (78%–91%) but 26 points at $K = 5$ (60%–86%), in strict order Dispersed > Compact-Simple > Compact-Complex, a hardness scale latent under low task load and revealed once enough operations are chained to reach each layout's worst-case configurations.

Effective speed also declines with layout difficulty at high K, beyond what path length alone reveals. At $K = 5$ it falls monotonically from 37.2 cm/s in Dispersed to 34.8 in Compact-Simple and 30.7 in Compact-Complex, a roughly 17% drop; averaged over $K = 3$ and $K = 5$, Compact-Complex remains the slowest layout. Between Dispersed and Compact-Complex at $K = 5$, path length rises only ~7% while simulated time rises ~30%, a gap path geometry alone cannot explain. Under harder layouts the planner waits, slows, or yields to incoming traffic more often, covering routes of similar length at lower throughput, the trace-level counterpart of the "interacts with rather than avoids" reading above. SW-DWNS thus responds to layout difficulty and pickup horizon progressively and predictably, not catastrophically at any single threshold. We next isolate these environmental factors.

4.6 Factor-Wise Disturbance Ablation

We use a 2^3 full factorial design on the Compact-Complex layout at $K = 3$ (100 episodes per cell, 8 cells total; Table 4), independently toggling three deployment-time perturbation factors: sensing noise (N), obstacle motion (D), and obstacle-size randomization (R). The nominal configuration N1·D1·R1 matches the Section 4.3 setting exactly; its 84% success rate coincides with the corresponding cell in Table 1, providing an independent consistency check between the factor-ablation run and the main grid.

Table 4: Factor-wise ablation on (compact-complex, $K = 3$).

Configuration	N	D	R	Success (%)	Collision (%)	Timeout (%)
N1·D1·R1	✓	✓	✓	84.0 [75.6, 89.9]	7.0 [3.4, 13.8]	9.0 [4.8, 16.2]
N1·D1·R0	✓	✓	–	69.0 [59.4, 77.2]	14.0 [8.5, 22.1]	17.0 [10.9, 25.6]
N0·D1·R1	–	✓	✓	75.0 [65.7, 82.5]	12.0 [7.0, 19.8]	13.0 [7.8, 21.0]
N0·D1·R0	–	✓	–	80.0 [71.1, 86.7]	8.0 [4.1, 15.0]	12.0 [7.0, 19.8]
N1·D0·R1	✓	–	✓	47.0 [37.5, 56.7]	2.0 [0.6, 7.0]	51.0 [41.4, 60.6]
N1·D0·R0	✓	–	–	44.0 [34.7, 53.8]	0.0 [0.0, 3.7]	56.0 [46.2, 65.3]
N0·D0·R1	–	–	✓	45.0 [35.6, 54.8]	0.0 [0.0, 3.7]	55.0 [45.2, 64.4]
N0·D0·R0	–	–	–	40.0 [30.9, 49.8]	0.0 [0.0, 3.7]	60.0 [50.2, 69.1]

Note: N = sensing noise; D = dynamic obstacle motion; R = obstacle-size randomization. ✓ = factor present (matches the nominal evaluation setting); – = factor disabled. D = 0 does not remove obstacles; it sets their maximum velocity to zero and disables velocity randomization, yielding static obstacles.

Main effects are computed by averaging success rate over the levels of the other two factors.

The dynamics factor dominates. Disabling obstacle motion (D = 0) cuts average success from 77% to 44%, a +33 pp main effect, more than seven times the next factor. Sensing noise is negligible (+1 pp), and obstacle-size randomization gives a small positive shift (+4.5 pp), comparable to a single cell's Wilson half-width. SW-DWNS is thus comparatively insensitive to sensor noise and size variability, but hinges on the presence of dynamic agents.

The failure mode flips under D = 0. In every D = 1 cell, collision is the dominant failure (7%–14%) and timeout remains low (9%–17%). In every D = 0 cell, the pattern inverts: collisions collapse to 0%–2%, while timeouts surge to 51%–60%. The planner now wastes its 200 s budget without reaching the targets, despite a near-zero risk of physical contact. This is the diagnostic signature of out-of-distribution behavior: the static-obstacle world is geometrically simpler than the nominal environment yet operationally harder for the policy to solve.

