Energy-Constrained Hybrid Repair for Lifelong Multi-Agent Path Finding in Smart Warehouses

Riyang Luo , Can Lu , Jin He *

Posted Date: 20 May 2026

doi: 10.20944/preprints202605.1283.v1

Keywords: lifelong multi-agent path finding; smart warehouse; autonomous mobile robot; energy-aware planning; charging constraints; hybrid repair; learning-compatible planning

 doi:10.20944/preprints202605.1283.v1

Article

Energy-Constrained Hybrid Repair for Lifelong Multi-Agent Path Finding in Smart Warehouses

Riyang Luo, Can Lu and Jin He *

School of Mechanical and Power Engineering, Nanjing Tech University, Nanjing 211816, China
* Correspondence: jinhe@njtech.edu.cn

Abstract

Smart warehouses rely on fleets of autonomous mobile robots that must continually assign tasks, plan paths, avoid collisions, and maintain battery energy. Existing lifelong multi-agent path finding studies often emphasize travel cost or makespan, while practical deployments also involve charging, payload-dependent energy use, turning and waiting costs, and congestion. This paper presents an energy-constrained hybrid repair framework for lifelong multi-agent path finding in warehouses. The method combines a risk-aware graph representation, search-based safety repair, and learning-compatible policy modules, and it is evaluated in a reproducible Python simulator. We compare independent and prioritized A* planning, windowed cooperative planning, large-neighborhood repair, lazy configuration search, bounded conflict-based search, the proposed repair variant, and a graph neural learning baseline. The virtual evaluation reports raw task completion separately from energy-feasible completion, together with collision, charger-conflict, energy, scalability, ablation, sensitivity, and case-study measures. On 40×40 warehouse maps with 20 robots, large-neighborhood repair improves raw success from 0.345 to 0.468 relative to windowed cooperative planning and reduces energy per completed task from 369.34 to 276.83. The proposed repair variant reduces several conflict measures, but throughput remains problem-dependent. The results support energy-aware repair as a practical direction for warehouse robot coordination.

Keywords: lifelong multi-agent path finding; smart warehouse; autonomous mobile robot; energy-aware planning; charging constraints; hybrid repair; learning-compatible planning

1. Introduction

Smart warehouses are increasingly operated by fleets of autonomous mobile robots (AMRs) that transport goods between storage shelves, picking stations, buffers, chargers, and delivery zones. In such systems, path planning is not a one-shot routing problem. Robots continuously receive new tasks, carry different payloads, deplete battery energy, wait at congested aisle intersections, and compete for charging stations. These properties lead naturally to lifelong multi-agent path finding (MAPF), an online variant of MAPF in which each robot is repeatedly assigned new goals over a long execution horizon.

Classical MAPF solvers provide strong algorithmic foundations. Conflict-based search (CBS) is optimal but expensive for large numbers of agents [1]. WHCA* and prioritized planning are faster and easier to deploy online [2], but they can be sensitive to priority order and local congestion. MAPF-LNS2 repairs large neighborhoods of conflicting paths and has shown strong empirical scalability [3]. LaCAM-style search further improves scalability through lazy successor generation [4]. Recent studies in *Electronics* have also examined multi-agent collaborative planning with multiple meeting points and autonomous patrol robot routing, reflecting the continuing practical relevance of coordinated robot path planning [5,6]. These methods remain essential baselines for any credible warehouse MAPF study, together with foundational shortest-path search and MAPF formulations [7–10].

 doi:10.20944/preprints202605.1283.v1

Learning-based MAPF has developed in parallel. PRIMAL and PRIMAL2 use reinforcement learning and imitation learning to learn decentralized coordination policies [11,12]. More recent graph, attention, and multi-agent reinforcement learning methods use communication or message passing to capture interactions beyond a single local field of view [13–15]. Benchmark platforms such as POGEMA emphasize the need for fair comparison between classical, learned, and hybrid methods [16]. Recent warehouse lifelong MAPF studies also indicate that learned guidance can help priority assignment, while the advantage of learning over strong search baselines remains problem-dependent [17].

This paper studies an energy-constrained, risk-aware version of warehouse lifelong MAPF. Instead of optimizing only path length or makespan, we explicitly include motion energy, turning energy, waiting energy, payload-dependent energy, congestion energy, battery capacity, low-battery routing, and charging station behavior. We then propose ECR-HR, an energy-constrained hybrid repair framework with a learning-compatible risk graph interface. The main executable variant uses search-based repair to prevent unsafe execution; a graph transformer policy and PPO pipeline are implemented and evaluated as a preliminary learning component rather than as the primary source of empirical superiority.

The contributions are:

- We formulate an energy-constrained lifelong MAPF problem for smart warehouses, incorporating motion energy, turning energy, waiting energy, payload cost, congestion cost, battery constraints, and charging-station capacity.
- We develop a reproducible warehouse lifelong MAPF simulation platform including task generation, AMR energy modeling, charging behavior, collision checking, and multiple online planning baselines.
- We propose a risk-aware hybrid repair framework in which candidate actions are checked and repaired using local search to reduce unsafe execution.
- We provide a comparative evaluation of classical, repair-based, lazy-search, and learning-compatible MAPF methods under warehouse energy constraints.
- We analyze the trade-off among throughput, energy efficiency, collision risk, battery safety, charger conflicts, and computation time using ablation studies, sensitivity analysis, scalability analysis, and case studies.

2. Related Work

2.1. Classical MAPF

CBS resolves conflicts by branching on constraints and planning individual paths under those constraints [1]. It is a cornerstone of optimal MAPF, complementing earlier optimal and suboptimal search formulations such as operator decomposition, increasing-cost tree search, graph decomposition, subdimensional expansion, and SAT/optimization-style encodings [18–23]. ECBS introduces bounded suboptimality and focal search to improve runtime [24], while ICBS, CBS heuristics, highways, disjoint splitting, and explicit-estimation variants further improve conflict handling [25–29]. Prioritized planning orders agents and reserves space-time cells for higher-priority agents; it is fast but incomplete. WHCA* improves online usability through a rolling reservation window [2]. These methods are attractive for warehouses because they are interpretable and can be adapted to online replanning.

2.2. Large Neighborhood and Lazy Search

MAPF-LNS2 repeatedly selects neighborhoods of agents and repairs their paths using local replanning [3]. This destroy-and-repair strategy is often effective in dense maps because it avoids replanning the whole system. LaCAM and LaCAM* use lazy successor generation over configurations to improve scalability [4]. Priority inheritance with backtracking (PIBT), Push-and-Swap, and consistent prioritization are influential iterative or prioritized MAPF mechanisms for dense online

 doi:10.20944/preprints202605.1283.v1

coordination [30–32]. Lifelong pickup-and-delivery MAPF has also been studied as a warehouse task-assignment and path-planning problem [33]. In this work, we implement online, bounded versions of these ideas that output the next lifelong action rather than solving a complete offline MAPF instance.

2.3. Learning-Based MAPF

PRIMAL introduced a combination of imitation and reinforcement learning for decentralized pathfinding [11]. PRIMAL2 extended the idea to lifelong MAPF [12]. PPO provides a standard policy-gradient backbone for such training [34]. Graph neural networks and attention models are natural for MAPF because agents form dynamic interaction graphs [35–38]. Recent *Applied Sciences* work further shows the growing use of reinforcement learning and multi-objective search for unmanned ground vehicle and mobile robot path planning [39,40]. However, learned policies can generate unsafe actions, especially under distribution shift. This motivates hybrid approaches in which learning proposes actions or priorities and search repairs unsafe decisions.

2.4. Energy-Aware Warehouse Planning

Real AMRs consume different amounts of energy when moving, turning, waiting, carrying loads, or operating in congested regions. They also need charging decisions and may block charging stations. These constraints are often simplified in benchmark MAPF, and continuous kinematic constraints further complicate direct deployment [41]. Recent robot path-planning studies in MDPI journals cover heuristic fusion, autonomous patrol routing, and broader construction-robot planning trends, but energy-aware lifelong warehouse coordination remains less directly addressed [6,42,43]. Industrial warehouse automation studies show that coordination, storage layout, and robot dispatch jointly determine throughput and congestion [44–46]. Our formulation adds energy and charging dynamics while retaining a grid-based representation that supports controlled and reproducible simulation studies.

2.5. Qualitative Comparison

Table 1 summarizes representative method families and clarifies the gap addressed by this work. The key distinction is that ECR-HR treats energy, risk, charging, and repair as coupled online decisions rather than isolated post-processing metrics.

Table 1. Qualitative comparison with representative MAPF method families.

Category	Methods	Main strength	Gap addressed here
Optimal search	CBS / ECBS	Strong conflict reasoning	Used as bounded baselines and repair inspiration
Rolling planning	WHCA*	Fast online reservations	Used for warm-start and local repair
Large-neighborhood search	MAPF-LNS2	Strong scalable repair	Extended with energy/risk evaluation
Lazy configuration search	LaCAM / LaCAM*	Scalable lazy successors	Adapted with warehouse energy metrics
Learning-based MAPF	PRIMAL / PRIMAL2	Distributed policies	Combined with explicit safety repair
Graph/attention models	GNN / Transformer	Models robot interaction	Used for risk-aware graph features

 doi:10.20944/preprints202605.1283.v1

3. Problem Formulation

Following standard MAPF graph notation [23,33], the warehouse is represented by a grid graph

$$G = (V, E), \tag{1}$$

where V denotes traversable cells and E connects four-neighbor cells. Obstacles O represent shelves, walls, or temporary blocked aisles. Let $R = \{r_1, \ldots, r_N\}$ be robots, $T = \{\tau_1, \ldots, \tau_M\}$ be pickup-and-delivery tasks, and $C = \{c_1, \ldots, c_K\}$ be charging stations.

Each robot state is

$$s_i^t = [p_i^t, g_i^t, b_i^t, v_i^t, l_i^t, o_i^t], \tag{2}$$

where p_i^t is position, g_i^t is current target, b_i^t is battery level, v_i^t is direction, l_i^t is payload state, and o_i^t is local observation. The action set is

$$a_i^t \in \{\text{up}, \text{down}, \text{left}, \text{right}, \text{wait}, \text{charge}\}. \tag{3}$$

The objective combines efficiency, energy, safety, and throughput:

$$\min J = \lambda_1 SOC + \lambda_2 MS + \lambda_3 EC + \lambda_4 CR + \lambda_5 DT - \lambda_6 TH, \tag{4}$$

where SOC is sum of costs, MS is makespan, EC is total energy consumption, CR is collision risk, DT is deadlock time, and TH is throughput.

The optimization is subject to standard MAPF and warehouse resource constraints. Vertex collisions are forbidden [23]:

$$p_i^t \neq p_j^t, \quad \forall i \neq j, \forall t. \tag{5}$$

Edge-swap collisions are forbidden [23]:

$$(p_i^t, p_i^{t+1}) \neq (p_j^{t+1}, p_j^t), \quad \forall i \neq j, \forall t. \tag{6}$$

Battery dynamics are

$$b_i^{t+1} = \min\{B_{\max}, b_i^t - E_i^t + \eta_i^t\}, \tag{7}$$

where η_i^t is nonzero only when robot i charges at a station. Operational safety requires

$$b_i^t \geq b_{\min}, \tag{8}$$

or the robot must be routed toward a charger. Charging station capacity is constrained by

$$\sum_i \mathbf{1}(p_i^t = c_k) \leq cap(c_k), \quad \forall c_k \in C. \tag{9}$$

Each task $\tau_m = (p_m^{pickup}, p_m^{delivery})$ is complete only if a robot visits pickup before delivery:

$$done(\tau_m) = 1 \Leftrightarrow \exists i, t_1 < t_2 : p_i^{t_1} = p_m^{pickup},\ p_i^{t_2} = p_m^{delivery}. \tag{10}$$

 doi:10.20944/preprints202605.1283.v1

Table 2. Main notation.

Symbol	Description
$G = (V, E)$	Warehouse grid graph
O	Obstacles such as shelves and walls
R	Robot set
T	Lifelong task stream
C	Charging station set
p_i^t	Position of robot i at time t
g_i^t	Current goal of robot i
b_i^t	Battery level of robot i
v_i^t	Motion direction of robot i
l_i^t	Payload state of robot i
SOC	Sum of costs
MS	Makespan
EC	Energy consumption
CR	Collision risk
DT	Deadlock time
TH	Throughput

4. Energy and Charging Model

At each step, robot i consumes

$$E_i^t = \alpha_1 m_i^t + \alpha_2 q_i^t + \alpha_3 w_i^t + \alpha_4 l_i^t + \alpha_5 c_i^t, \tag{11}$$

where m_i^t indicates movement, q_i^t turning, w_i^t waiting, l_i^t loaded movement, and c_i^t local congestion. Unless stated otherwise, $\alpha_1 = 1.0$, $\alpha_2 = 0.3$, $\alpha_3 = 0.2$, $\alpha_4 = 0.5$, and $\alpha_5 = 0.4$. Battery capacity is 100, the charging rate is 10 per step, and robots below a threshold of 20 are routed to the nearest charger. Robots already at a charger wait until sufficiently recharged.

5. Risk-Aware Graph and Learning-Compatible Policy

At every time step, robots form a dynamic interaction graph

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t). \tag{12}$$

Each node corresponds to one robot. Node features include normalized position, normalized goal, battery level, payload flag, current direction, local obstacle density, distance to goal, and distance to nearest charger. Edge features include relative distance, relative angle, predicted conflict probability, same-corridor flag, and opposite-direction flag. Edges are created when robots are within communication radius, share a corridor, or have predicted short-horizon conflict.

The learning-compatible policy module uses self-attention [37] to encode long-range dependencies. A policy head outputs per-agent action logits and a value head estimates the global value for PPO. In the current paper, this module is treated as an executable preliminary learning baseline; the main evaluation claims are based on the repair-oriented ECR-HR variant and the search baselines.

$$h_i^{(0)} = \phi_x(x_i), \tag{13}$$

where $x_i \in \mathbb{R}^{10}$ is the node feature vector. Edge features are encoded as

$$e_{ij} = \phi_e([d_{ij}, \theta_{ij}, risk_{ij}, corridor_{ij}, opposite_{ij}]). \tag{14}$$

 doi:10.20944/preprints202605.1283.v1

For attention head m in layer ℓ, edge-biased attention is

$$\alpha_{ij}^{(\ell,m)} = \mathrm{softmax}_j\left(\frac{Q_i^{(\ell,m)} K_j^{(\ell,m)T} + \psi_m(e_{ij})}{\sqrt{d_h}}\right), \tag{15}$$

and node states are updated by

$$h_i^{(\ell+1,m)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(\ell,m)} V_j^{(\ell,m)}. \tag{16}$$

The multi-head outputs are concatenated and passed through feed-forward layers with residual connections. The policy and value heads are

$$\pi_\theta(a_i^t|\mathcal{G}_t) = \mathrm{softmax}(W_\pi h_i^{(L)}), \tag{17}$$

$$V_\phi(\mathcal{G}_t) = W_v\ \mathrm{pool}(\{h_i^{(L)}\}_{i=1}^N). \tag{18}$$

Invalid moves into obstacles or outside the map are masked before execution, and the repair layer further checks multi-agent conflicts.

PPO uses the clipped objective from Schulman et al. [34]:

$$L^{CLIP}(\theta) = \mathbb{E}_t\left[\min(\rho_t(\theta)\hat{A}_t, \mathrm{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right], \tag{19}$$

where $\rho_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$.

The reward is

$$r_t = 20\Delta N_{\text{task}} - 0.05E_t - 50C_t - 30D_t - 0.2W_t - R_t, \tag{20}$$

where ΔN_{task} is newly completed tasks, E_t energy, C_t collisions, D_t deadlock, W_t waiting, and R_t predicted risk.

6. Hybrid Search Repair

The policy action is checked before execution. The repair module detects vertex collisions, edge-swap collisions, battery violations, and charging conflicts. Conflicting robots are ranked by battery, payload, urgency, and distance to goal. Higher-priority agents keep their moves, while lower-priority agents are replanned locally using A* or WHCA*. If repair fails, the robot waits.

6.1. Full Framework and Executable Variant

The full ECR-HR framework is learning-compatible: candidate actions may come from a graph transformer policy, a heuristic planner, or a search baseline. The safety layer is shared across these sources and is responsible for checking energy, collision, and charging feasibility before execution. Because the current neural policy is not trained long enough to establish large-scale dominance over mature search solvers, the main reproducible simulation study evaluates ECR-HR as a repair-oriented executable variant. This variant uses the same energy model, risk representation, conflict detection, and local repair logic without relying on a fully mature learned policy. The GNN-PPO results are therefore reported separately as a preliminary learning pipeline, not as the central evidence for the paper's claims.

 doi:10.20944/preprints202605.1283.v1

Algorithm 1 ECR-HR Online Execution

Initialize robots, task queue, chargers, and policy π_θ
while simulation is not terminated **do**
    Assign idle robots to available tasks
    Construct interaction graph $\mathcal{G}_t$
    Sample or select candidate actions from π_θ
    Detect vertex, edge-swap, charging, and battery violations
    if conflict exists **then**
        Rank conflicting robots by energy and task urgency
        Repair low-priority robots using local A*/WHCA*
    end if
    Execute repaired actions
    Update energy, charging, payload, congestion, and task states
end while

Algorithm 2 Online MAPF-LNS2-Style Destroy-and-Repair

Generate initial actions using WHCA*
Detect conflicting robots and nearby robots
for iteration $= 1, \ldots, K$ **do**
    Select a conflict neighborhood
    Keep non-neighborhood actions fixed in the reservation table
    for each robot in the neighborhood **do**
        Replan a local path with A* under reservations
        Reserve the repaired next action
    end for
    if the repaired joint action improves collision and distance score **then**
        Accept the repaired joint action
    end if
end for
Return the best next action

Algorithm 3 GNN-PPO with Search Warm-Start

Initialize graph transformer policy π_θ and value function V_ϕ
for imitation step $= 1, \ldots, M$ **do**
    Generate an expert next action using WHCA*
    Update π_θ by cross-entropy imitation loss
end for
for episode $= 1, \ldots, E$ **do**
    Roll out π_θ in the warehouse simulator
    Apply safety fallback when learned action loses progress
    Store observations, actions, log probabilities, rewards, and done flags
    Update π_θ, V_ϕ with clipped PPO objective
end for

7. Implementation

The codebase is implemented in Python and supports reproducible execution through fixed seeds. It includes map generation, task generation, energy simulation, charging, collision checking, metrics, visualization, and multiple planners. Bounded CBS/ECBS solve a limited active-agent snapshot at each step and fall back to prioritized planning when the high-level budget is exhausted. MAPF-LNS2-style repair performs conflict-neighborhood destroy-and-repair. LaCAM-style planning lazily constructs a joint one-step successor using A*-guided candidate moves. GNN-PPO uses WHCA* imitation warm-start followed by PPO updates and a safety fallback.

 doi:10.20944/preprints202605.1283.v1

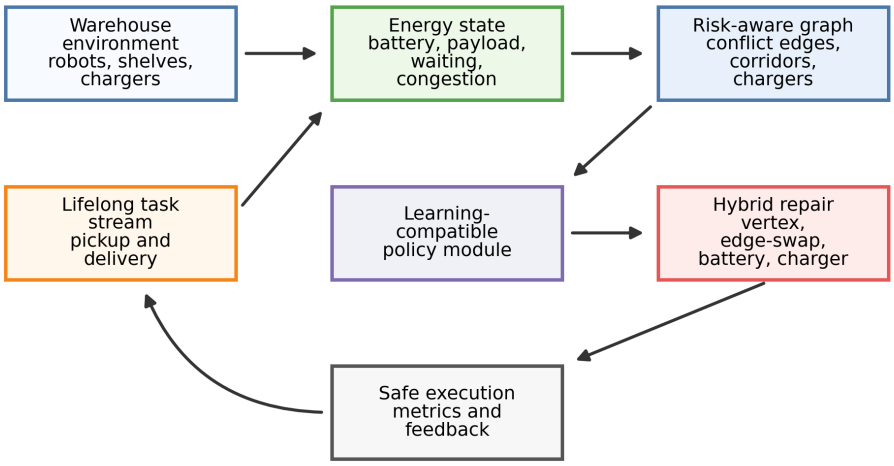

Figure 1. Overall ECR-HR framework.

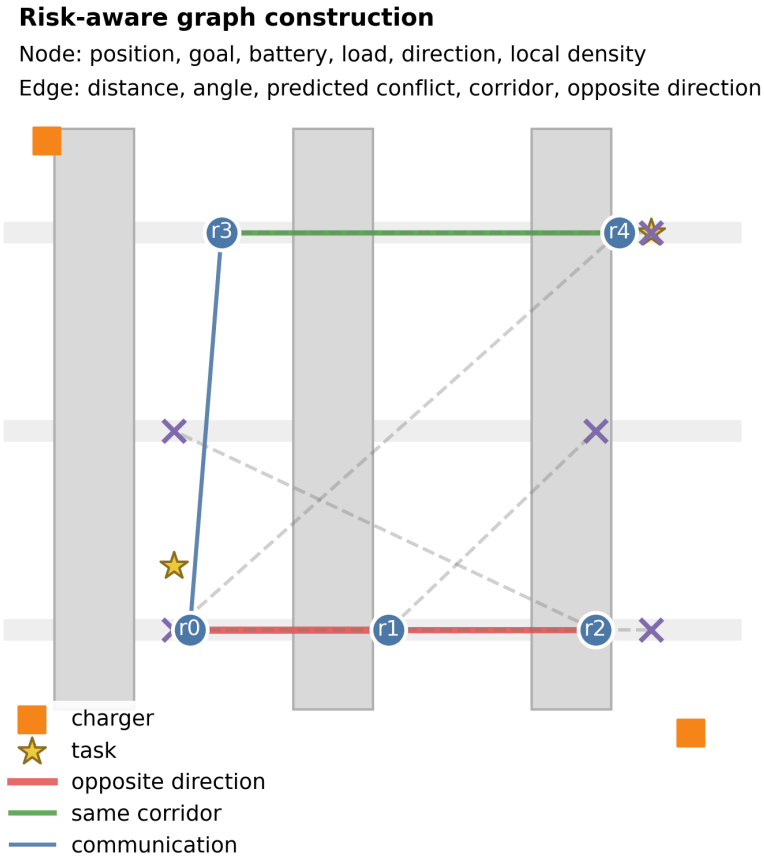

Figure 2. Risk-aware graph construction.

 doi:10.20944/preprints202605.1283.v1

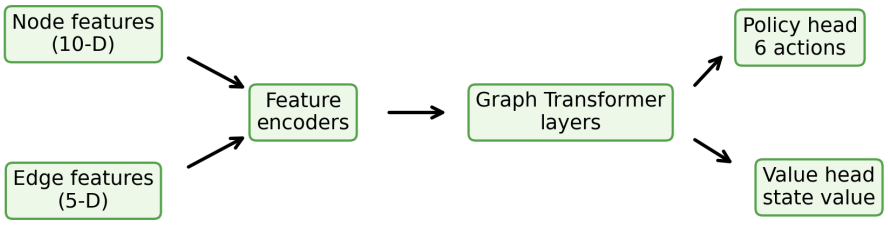

Figure 3. Learning-compatible graph policy architecture.