Within this stress setting, SW-DWNS thus tolerates sensor and obstacle-shape variability without retraining (these perturbations leave the core dynamics assumption intact) while its competence is load-bearing on that assumption; we discuss this out-of-distribution behavior and its deployment implications in Section 5.2.

4.7 Pickup Threshold Sensitivity

Section 3.5 derived an admissible band $\varepsilon \in [14, 31]$ cm from the sensor-uncertainty footprint (lower bound) and the inter-shelf clearance (upper bound); the nominal $\varepsilon = 30$ cm sits at its upper end. This section sweeps $\varepsilon \in \{15, 30, 50, 80, 120\}$ cm on the same stress cell used in Section 4.6 (Compact-Complex, $K = 3$, 100 episodes each) to verify that the success rates reported throughout Section 4 are neither artificially inflated nor artificially deflated by the threshold choice (Table 5).

Table 5: Pickup-threshold sensitivity on (compact-complex, $K = 3$).

ε (cm)	**Success Rate (%)**	**Collision (%)**	**Timeout (%)**	**Avg Steps (Success)**
15	74.0 [64.6, 81.6]	12.0 [7.0, 19.8]	14.0 [8.5, 22.1]	714
30	82.0 [73.3, 88.3]	5.0 [2.2, 11.2]	13.0 [7.8, 21.0]	631
50	88.0 [80.2, 93.0]	4.0 [1.6, 9.8]	8.0 [4.1, 15.0]	656
80	82.0 [73.3, 88.3]	5.0 [2.2, 11.2]	13.0 [7.8, 21.0]	549
120	86.0 [77.9, 91.5]	6.0 [2.8, 12.5]	8.0 [4.1, 15.0]	509

Note: The admissible band is $\varepsilon \in [14, 31]$ cm; $\varepsilon \geq 50$ violates the upper bound and $\varepsilon \geq 80$ also exceeds the minimum inter-shelf clearance ($d_{\min} = 80$ cm).

Read along the success column alone, the sweep would misleadingly suggest that values above the band remain acceptable or even preferable: $\varepsilon = 15$ underperforms (74%), $\varepsilon = 30$ reaches 82%, $\varepsilon = 50$ peaks at 88%, and $\varepsilon \in \{80, 120\}$ stay at or above 82%. The corresponding collision and timeout patterns reinforce this surface reading: $\varepsilon = 15$ incurs both higher collisions (12%) and higher timeouts (14%) as the agent overshoots a too-tight target and is forced into corrective maneuvers. If success rate were the only criterion, $\varepsilon = 50$ would appear to be the operating point of choice.

The step-count column reveals a different story. Average steps per successful episode fall from 631 at $\varepsilon = 30$ to 549 at $\varepsilon = 80$ and 509 at $\varepsilon = 120$, a roughly 19% reduction. This is not improved policy efficiency: the policy is identical across the sweep. With a larger acceptance region, the same trajectory is credited for "arrival" tens of centimeters short of the shelf. The step-count decline is therefore the operational signature of success inflation rather than evidence of better navigation.

The geometric bound prohibits this regime on principled grounds. $\varepsilon = 50$ is already geometrically inadmissible, although its step count has not yet collapsed; $\varepsilon = 80$ and $\varepsilon = 120$ make the inflation mechanism visible. Under such ε, the reported success rate is a function of the threshold not the planner. The 88% peak at $\varepsilon = 50$ and the 86% at $\varepsilon = 120$ are not stronger evidence of capability than the 82% at $\varepsilon = 30$; they are weaker evidence, achieved against a relaxed criterion.

The choice $\varepsilon = 30$ cm is therefore not a tuning hyperparameter but a principled selection near the upper end of the admissible band, maximizing tolerance without violating the inter-shelf constraint.

The Sections 4.3–4.6 results are reported under this admissible setting, avoiding both the corrective-overhead regime represented by $\varepsilon = 15$ and the threshold-inflation regime that becomes visible at $\varepsilon \geq 80$. Tightening ε within the admissible band is unlikely to erase the reported gains, since $\varepsilon = 15$ still reaches 74% on the stress cell; loosening ε outside the band, by contrast, makes success increasingly dependent on the benchmark definition rather than the planner. We discuss the broader implications of threshold choice for benchmarking practice in Section 5.