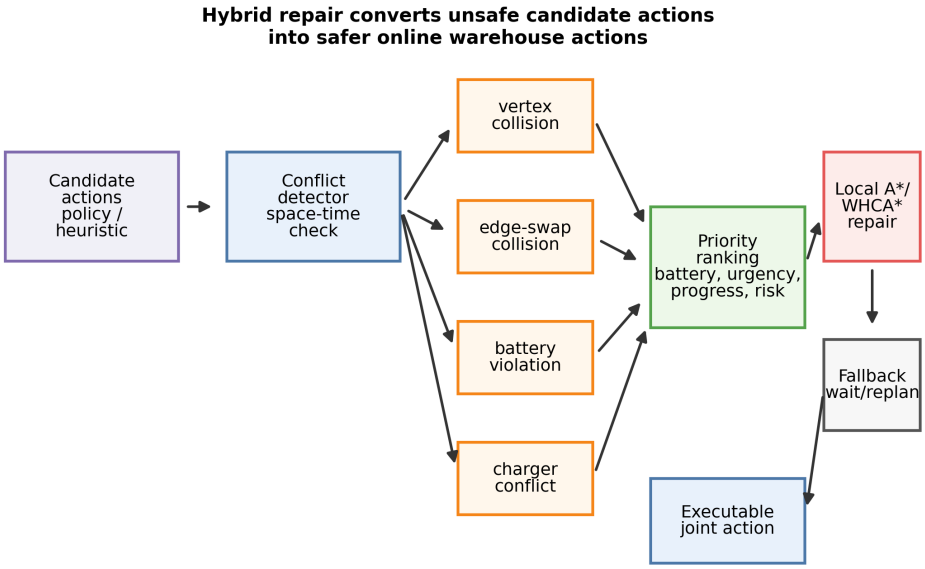

Figure 4. Hybrid repair mechanism for converting candidate actions into safer executable actions.

8. Simulation Setup

We evaluate warehouse maps of size 20×20 with 10 robots and 30 tasks, and 40×40 with 20 robots and 80 tasks. Each main setting is evaluated over five random seeds. The maximum horizon is 180 steps for 20×20 and 420 steps for 40×40. Extended CBS/ECBS simulation runs use 20×20 maps with 8 robots and 20 tasks because bounded CBS is substantially slower. The GNN-PPO virtual training run uses 6 robots, 16 tasks, 180 steps, 1800 imitation warm-start steps, and 12 PPO episodes.

Metrics include raw success rate, energy-feasible success rate, candidate conflict rate, executed collision rate, deadlock rate, total energy, energy per completed task, throughput, waiting time, charging waiting time, battery violation events, charger conflict events, and computation time. Raw success counts all completed tasks. Energy-feasible success counts only tasks completed by robots that have not experienced battery depletion before completion; this prevents policies from being rewarded for completing tasks after violating battery feasibility. In the tables, Cand. denotes candidate conflict rate, Exec. denotes executed collision rate, Feas. denotes feasible success, Batt. denotes battery-violation events, Chg. denotes charger-conflict events, E/task or Energy/task is reported in simulator energy units, and Time is reported in seconds.

 doi:10.20944/preprints202605.1283.v1

Table 3. Simulation parameters.

Parameter	Value
Map sizes	20×20, 40×40
Main robot counts	10, 20
Main task counts	30, 80
Extended-search setting	20×20, 8 robots, 20 tasks
GNN-PPO setting	20×20, 6 robots, 16 tasks
Random seeds	42–46 for main baselines; 42–43 for extended baselines
Movement energy α_1	1.0
Turning energy α_2	0.3
Waiting energy α_3	0.2
Payload energy α_4	0.5
Congestion energy α_5	0.4
Battery capacity	100 energy units
Charging rate	10 energy units per step
Low-battery threshold	20 energy units
WHCA* window	12 steps
GNN-PPO hidden dimension	96 dimensions
PPO episodes	12 episodes
Imitation warm-start steps	1800 steps
Main hardware	Windows 10, 16 CPU threads, CPU-only PyTorch
Software versions	Python 3.10.7, NumPy 2.2.6, pandas 2.3.3, PyTorch 2.11.0+cpu

8.1. Environment and Solver Details

Warehouse maps are generated procedurally. The map boundary is blocked, and shelf blocks are inserted every six columns with periodic cross aisles every nine rows. Four charging stations are placed at the traversable cells nearest the four map corners. Pickup locations are sampled from the left-side warehouse zone and delivery locations from the right-side zone; if these zones are empty, all free cells are used. Tasks are generated with fixed random seeds by sampling pickup and delivery cells independently while preventing identical pickup–delivery pairs.

Idle robots receive tasks in first-in-first-out order from the pending task list. Robots first travel to pickup, then to delivery, and are assigned another task after completion. If a robot is located at a charger and its battery is below 80% capacity, it remains at the charger. If its battery is below the low-battery threshold, its goal is changed to the nearest charger. Charging adds 10 units per step up to the battery capacity of 100. The local congestion cost is computed from the number of robots within Manhattan distance two of the robot's current cell, clipped to $[0,1]$. All A* calls use Manhattan distance as the heuristic. WHCA* uses a reservation window of 12 unless otherwise stated. MAPF-LNS2-style repair uses a bounded number of destroy-and-repair iterations, and CBS/ECBS are bounded online adaptations with a limited active-agent snapshot. All reported wall-clock times are measured on the same CPU-only environment. The reproducibility package includes all scripts; an anonymized repository or archival DOI should be added at submission time.

8.2. Evaluation Metrics

Raw success rate is the fraction of tasks completed before the horizon. Energy-feasible success rate is the fraction of tasks completed without prior battery violation by the completing robot. Candidate conflict rate is the number of vertex and edge-swap conflicts in the proposed joint action before repair or fallback, normalized by robot-time steps. Executed collision rate is the same measurement after the simulator applies conflict blocking and action execution. Deadlock rate measures the fraction of steps in which no additional tasks are completed over a sliding window. Energy per completed task normalizes total energy by task completion and is used to compare energy efficiency across methods with different throughputs. Throughput is completed tasks per time step. Battery violation events count cases where a robot depletes its battery away from a charger, and charger conflict events count

 doi:10.20944/preprints202605.1283.v1

attempted simultaneous occupation of a charger. Computation time is wall-clock runtime for the simulated episode. The abbreviated table headings Cand., Exec., Feas., Batt., Chg., and E/task are used only after these definitions.

8.3. Ablation Protocol

We run an executable ablation study on the 20×20, 10-robot, 30-task setting over five seeds. Since the objective of this study is to understand the interaction between energy modeling and online repair, the ablation focuses on the repair-and-energy system: full ECR-HR, removing energy-aware routing, removing congestion cost, removing payload cost, removing turning/waiting cost, removing hybrid repair, and replacing the repair layer with prioritized A*. These variants isolate the contribution of energy modeling and local repair in the current reproducible implementation.

9. Main Results

Table 4 summarizes the main baseline results over five random seeds. Boldface values mark the best value in each metric column for the reported setting, with higher values preferred for success-rate metrics and lower values preferred for conflict rates, energy/task, and event counts. Independent A* fails under multi-agent interaction, reaching only 0.015 raw success on the 40×40 setting. Prioritized A* and WHCA* improve safety and throughput. MAPF-LNS2-style repair obtains the highest raw success in both main settings and the strongest energy-feasible success on the 20×20 setting. On the 40×40 setting, MAPF-LNS2 improves raw success from 0.345 for WHCA* to 0.468 and reduces energy per completed task from 369.34 to 276.83, but its energy-feasible success is comparable to WHCA*. ECR-HR reduces candidate conflict and charger conflicts relative to WHCA*, but its 40×40 raw and energy-feasible success remain below MAPF-LNS2. The correct empirical interpretation is therefore that ECR-HR is a repair-oriented framework with useful safety behavior, while MAPF-LNS2 remains the strongest throughput baseline in the tested configurations.

Table 4. Main results averaged over five seeds.

20×20, 10 robots	Raw rate	Feas. rate	95% CI rate	Cand. rate	Exec. rate	Batt. events	Chg. events	E/task units
A*	0.047	0.040	±0.024	0.617	0.498	53.8	28.8	718.87
Prioritized A*	0.727	0.620	±0.159	0.042	0.054	230.6	74.4	91.84
WHCA*	0.733	0.620	±0.158	0.044	0.053	216.2	78.6	90.10
MAPF-LNS2	**0.853**	**0.687**	±0.094	0.031	0.013	53.2	52.6	**79.24**
LaCAM*	0.740	**0.687**	±0.102	**0.001**	**0.000**	**7.8**	1.0	103.21
ECR-HR	0.767	0.633	±0.134	0.027	0.018	182.0	**0.0**	87.19

40×40, 20 robots	Raw rate	Feas. rate	95% CI rate	Cand. rate	Exec. rate	Batt. events	Chg. events	E/task units
A*	0.015	0.008	±0.006	0.668	1.202	3549.0	198.2	2662.01
Prioritized A*	0.385	0.065	±0.016	0.068	0.066	2356.6	356.4	283.27
WHCA*	0.345	**0.075**	±0.008	0.059	0.070	3174.4	462.6	369.34
MAPF-LNS2	**0.468**	0.073	±0.018	**0.045**	**0.031**	**1672.0**	308.2	**276.83**
LaCAM*	0.240	0.055	±0.026	0.057	0.078	3585.8	434.8	553.68
ECR-HR	0.333	0.073	±0.009	0.049	0.061	3143.2	**0.0**	378.40

 doi:10.20944/preprints202605.1283.v1

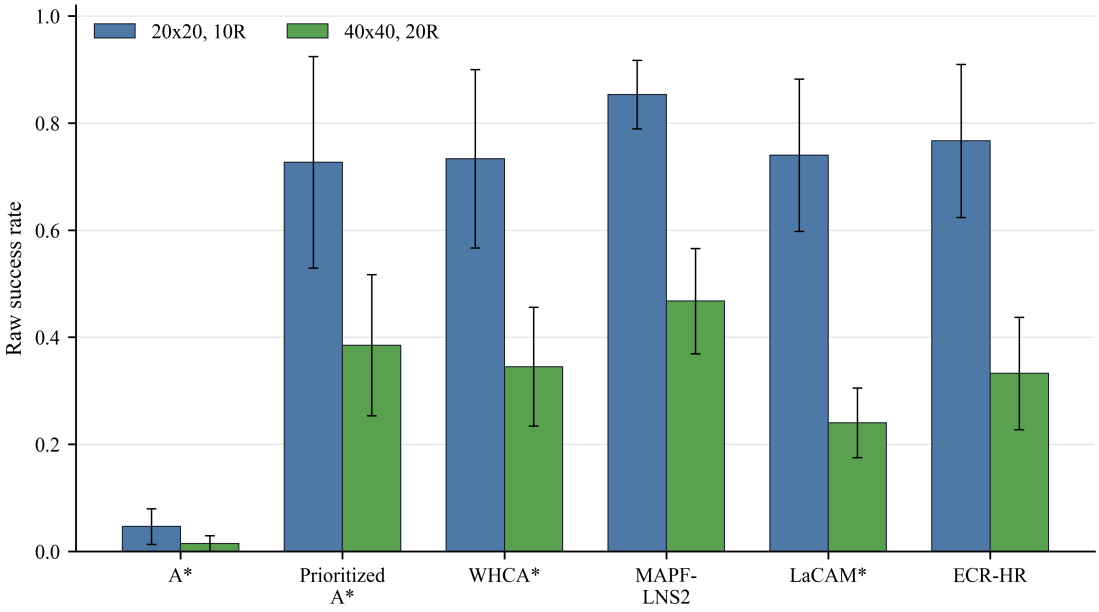

Figure 5. Raw success-rate comparison.

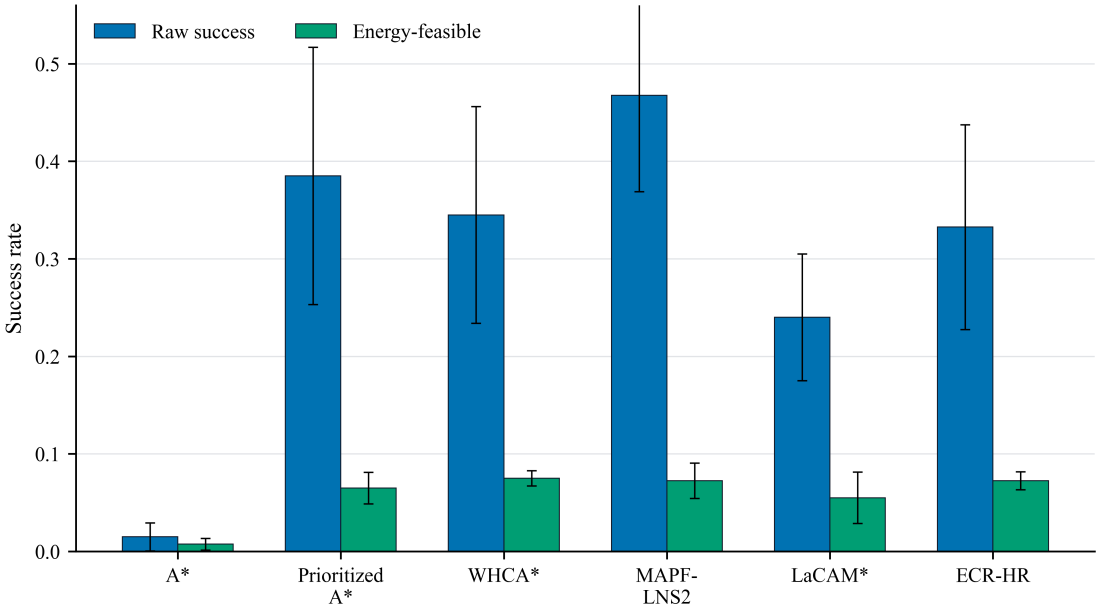

Figure 6. Rate-level comparison on the 40×40 setting.

10. Extended Search Baselines

Table 5 reports bounded CBS/ECBS, MAPF-LNS2, LaCAM*, and WHCA* on a smaller setting. Boldface values mark the best value in each metric column, with higher values preferred for success-rate metrics and lower values preferred for conflict rates, energy/task, battery events, and computation time. CBS and ECBS obtain high raw success but require much more computation than rolling-horizon methods. ECBS is faster than CBS in this bounded setting. MAPF-LNS2 obtains the highest raw success, while LaCAM* and WHCA* obtain strong energy-feasible success with low runtime. These results further show why both raw task completion and energy-feasible completion should be reported.

 doi:10.20944/preprints202605.1283.v1

Table 5. Extended search baselines on 20×20 maps with 8 robots and 20 tasks.

Method	Raw succ. rate	Feasible succ. rate	Cand. conflict rate	Exec. collision rate	Energy/task (units)	Battery events	Time (s)
CBS	0.875	0.700	0.041	0.025	79.27	33.5	48.21
ECBS	0.875	0.700	0.041	0.025	79.27	33.5	36.16
WHCA*	0.875	0.775	0.050	0.009	76.57	**2.0**	0.30
MAPF-LNS2	**0.900**	**0.800**	0.020	0.009	**74.99**	12.5	3.34
LaCAM*	0.800	0.750	**0.001**	**0.000**	93.43	2.5	**0.29**

11. GNN-PPO Training

The GNN-PPO policy was trained with imitation warm-start and PPO fine-tuning. During training, episode success rate reached 0.875 in the best episode, but the held-out greedy evaluation achieved 0.188 success with high collision rate. Figure 7 shows the training curve. This result indicates that the learning pipeline is executable and can improve during training, while also confirming that reliable learning-based MAPF requires safety-aware policy optimization. In future extensions, collision-aware action masking, larger replay diversity, curriculum learning, and direct integration of the repair layer during policy optimization are expected to improve robustness.

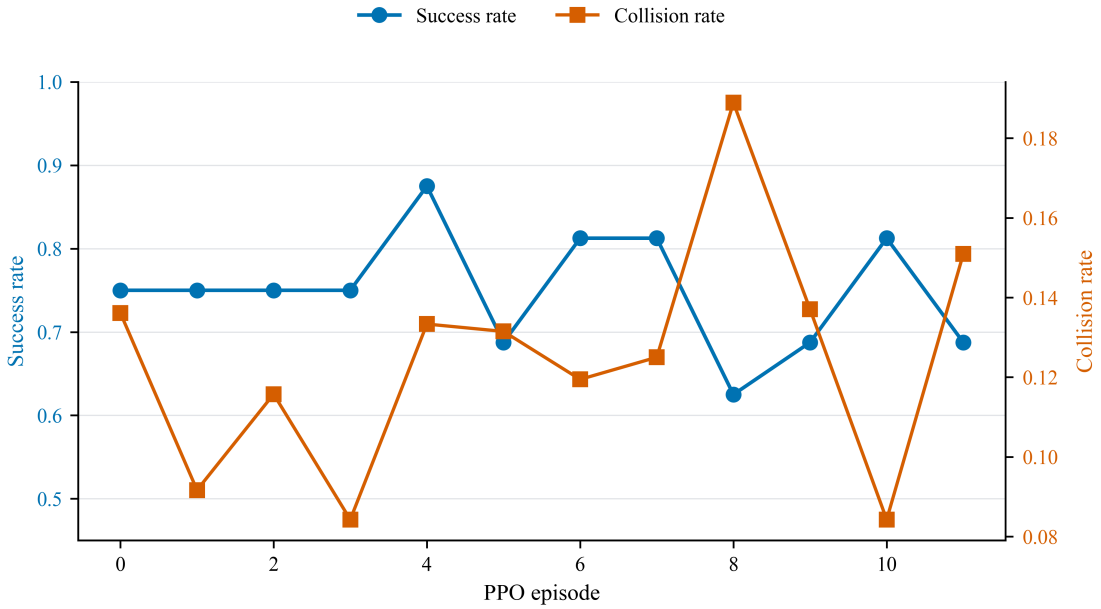

Figure 7. GNN-PPO training curve over 12 PPO episodes after WHCA* imitation warm-start.

12. Ablation Study

Table 6 reports the ablation results. Boldface values mark the best value in each metric column, with higher values preferred for success-rate metrics and lower values preferred for conflict rates, energy/task, battery events, and charger events. The raw success column alone would suggest that removing energy-aware routing is best, with 0.980 raw success compared with 0.767 for full ECR-HR. The energy-feasible success column changes the interpretation: without energy-aware routing, feasible success drops to 0.433 because the variant completes many tasks after battery feasibility has already been violated. Full ECR-HR reaches 0.633 energy-feasible success, reduces charger conflicts to zero, and keeps candidate conflict lower than WHCA* fallback. This resolves the main modeling ambiguity: energy-aware routing should be evaluated by feasibility-aware completion and safety events, not raw completion alone.

 doi:10.20944/preprints202605.1283.v1

Table 6. Ablation results on 20×20 maps with 10 robots and 30 tasks over five seeds.

Variant	Raw succ. rate	Feasible succ. rate	Cand. conflict rate	Exec. collision rate	Energy/task (units)	Battery events	Charger events
Full ECR-HR	0.767	0.633	0.027	0.018	87.19	182.0	**0.0**
No energy route	**0.980**	0.433	**0.000**	**0.000**	62.94	819.0	**0.0**
No congestion cost	0.793	0.653	0.023	0.016	81.91	110.4	**0.0**
No payload cost	0.913	**0.827**	0.007	0.004	**58.77**	**15.2**	**0.0**
No turn/wait cost	0.767	0.613	0.033	0.021	79.03	169.2	**0.0**
No hybrid repair	0.733	0.620	0.044	0.053	90.10	216.2	78.6
Prioritized fallback	0.727	0.620	0.042	0.054	91.84	230.6	74.4

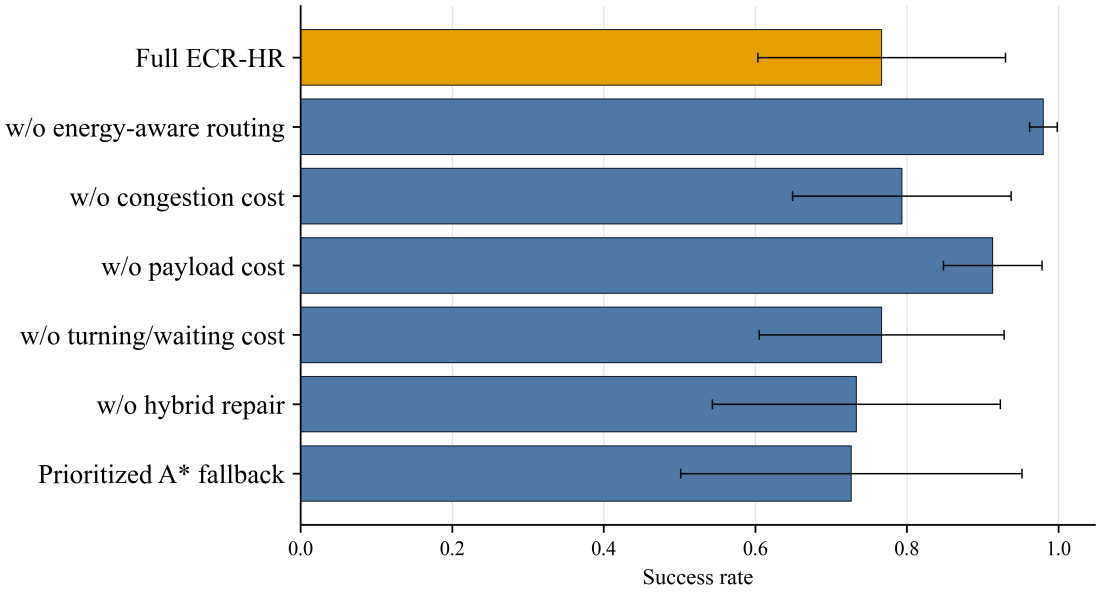

Figure 8. Ablation results for success rate.