5 Discussion

5.1 Scheduling as a Decoupled Task Layer

The methodological contribution of SW-DWNS is architectural rather than a new local-control policy. CD is used as a pretrained point-to-point dynamic navigator; SW-DWNS adds an explicit task-state layer above it, converting an ordered shelf queue and a delivery station into the single time-varying goal consumed by the local planner. In this sense, SW-DWNS addresses the temporal logic of warehouse picking without retraining the navigation policy.

This differs from hierarchical reinforcement learning or the Options Framework [6]. SW-DWNS does not learn option policies, termination functions, or a high-level value function. Its high-level policy is the known order sequence, and its termination rule is a geometric, auditable threshold derived in Section 3.5. The result is less expressive than learned hierarchy, but also lighter, more stable, and easier to reproduce when the task already provides an explicit goal order.

The ablations in Section 4.3 show that this layer is load-bearing. CD-only, the same policy weights without access to the ordered shelf queue, averages only 1.3% success; with the scheduling layer it reaches 87.3%. The A* + DWA baseline further shows the benefit is not from merely adding any sequential wrapper: it averages 26.8% success and fails primarily through collision. SW-DWNS therefore occupies a middle ground: it preserves a learned dynamic-obstacle local controller, but supplies the missing temporal task semantics explicitly.

This design has clear boundaries. In principle, any local planner with a compatible state–goal interface could replace CD, but this paper validates the interface only with CD; generality across MPC, DWA, or other learned point-to-point navigators remains future work. The present claim is narrower and better supported: when a competent pretrained local policy exists and the task has an explicit ordered goal structure, a decoupled scheduling layer can extend the policy to sequential warehouse picking without the cost and instability of end-to-end retraining.

5.2 Dynamics Assumption and Deployment Envelope

Section 4.6 isolated obstacle motion as the dominant environmental factor: disabling it costs 33 percentage points of success and inverts the failure mode from collision to timeout. This pattern is consistent with an out-of-distribution shift in the local policy's operating regime, and it identifies the assumption on which the deployment envelope of SW-DWNS rests.

Under CD's operating regime, moving obstacles carry information, so the policy's waiting and yielding is rational: hesitation in front of a moving obstacle is rewarded by the obstacle moving away. Under $D = 0$ the same behavior gets no progress signal in return, and the planner exhausts its step budget without abandoning a futile course. The failure is temporal rather than geometric (collisions near zero, timeouts 51%–60%) and reflects situations the regime never required the policy to recover from.

The deployment envelope of SW-DWNS is therefore narrower than its main-grid success rates would suggest. The system can be expected to perform competently in warehouses where the dominant traffic comes from other AMRs, active forklifts, and human pickers in motion; it may degrade into a distinctive, detectable signature when obstacles are predominantly stationary. This signature is itself useful: it allows the "policy missing competence" failure mode (D = 0) to be distinguished online from the "policy meeting unexpected dynamics" mode (e.g., traffic exceeding training density), without requiring detailed trajectory labelling.

Two limits to this reading must be stated explicitly. First, the 33 pp effect is measured on a single stress cell (Compact-Complex, $K = 3$); the magnitude in other layouts and pickup horizons may differ. Second, the dynamic obstacles used in both training and evaluation follow simple kinematic models that do not capture richer human or forklift behavior. Because these behaviors preserve motion but alter its intent and reactivity, the qualitative claim that dynamics matters is likely to remain relevant, whereas the reported success rates should not be extrapolated.

The architectural choice in Section 5.1 makes this critical operational assumption structurally visible: because the local policy is fixed and the scheduling layer is policy-blind, SW-DWNS contains no mechanism to compensate for an out-of-distribution local environment. Two practical mitigations follow: retraining CD with a static-obstacle component in its training data or coupling the deployed system with an explicit replanning trigger. Both expand the deployment envelope without abandoning the decoupling principle that gives SW-DWNS its main strength.

5.3 Pickup Thresholds and Benchmarking Validity

The pickup-threshold sweep (Section 4.7) showed that raising ε from 30 to 120 cm lifts reported success from 82% to 86% while cutting average steps from 631 to 509, though the policy is unchanged: neighboring acceptance regions overlap, so success becomes a function of the threshold, not the planner.