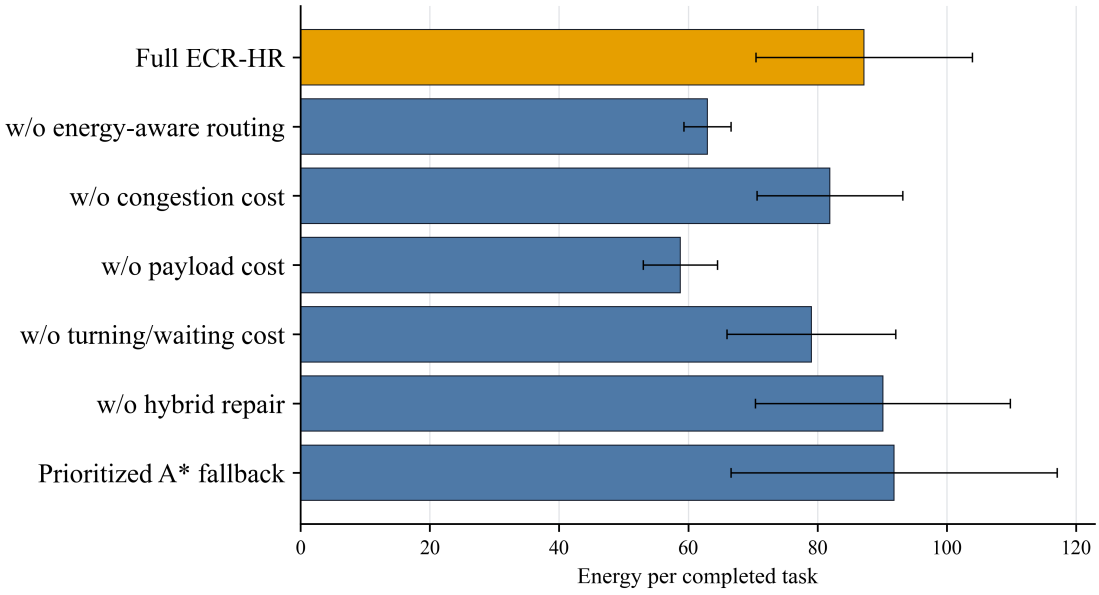

Figure 9. Ablation results for energy per completed task.

 doi:10.20944/preprints202605.1283.v1

13. Parameter Sensitivity

Table 7 summarizes the sensitivity analysis. The low-battery threshold strongly affects performance: a threshold of 20 is conservative in the current simulator, while 10 and 30 produce higher success. Increasing the WHCA* window from 12 to 24 increases computation time without improving success in this setting. Payload cost affects energy per task as expected, while congestion cost changes the collision-throughput balance.

Table 7. Parameter sensitivity summary on 20×20 maps with 10 robots and 30 tasks.

Parameter	Values tested (units where applicable)	Best setting	Main observation
Battery threshold	10, 20, 30, 40	10 for success and energy	Too-conservative charging can reduce throughput.
WHCA* window	8, 12, 16, 24	8 for success and energy	Larger windows increased time without benefit.
Payload cost	0.2, 0.5, 0.8	0.2 for success and energy	Higher load penalty increases energy per task.
Congestion cost	0.0, 0.4, 0.8	0.8 for success; 0.0 for energy	Congestion weighting changes the safety/throughput balance.

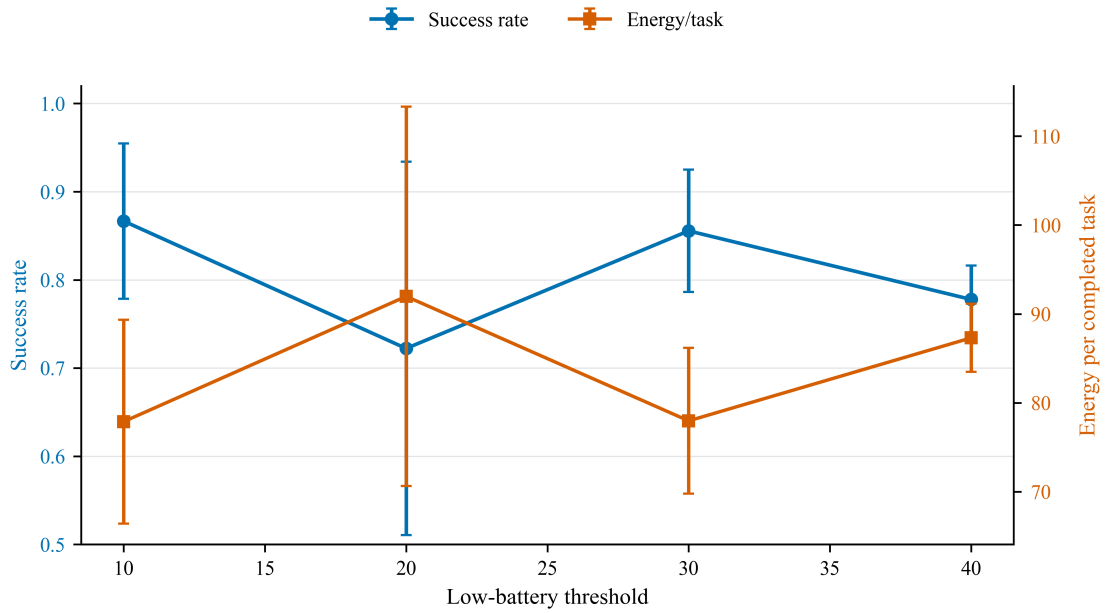

Figure 10. Sensitivity curve for the battery threshold.

 doi:10.20944/preprints202605.1283.v1

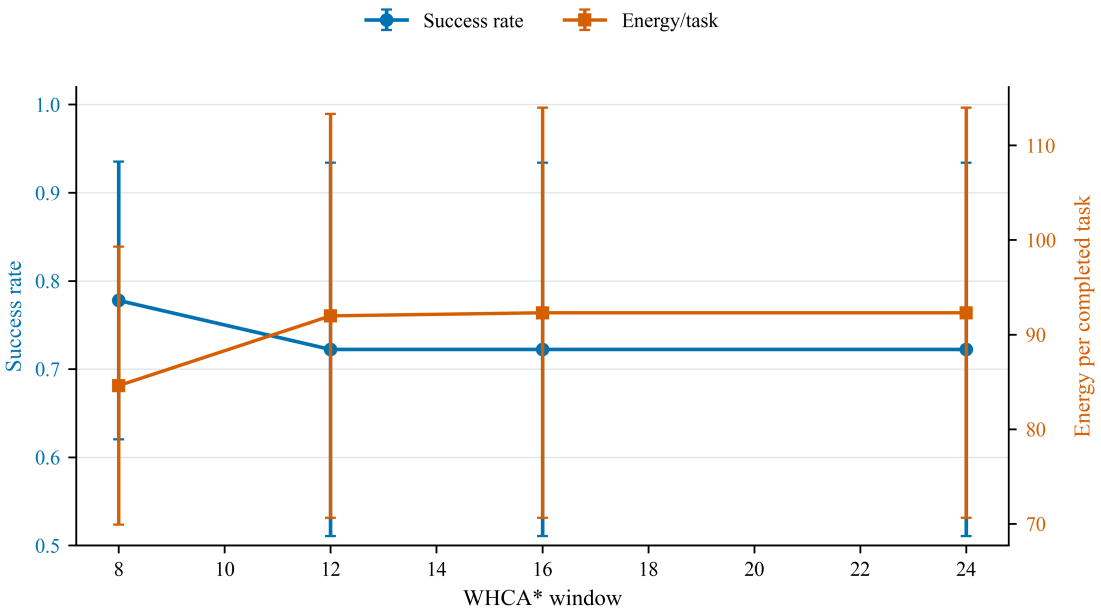

Figure 11. Sensitivity curve for the WHCA* window.

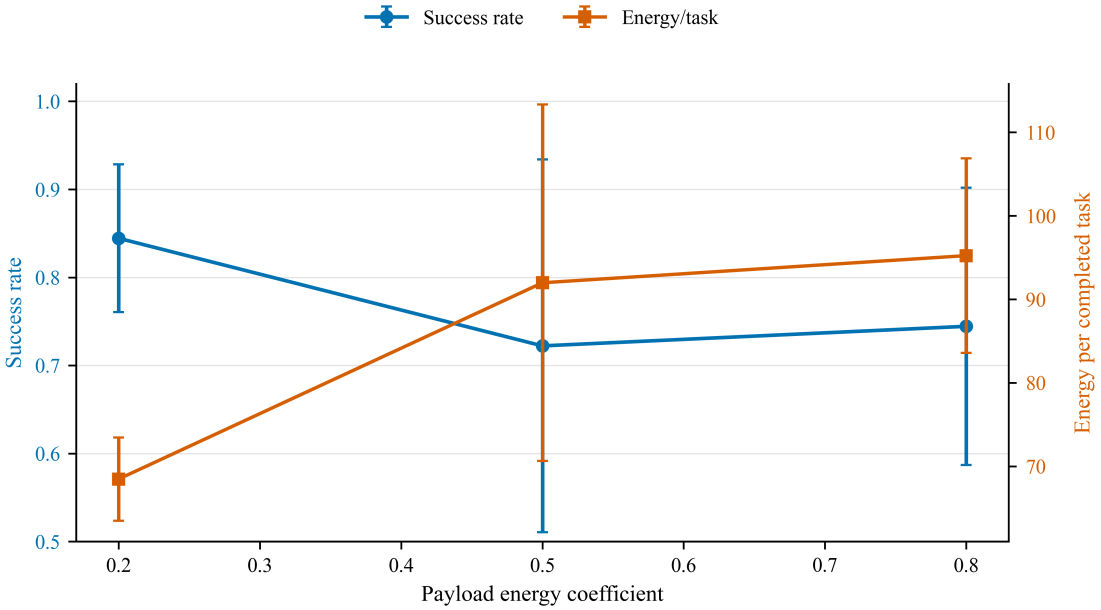

Figure 12. Sensitivity curve for the payload cost.

 doi:10.20944/preprints202605.1283.v1

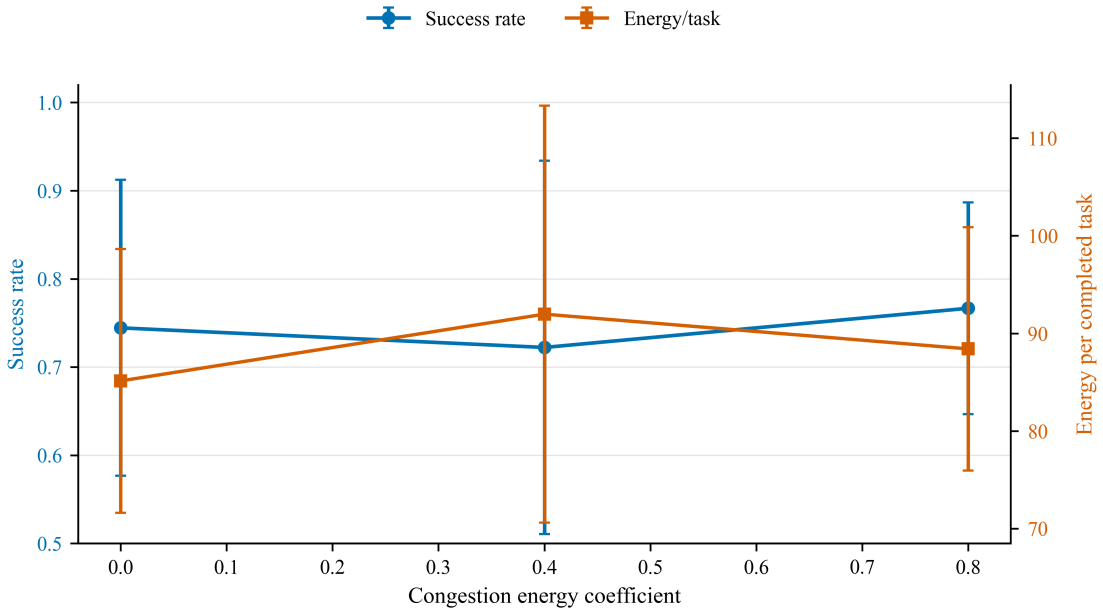

Figure 13. Sensitivity curve for the congestion cost.