The observation is not specific to ε. Any metric in mobile-robotics benchmarking that depends on a tolerance (pickup radius, goal-reaching tolerance, episode timeout, collision exclusion zone) couples the reported number to a parameter that can be tightened or loosened by the experimenter. When the tolerance is chosen for convenience and reported as a single value, the resulting success rate has limited operational meaning across studies.

We do not propose a universal benchmarking standard. We do, however, suggest that for warehouse navigation benchmarks specifically, three practices improve cross-study comparability. First, tolerance thresholds should be derived from problem geometry where possible (from sensor uncertainty, robot footprint, target spacing, or task semantics) rather than chosen by convenience. Second, the admissibility argument should be reported alongside the threshold, so that reviewers and downstream users can verify the threshold is not silently inflating the criterion. Third, where multiple admissible thresholds are plausible, a sensitivity sweep and a companion metric such as successful-episode step count should be reported, so that criterion relaxation can be distinguished from genuine policy improvement.

This paper follows these practices for ε. With the admissible band $\varepsilon \in [14, 31]$ cm of Section 3.5 placing the nominal $\varepsilon = 30$ cm at its upper end, the headline 87.3% in Section 4.3 is measured at the largest geometrically admissible tolerance, permissive within the task semantics, but not inflated beyond them.

6 Conclusions and Future Work

6.1 Conclusions

This paper presented SW-DWNS, a lightweight scheduling framework that places an explicit task-state layer above a pretrained dynamic-obstacle local planner, extending it from point-to-point navigation to sequential warehouse picking without retraining. Across three layouts and pickup horizons $K \in \{1, \ldots, 5\}$, SW-DWNS achieved an average success rate of 87.3%, compared with 26.8% for A* + DWA and 1.3% for the

CD-only ablation. The gain was not obtained by trading off safety: collision failures were reduced from 69.0% for A* + DWA to 7.5% for SW-DWNS. Factor-wise ablation revealed an uneven robustness across factors: SW-DWNS is comparatively insensitive to sensing noise (+1 pp main effect) and obstacle-size randomization (+4.5 pp), but its competence depends strongly on obstacle motion (+33 pp). Finally, the pickup-threshold sweep showed that geometrically justified arrival criteria are necessary for fair evaluation, since overly large thresholds can inflate reported success without improving the planner.

6.2 Limitations and Future Work

The present study has four main limitations, each of which defines a direction for future work.

1. Dynamic Target-Sequence Management. The target sequences in our experiments were fixed upon issuance, and the system did not account for dynamic scenarios such as the real-time insertion of new targets or changes to existing ones. Future work should support mutable queues, including online order insertion, cancellation, reprioritization, and mid-mission reassignment.
2. Broader Obstacle Distributions. The simulated obstacles capture controlled dynamic interference, yet they stop short of modeling the full range of human or forklift behaviors, such as Optimal Reciprocal Collision Avoidance-style social navigation [12], sudden stops [13], or mechanical failures. Future work should broaden the obstacle distribution to include static blockers and these richer behaviors.
3. Multi-Robot Coordination. The current system focuses exclusively on single-robot operations and does not address the complexities of multi-robot collaboration and resource contention [14,15]. Extending the framework to multi-robot warehouses requires mechanisms for task allocation, shared-aisle contention, communication constraints, and distributed scheduling [16].
4. Real-World Hardware Validation. All experiments in this paper are simulation-based. Deploying SW-DWNS on physical warehouse robots is necessary to evaluate wheel slip, calibration drift, sensing failures, surface variation, and other operational effects.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Conceptualization, Xianhui Fan; methodology, Xianhui Fan; software, Xianhui Fan; validation, Xianhui Fan; formal analysis, Xianhui Fan; investigation, Xianhui Fan and Yuxuan Zhai; resources, Zongwei Li and Zhenyu Li; data curation, Xianhui Fan; writing—original draft preparation, Xianhui Fan; writing—review and editing, Zongwei Li, Yuxuan Zhai and Zhenyu Li; visualization, Xianhui Fan; supervision, Zongwei Li; project administration, Zongwei Li and Zhenyu Li; funding acquisition, Zongwei Li. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that all data supporting the findings of this study are included in the paper.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Supplementary Materials: The supplementary material is available online at https://www.techscience.com/doi/10.32604/cmc.2026.083491/s1.