14. Scalability and Robot Density

Table 8 reports an additional robot-density simulation study over three seeds. Boldface values mark the best value within each robot-density block, with higher values preferred for success rate and lower values preferred for collision rate, energy/task, and computation time. Success drops sharply at 30–50 robots because the current warehouse layout and time horizon become highly congested. MAPF-LNS2 is strongest at 20 and 30 robots, while all methods struggle at 50 robots. The result highlights that supporting 50–100 robots requires either wider maps, stronger task allocation, better charging coordination, or deeper learned policies.

Table 8. Scalability results over three seeds.

Method and setting	Success rate	Collision rate	Energy/task (units)	Time (s)
WHCA*, 20×20, 10R	0.667 ± 0.233	0.051	96.48	0.79
ECR-HR, 20×20, 10R	0.722 ± 0.212	0.039	91.97	0.73
MAPF-LNS2, 20×20, 10R	**0.856** ± 0.069	0.033	**79.06**	2.32
WHCA*, 40×40, 20R	0.413 ± 0.109	0.052	293.48	6.89
ECR-HR, 40×40, 20R	0.392 ± 0.095	0.041	306.49	6.53
MAPF-LNS2, 40×40, 20R	**0.454** ± 0.156	0.047	**294.06**	111.56
WHCA*, 50×50, 30R	0.103 ± 0.010	**0.066**	1377.44	17.61
ECR-HR, 50×50, 30R	0.086 ± 0.013	0.075	1524.82	19.87
MAPF-LNS2, 50×50, 30R	**0.181** ± 0.027	0.068	**861.19**	88.73
WHCA*, 60×60, 50R	0.040 ± 0.010	0.090	4442.70	35.38
ECR-HR, 60×60, 50R	**0.043** ± 0.015	**0.082**	**3564.36**	37.84
MAPF-LNS2, 60×60, 50R	0.040 ± 0.015	0.091	4355.26	122.98

 doi:10.20944/preprints202605.1283.v1

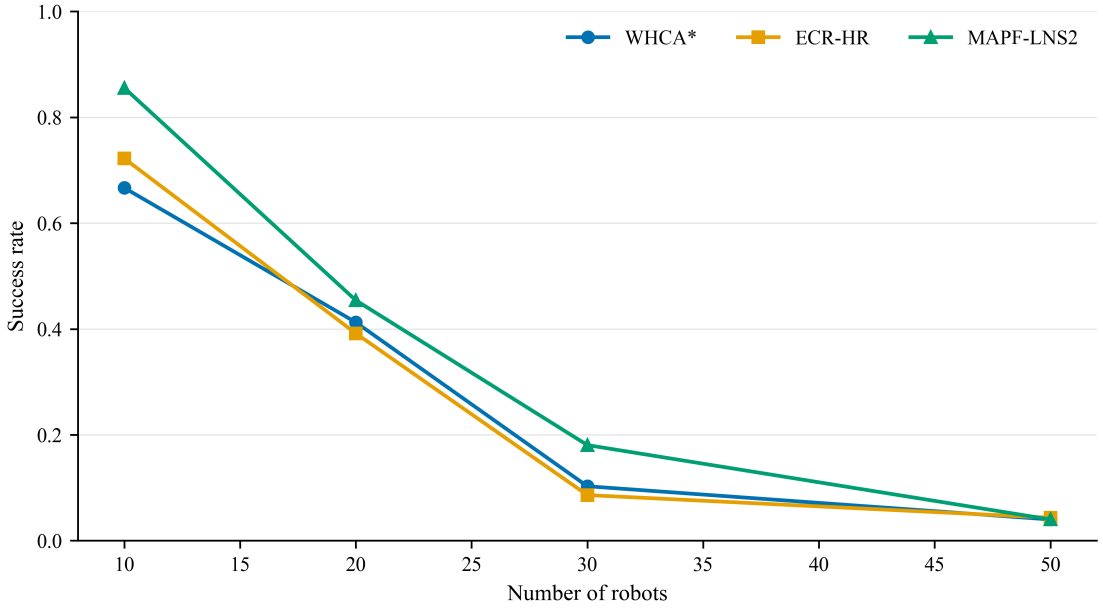

Figure 14. Robot density success.

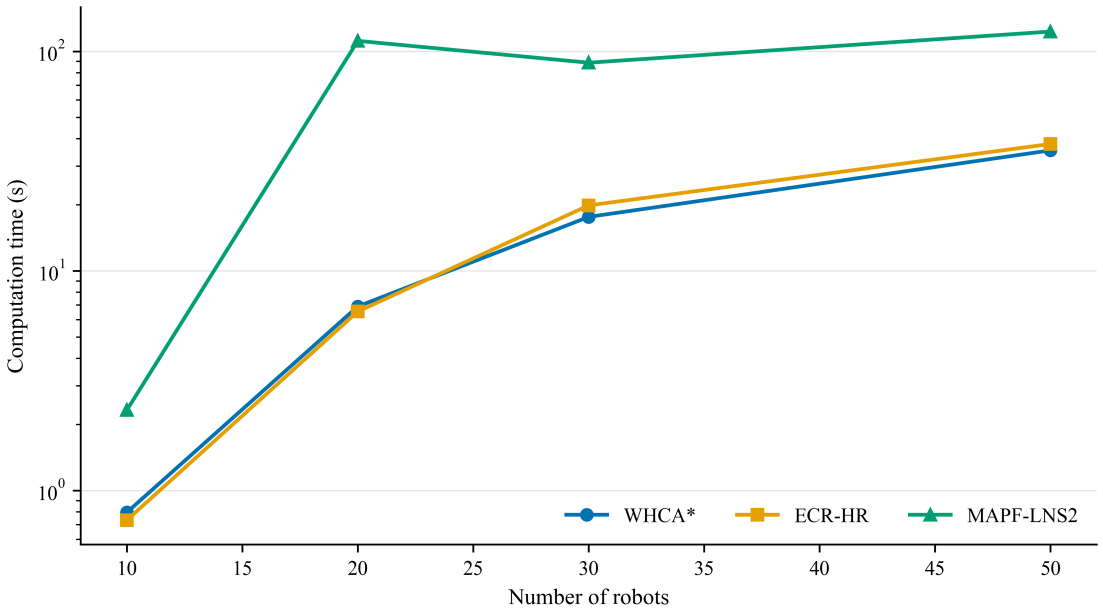

Figure 15. Computation-time scaling.

15. Descriptive Comparison Against WHCA*

Table 9 reports descriptive relative changes against WHCA* on the five-seed main results. We avoid hypothesis-test claims because five random seeds are useful for trend analysis but still insufficient for strong statistical significance claims. MAPF-LNS2 shows the clearest improvement over WHCA* in raw success and energy per completed task, especially on the 40×40 setting. ECR-HR improves candidate conflict and executed collision rates relative to WHCA* on both map sizes, while its 40×40 raw and energy-feasible success are slightly lower.

 doi:10.20944/preprints202605.1283.v1

Table 9. Descriptive comparison against WHCA* under limited random seeds. Negative relative change is better for energy/task and collision rate.

Method	Metric	20×20 (%)	40×40 (%)
MAPF-LNS2	Raw success	+16.4%	+35.5%
MAPF-LNS2	Energy-feasible success	+10.8%	-3.3%
MAPF-LNS2	Energy/task	-12.1%	-25.0%
MAPF-LNS2	Candidate conflict	-29.8%	-23.9%
MAPF-LNS2	Executed collision	-74.8%	-55.0%
ECR-HR	Raw success	+4.5%	-3.6%
ECR-HR	Energy-feasible success	+2.2%	-3.3%
ECR-HR	Energy/task	-3.2%	+2.5%
ECR-HR	Candidate conflict	-38.4%	-16.9%
ECR-HR	Executed collision	-67.1%	-11.9%

16. Case Study

Figures 16–18 visualize a 40×40 warehouse scenario with 20 robots and 80 tasks. WHCA* frequently forms dense waiting regions around narrow aisles. MAPF-LNS2 reduces some of these local conflicts through neighborhood repair and achieves higher throughput. ECR-HR reduces candidate conflicts and charger conflicts relative to WHCA* in the main 40×40 result, but the case study also shows that conservative repair can reduce task completion when congestion is severe. This qualitative observation matches the quantitative trade-off observed in the tables.

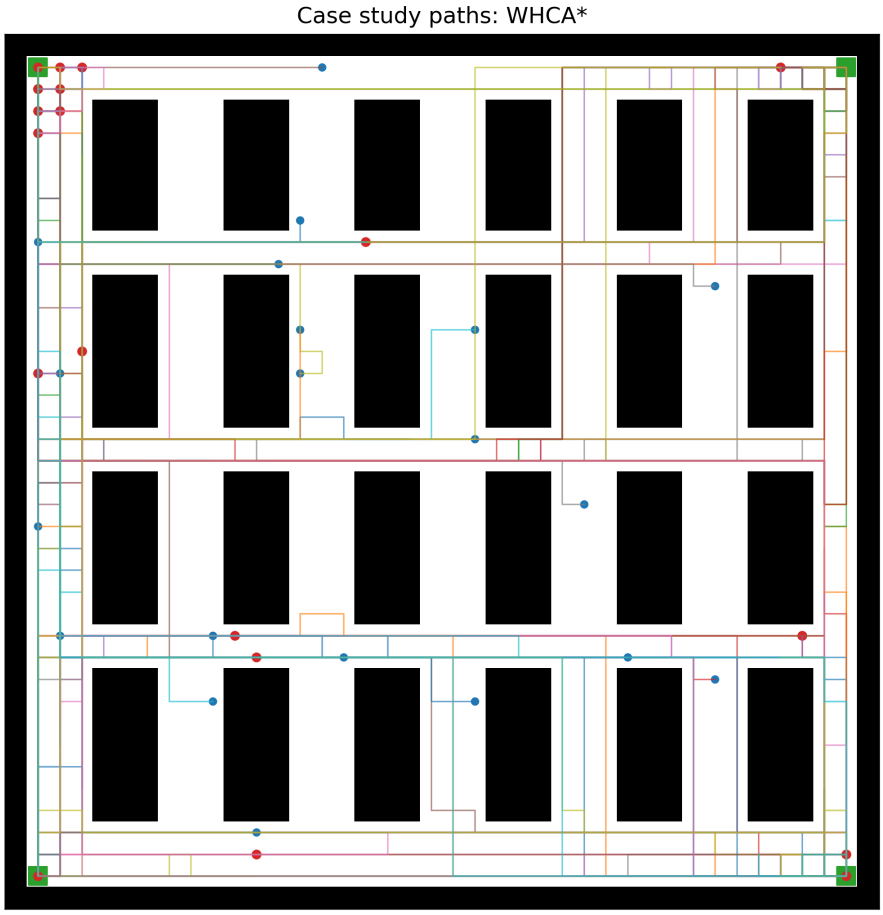

Figure 16. Case-study trajectory for WHCA*.

 doi:10.20944/preprints202605.1283.v1

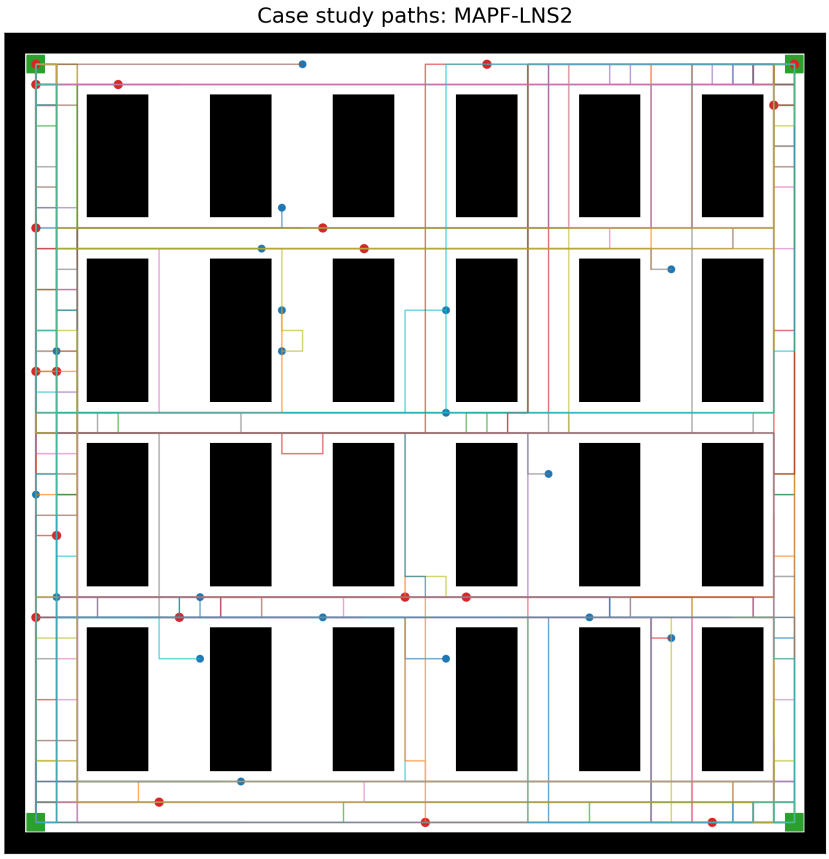

Figure 17. Case-study trajectory for MAPF-LNS2.

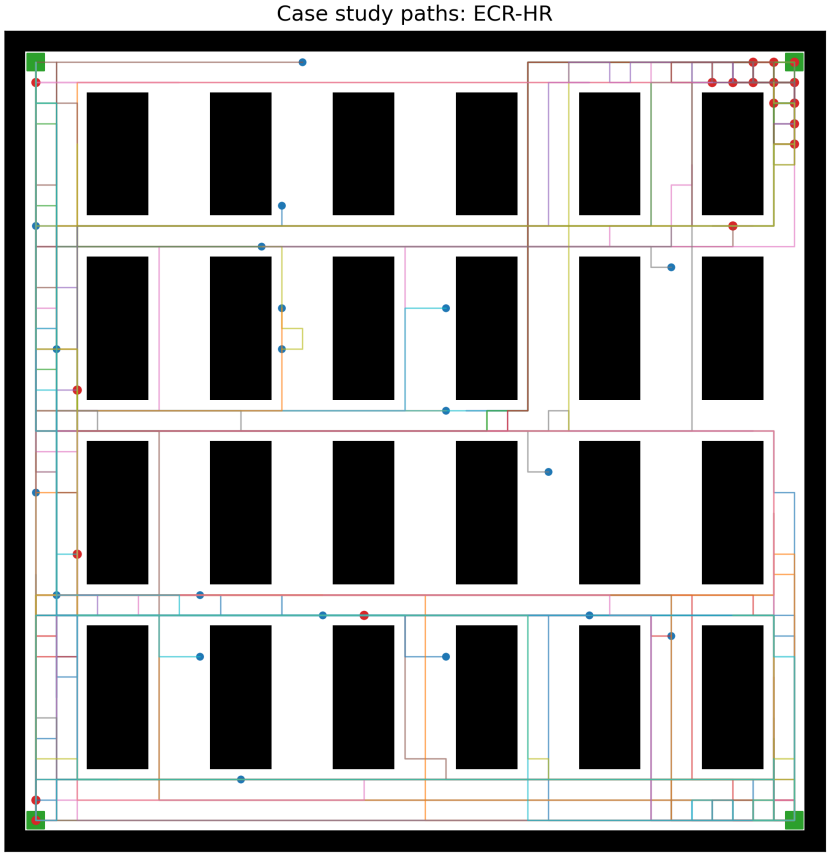

Figure 18. Case-study trajectory for ECR-HR.

 doi:10.20944/preprints202605.1283.v1

17. Discussion

The simulation results support several conclusions. First, independent shortest-path planning is unsuitable for lifelong warehouse MAPF because it ignores inter-agent constraints. Second, search-based coordination remains a very strong baseline. Third, local repair and large-neighborhood repair are effective for improving throughput and energy efficiency. Fourth, raw success can be misleading under battery constraints: a planner may complete tasks after battery violations, so energy-feasible success should be reported alongside raw success. Fifth, reducing conflict rates and improving throughput are not always aligned: LaCAM* and ECR-HR often reduce conflicts, but MAPF-LNS2 obtains better raw task completion. Sixth, learned graph policies require substantial training effort and safety mechanisms before they can reliably outperform search.

The main empirical lesson is that strong search baselines remain difficult to surpass. Therefore, the proposed learning-guided framework should be interpreted as a path toward safer and more adaptive online planning, not as a claim that short-horizon graph RL already dominates mature MAPF solvers.

17.1. Why Learning-Guided Hybrid Planning Still Matters

The strongest current numerical results come from MAPF-LNS2-style repair, not from the short-trained GNN-PPO model. This should be interpreted carefully. MAPF-LNS2 is an excellent non-learning baseline and is expected to perform strongly on structured grid maps. The motivation for the learning-compatible part of ECR-HR is not that a briefly trained policy automatically beats mature search, but that risk-aware learning can provide useful predictions under conditions where search alone becomes expensive or myopic: dynamic obstacles, changing charging queues, payload-dependent energy, unseen congestion patterns, and rolling task arrivals. The hybrid repair layer remains necessary because learned actions alone are unsafe. Thus the correct positioning is learning-guided repair rather than pure end-to-end replacement of search.

17.2. Why Hybrid Methods Matter

The results reinforce a practical design lesson. Search methods are reliable but can be expensive, especially when conflict resolution requires high-level branching. Learned policies are fast after training but can be unsafe. Hybrid planning allows each component to do what it is best suited for: learning can estimate risk, congestion, and priority, while search enforces local feasibility. This is particularly important in warehouses because safety failures are not merely optimization errors; they can block aisles, delay charging, and reduce sustained throughput.

17.3. Energy-Throughput Trade-Off

Energy efficiency and task throughput do not always improve together. A conservative planner may reduce conflicts but increase waiting, while an aggressive planner may complete more tasks but induce congestion or battery infeasibility. This is visible in the difference between raw success and energy-feasible success in the ablation table. Removing energy-aware routing completes many tasks in the short horizon, but feasible success drops sharply because many completions occur after battery violations. This supports the paper's central motivation: warehouse lifelong MAPF should be evaluated using a multi-objective metric set rather than only path length, makespan, or raw task completion.

17.4. Computation-Time Trade-Off

CBS and ECBS are informative but expensive in online lifelong settings. In the extended search evaluation, bounded CBS and ECBS reach high success but are substantially slower than WHCA* and LaCAM-style search. MAPF-LNS2 provides a compromise: it improves success and energy efficiency relative to WHCA* while remaining faster than bounded CBS/ECBS in the tested setting. This makes repair-based methods attractive for practical deployment.

 doi:10.20944/preprints202605.1283.v1

18. Complexity Analysis

Let N be the number of robots, $|V|$ the number of traversable cells, W the WHCA* window size, d the hidden dimension, H the number of attention heads, L the number of transformer layers, and K the number of repair iterations. A single A* call has worst-case complexity $O(|V| \log |V|)$ on the grid [8,9]. Prioritized planning over all robots requires approximately $O(N|V| \log |V|)$. WHCA* limits the time-expanded search to a window, giving practical complexity $O(NW|V_W| \log |V_W|)$ where $|V_W|$ is the reachable local space within the window [2].

The graph transformer processes a dense interaction graph in $O(LHN^2d)$ time and $O(N^2)$ attention memory. Sparse radius-based edges reduce this to $O(LH|\mathcal{E}_t|d)$. The hybrid repair layer selects a conflict group of size $q \leq N$ and performs local replanning, with approximate complexity $O(Kq|V_{local}| \log |V_{local}|)$. The overall online step complexity is therefore

$$O(LH|\mathcal{E}_t|d + Kq|V_{local}| \log |V_{local}|), \tag{21}$$

plus environment and collision checking. This explains the observed trade-off: MAPF-LNS2 improves success but costs more runtime, while LaCAM-style successor selection is fast but can be less robust in highly congested medium maps.

19. Engineering Deployment Considerations

In an industrial AMR stack, ECR-HR would sit between the warehouse management system (WMS/MES) task dispatcher and the low-level robot controller. The WMS provides task arrivals, pickup and delivery locations, and priority constraints. The planner returns the next safe grid action or short horizon route. Low-battery robots are redirected to chargers before task assignment, and charging-station capacity constraints prevent multiple robots from occupying the same charger. In a real deployment, continuous localization and obstacle detection would be handled by the robot controller, while the MAPF layer would operate on a discretized warehouse topology. Practical deployment requires a strict per-step latency budget; for example, in a system planning at 1 Hz, the online decision should complete well below one second for the active local region. The current results indicate that WHCA*, LaCAM-style planning, and ECR-HR are closer to this regime than bounded CBS/ECBS, while MAPF-LNS2 provides a stronger but slower repair option.

20. Limitations

This study uses a grid simulator rather than a real AMR platform. The CBS/ECBS, MAPF-LNS2, and LaCAM* components are bounded online adaptations rather than full official offline solvers. The GNN-PPO policy is evaluated as a learning baseline and proof-of-concept component rather than as the sole source of performance gains. The largest completed scalability setting contains 50 robots; larger 80–100 robot simulation runs are supported by the environment but require additional optimization of the strongest search baselines. These limitations define the boundary of the reported evidence.

21. Reproducibility

All results are generated by the Python codebase. Main baselines can be reproduced with:

```
python main.py
```

For a longer 20-seed run suitable for stronger statistical testing:

```
python main.py --seeds 20
```

Extended search baselines can be reproduced with:

```
python -m experiments.run_extended_baselines
```

GNN-PPO training can be reproduced with:

 doi:10.20944/preprints202605.1283.v1


```
python train.py --episodes 12 --robots 6 --tasks 16 \
  --max-steps 180 --imitation-steps 1800
```

CSV files, figures, and trained-model checkpoints are saved under the results directory.