References

1. Fu J, Zhang H, Jian Z, Jiang L. Review on AGV scheduling optimization. J Syst Simul. 2020;32(9):1664–75. (In Chinese). doi:10.16182/j.issn1004731x.joss.19-0042.

2. Hu E, He J, Shen S. A dynamic integrated scheduling method based on hierarchical planning for heterogeneous AGV fleets in warehouses. Front Neurorobot. 2023;16:1053067. doi:10.3389/fnbot.2022.1053067.
3. Nguyen TMB, Nguyen THH, Do TNH. Optimized metaheuristic strategies for addressing the multi-picker robot routing problem in 3D warehouse operations. Comput Mater Contin. 2025;84(3):5063–76. doi:10.32604/cmc.2025.064610.
4. Tariq MT, Hussain Y, Wang C. Robust mobile robot path planning via LLM-based dynamic waypoint generation. Expert Syst Appl. 2025;282:127600. doi:10.1016/j.eswa.2025.127600.
5. Xin J, Liang Z, Zhang Z, Wang P, Li N. ColorDynamic: generalizable, scalable, real-time, end-to-end local planner for unstructured and dynamic environments. arXiv:2502.19892. 2025. doi:10.48550/arxiv.2502.19892.
6. Sutton RS, Precup D, Singh S. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif Intell. 1999;112(1):181–211. doi:10.1016/S0004-3702(99)00052-1.
7. Tang H, Wang A, Xue F, Yang J, Cao Y. A novel hierarchical soft actor-critic algorithm for multi-logistics robots task allocation. IEEE Access. 2021;9:42568–82. doi:10.1109/ACCESS.2021.3062457.
8. Li K, Zhang T, Wang R. Deep reinforcement learning for multiobjective optimization. IEEE Trans Cybern. 2021;51(6):3103–14. doi:10.1109/TCYB.2020.2977661.
9. Lin Y, Hue G, Wang L, Li Q, Zhu J. A multi-AGV routing planning method based on deep reinforcement learning and recurrent neural network. IEEE/CAA J Autom Sin. 2024;11(7):1720–2. doi:10.1109/JAS.2023.123300.
10. Xin J, Kim J, Li Z, Li N. Train a real-world local path planner in one hour via partially decoupled reinforcement learning and vectorized diversity. Eng Appl Artif Intell. 2025;141(11):109726. doi:10.1016/j.engappai.2024.109726.
11. Xie Z, Dames P. DRL-VO: learning to navigate through crowded dynamic scenes using velocity obstacles. IEEE Trans Robot. 2023;39(4):2700–19. doi:10.1109/TRO.2023.3257549.
12. van den Berg J, Guy SJ, Lin M, Manocha D. Reciprocal n-body collision avoidance. In: Pradalier C, Siegwart R, Hirzinger G, editors. Robotics research. Berlin, Heidelberg: Springer; 2011. p. 3–19. doi:10.1007/978-3-642-19457-3_1.
13. Wang H, Xu J, Yan G. Research on pedestrian avoidance strategy for AGV based on deep reinforcement learning. J Syst Simul. 2025;37(3):595–606. (In Chinese). doi:10.16182/j.issn1004731x.joss.24-0088.
14. Jones S, Hauert S. Distributed spatial awareness for robot swarms. Auton Robots. 2025;49(4):41. doi:10.1007/s10514-025-10228-1.
15. Krnjaic A, Steleac RD, Thomas JD, Papoudakis G, Schäfer L, To AWK, et al. Scalable multi-agent reinforcement learning for warehouse logistics with robotic and human co-workers. In: Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2024 Oct 14–18; Abu Dhabi, United Arab Emirates. p. 677–84. doi:10.1109/IROS58592.2024.10802813.
16. Chen F, Li L, Tang J. Multi-agent task assignment and path finding for heterogeneous AGV systems. Expert Syst Appl. 2026;307(5):130973. doi:10.1016/j.eswa.2025.130973.