22. Data and Code Availability

The simulator, planners, training scripts, generated CSV files, figures, and LaTeX source are included in the project directory. The implementation uses Python, NumPy, pandas, matplotlib, PyYAML, and PyTorch. No external proprietary dataset is required because maps and tasks are generated procedurally with fixed random seeds. For double-blind review, an anonymized repository URL or Zenodo DOI should be inserted here before journal submission.

23. Threats to Validity

The main internal validity threat is that the online adaptations of CBS/ECBS, MAPF-LNS2, and LaCAM* are not identical to their official offline solver implementations. The main external validity threat is that grid simulation abstracts away continuous robot dynamics, localization error, acceleration limits, and real charging hardware. The main conclusion validity threat is the limited number of seeds and the short GNN-PPO training horizon. We therefore report mean, standard deviation, and 95% confidence intervals for success metrics where appropriate, but avoid claims of statistical significance. The code supports larger seed counts through `python main.py -seeds 20`; these longer runs should be used before making strong statistical claims. These threats are mitigated by reporting exact settings, using multiple baselines, releasing the code, and avoiding unsupported claims about large-scale dominance.

24. Conclusion

This paper presented ECR-HR, an energy-constrained hybrid repair framework for lifelong warehouse MAPF. The proposed formulation incorporates energy, charging, payload, congestion, and collision constraints. A reproducible simulator and extensive baseline implementation were provided. Simulation results show that MAPF-LNS2-style repair achieves the best raw success and energy-per-task among the main baselines, while ECR-HR reduces candidate conflicts and charger conflicts relative to WHCA* in several settings. The new energy-feasible success metric shows why battery constraints must be evaluated explicitly rather than treated as a secondary log statistic. The GNN-PPO component demonstrates an executable but preliminary learning pipeline with imitation warm-start and PPO fine-tuning. Overall, the results indicate that energy-aware hybrid repair is a practical and reproducible direction for warehouse lifelong MAPF, but stronger policy training, official solver comparisons, additional seeds, and larger realistic warehouse layouts are required before claiming learning-based dominance.

Funding: This research received no external funding.
Code Availability: The source code, generated tables, figures, and LaTeX files are included in the project directory. An anonymized repository URL or archival DOI should be inserted before journal submission.
Data Availability: The procedural maps and tasks require no external proprietary dataset. CSV result files are provided under `results/tables/`.
Conflicts of Interest: The authors declare no conflicts of interest.

 doi:10.20944/preprints202605.1283.v1

References

1. Sharon, G.; Stern, R.; Felner, A.; Sturtevant, N.R. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* **2015**, *219*, 40–66. https://doi.org/10.1016/j.artint.2014.11.006.
2. Silver, D. Cooperative pathfinding. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2005.
3. Li, J.; Chen, Z.; Harabor, D.; Stuckey, P.J.; Koenig, S. MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2022, Vol. 36, pp. 10256–10265. https://doi.org/10.1609/aaai.v36i9.21266.
4. Okumura, K. Improving LaCAM for scalable eventually optimal multi-agent pathfinding. In Proceedings of the Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 2023, pp. 243–251.
5. Mao, J.; He, Z.; Li, D.; Li, R.; Zhang, S.; Wang, N. Multi-Agent Collaborative Path Planning Algorithm with Multiple Meeting Points. *Electronics* **2024**, *13*. https://doi.org/10.3390/electronics13163347.
6. Zou, Q.; Wang, H.; Zhang, T.; Li, Z.; Zhuang, Y. Research on Path Planning Method for Autonomous Patrol Robots. *Electronics* **2024**, *13*. https://doi.org/10.3390/electronics13142865.
7. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numerische Mathematik* **1959**, *1*, 269–271. https://doi.org/10.1007/BF01386390.
8. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* **1968**, *4*, 100–107. https://doi.org/10.1109/TSSC.1968.300136.
9. Dechter, R.; Pearl, J. Generalized best-first search strategies and the optimality of A*. *Journal of the ACM* **1985**, *32*, 505–536. https://doi.org/10.1145/3828.3830.
10. Yu, J.; LaValle, S.M. Structure and intractability of optimal multi-robot path planning on graphs. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2013.
11. Sartoretti, G.; Kerr, J.; Shi, Y.; Wagner, G.; Kumar, T.K.S.; Koenig, S.; Choset, H. PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics and Automation Letters* **2019**, *4*, 2378–2385.
12. Damani, M.; Luo, Z.; Wenzel, E.; Sartoretti, G. PRIMAL2: Pathfinding via reinforcement and imitation multi-agent learning – lifelong. *IEEE Robotics and Automation Letters* **2021**, *6*, 2666–2673. https://doi.org/10.1109/LRA.2021.3062803.
13. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Proceedings of the Advances in Neural Information Processing Systems, 2017.
14. Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Proceedings of the Proceedings of the International Conference on Machine Learning, 2018, pp. 4295–4304.
15. Yu, C.; Velu, A.; Vinitsky, E.; Gao, Y.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems* **2022**, *35*, 24611–24624.
16. Skrynnik, A.; Andreychuk, A.; Borzilov, A.; Chernyavskiy, A.; Yakovlev, K.; Panov, A. POGEMA: A benchmark platform for cooperative multi-agent navigation. *arXiv preprint arXiv:2407.14931* **2024**.
17. Zheng, H.; Ma, Y.; Araki, B.; Chen, J.; Wu, C. Learning-guided prioritized planning for lifelong multi-agent path finding in warehouse automation. *Journal of Artificial Intelligence Research* **2026**, *85*. https://doi.org/10.1613/jair.1.20611.
18. Standley, T. Finding optimal solutions to cooperative pathfinding problems. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2010, pp. 173–178.
19. Sharon, G.; Stern, R.; Goldenberg, M.; Felner, A. The increasing cost tree search for optimal multi-agent pathfinding. *Artificial Intelligence* **2013**, *195*, 470–495. https://doi.org/10.1016/j.artint.2012.11.006.
20. Ryan, M.R.K. Exploiting subgraph structure in multi-robot path planning. In Proceedings of the Journal of Artificial Intelligence Research Workshop and Symposium Proceedings, 2008.
21. Wagner, G.; Choset, H. Subdimensional expansion for multirobot path planning. *Artificial Intelligence* **2015**, *219*, 1–24. https://doi.org/10.1016/j.artint.2014.11.001.
22. Surynek, P. An optimization variant of multi-robot path planning is intractable. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2010.
23. Stern, R.; Sturtevant, N.R.; Felner, A.; Koenig, S.; Ma, H.; Walker, T.T.; Li, J.; Atzmon, D.; Cohen, L.; Kumar, T.K.S.; et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. In Proceedings of the Proceedings of the International Symposium on Combinatorial Search, 2019.

 doi:10.20944/preprints202605.1283.v1

24. Barer, M.; Sharon, G.; Stern, R.; Felner, A. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In Proceedings of the Proceedings of the International Symposium on Combinatorial Search, 2014.
25. Boyarski, E.; Felner, A.; Stern, R.; Sharon, G.; Tolpin, D.; Betzalel, O.; Shimony, S.E. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In Proceedings of the Proceedings of the International Symposium on Combinatorial Search, 2015.
26. Felner, A.; Li, J.; Boyarski, E.; Ma, H.; Cohen, L.; Kumar, T.K.S.; Koenig, S. Adding heuristics to conflict-based search for multi-agent path finding. In Proceedings of the Proceedings of the International Conference on Automated Planning and Scheduling, 2018, Vol. 28, pp. 83–87.
27. Cohen, L.; Uras, T.; Kumar, T.K.S.; Xu, H.; Ayanian, N.; Koenig, S. Conflict-based search for optimal multi-agent pathfinding with highways. In Proceedings of the Proceedings of the International Joint Conference on Artificial Intelligence, 2015, pp. 358–364.
28. Li, J.; Harabor, D.; Stuckey, P.J.; Ma, H.; Gange, G.; Koenig, S. Disjoint splitting for multi-agent path finding with conflict-based search. In Proceedings of the Proceedings of the International Conference on Automated Planning and Scheduling, 2019, Vol. 29, pp. 279–283.
29. Li, J.; Ruml, W.; Koenig, S. EECBS: A bounded-suboptimal search for multi-agent path finding. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2021.
30. Okumura, K.; Machida, M.; Défago, X.; Tamura, Y. Priority inheritance with backtracking for iterative multi-agent path finding. *Artificial Intelligence* **2022**, *310*, 103752. https://doi.org/10.1016/j.artint.2022.103752.
31. Luna, R.; Bekris, K.E. Push and swap: Fast cooperative path-finding with completeness guarantees. In Proceedings of the Proceedings of the International Joint Conference on Artificial Intelligence, 2011, pp. 294–300.
32. Ma, H.; Kumar, T.K.S.; Koenig, S. Searching with consistent prioritization for multi-agent path finding. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2019, Vol. 33, pp. 7643–7650. https://doi.org/10.1609/aaai.v33i01.33017643.
33. Ma, H.; Li, J.; Kumar, T.K.S.; Koenig, S. Lifelong multi-agent path finding for online pickup and delivery tasks. In Proceedings of the Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, 2017, pp. 837–845.
34. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. In Proceedings of the arXiv preprint arXiv:1707.06347, 2017.
35. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations, 2017.
36. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations, 2018.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, 2017.
38. Anonymous. ALPHA: Attention-based long-horizon pathfinding in highly-structured areas. *arXiv preprint arXiv:2310.08350* **2023**.
39. Zhang, S.; Zeng, Q. Online Unmanned Ground Vehicle Path Planning Based on Multi-Attribute Intelligent Reinforcement Learning for Mine Search and Rescue. *Applied Sciences* **2024**, *14*. https://doi.org/10.3390/app14199127.
40. You, D.; Yu, J.; Jia, Z.; Zhang, Y.; Yang, Z. Mobile Robot Path Planning Based on Fused Multi-Strategy White Shark Optimisation Algorithm. *Applied Sciences* **2025**, *15*. https://doi.org/10.3390/app15158453.
41. Hönig, W.; Kumar, T.K.S.; Cohen, L.; Ma, H.; Xu, H.; Ayanian, N.; Koenig, S. Multi-agent path finding with kinematic constraints. In Proceedings of the Proceedings of the International Conference on Automated Planning and Scheduling, 2016.
42. You, D.; Kang, S.; Yu, J.; Wen, C. Path Planning of Robot Based on Improved Multi-Strategy Fusion Whale Algorithm. *Electronics* **2024**, *13*. https://doi.org/10.3390/electronics13173443.
43. Fu, S.; Yang, D.; Mei, Z.; Zheng, W. Progress in Construction Robot Path-Planning Algorithms: Review. *Applied Sciences* **2025**, *15*. https://doi.org/10.3390/app15031165.
44. Wurman, P.R.; D'Andrea, R.; Mountz, M. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine* **2008**, *29*, 9–20.

 doi:10.20944/preprints202605.1283.v1

45. Boysen, N.; de Koster, R.; Weidinger, F. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research* **2019**, *277*, 396–411. https://doi.org/10.1016/j.ejor.2018.08.023.
46. Azadeh, K.; de Koster, R.; Roy, D. Robotized and automated warehouse systems: Review and recent developments. *Transportation Science* **2019**, *53*, 917–945. https://doi.org/10.1287/trsc.2018.0873.