robotics

MDPI

Article

LE-HG-PRM: A Structure-Aware Roadmap Planner for Intelligent Warehouse Logistics †

Siyuan Wang [1,*], Gongsen Wang [2,3], Feng Yang [4], Dawu Peng [5], Xingyu Yan [6], Shuyi Zhang [1], Xinyi Li [7] and Zhen Tian [8,*]

1 School of Electronic and Optical Engineering, Nanjing University of Science and Technology Zijin College, Nanjing 210023, China
2 State Key Laboratory of Transducer Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China
3 School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
4 School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210023, China
5 Hubei University of Technology Engineering and Technology College, Wuhan 430068, China
6 Aerospace Discipline, Changkong College, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
7 School of Business, Nanjing University of Science and Technology Zijin College, Nanjing 210023, China
8 James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK
* Correspondence: 19852225218@163.com (S.W.); 2620920z@student.gla.ac.uk (Z.T.)
† This paper is an extended version of our paper published in "Heuristic-guided PRM: A Dynamic Weighting Approach for Optimal Path Planning in Complex Environments", presented at the 2025 IEEE 7th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Yinchuan, China, 22–24 October 2025; pp. 715–723.

Abstract

Efficient AGV/AMR path planning is essential for intelligent warehouse logistics, where regular shelves, narrow aisles, local bottlenecks, and heterogeneous obstacles strongly affect roadmap quality. This study proposes LE-HG-PRM, a structure-aware extension of heuristic-guided probabilistic roadmap planning. The method embeds warehouse geometric priors into probability-field sampling, region-adaptive neighborhood connection, and cache-accelerated progressive path refinement. Compared with the preliminary conference version, the journal version introduces a redesigned warehouse-oriented planning framework and substantially expands the experimental validation. Four experimental campaigns are conducted, covering static-complexity progression, corridor-width sensitivity, parameter sensitivity, and map-scale expansion, with A*, JPS, PRM, RRT, RRT*, and HG-PRM as baselines. Each scenario uses 50 paired start–goal tasks, and sampling-based methods are repeated with 12 independent random seeds. The results show that LE-HG-PRM provides competitive path quality and structurally regular paths in representative warehouse layouts. Statistical tests further confirm that its path-length advantage is scenario-dependent but significant in several structured and bottleneck-constrained settings. The findings suggest that incorporating explicit warehouse-structure priors can improve roadmap-based global planning for intelligent logistics, while future work should validate the method in Gazebo and physical AGV/AMR platforms.

Keywords: warehouse logistics; AGV path planning; probabilistic roadmap; structure-aware sampling; occupancy grid; statistical validation

Check for updates

Academic Editor: Iyad Kanj

Received: 8 May 2026
Revised: 17 June 2026
Accepted: 24 June 2026
Published: 29 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

With the development of advanced manufacturing systems toward flexibility, digitalization, and intelligent automation, production-oriented warehouse logistics has become a key factor affecting material-flow efficiency, equipment utilization, and production-rhythm stability [1,2]. In conventional labor-intensive warehouses and shop-floor logistics units, limited planning efficiency and slow operational response increasingly constrain throughput, responsiveness, and system-level reliability [3,4]. Intelligent warehousing systems, represented by autonomous mobile robots (AMRs) and automated guided vehicles (AGVs), provide a practical solution by integrating perception, decision making, and automated execution into manufacturing logistics operations [5–10]. At the warehouse-system level, related studies have further highlighted the importance of robot path planning, coordinated routing, and multi-agent operation in structured logistics environments [11–14]. Within this pipeline, path planning plays a central role because it directly shapes material-flow continuity, operational cost, order-fulfillment speed, and overall manufacturing-system stability. Consequently, the rapid generation of safe, efficient, and high-quality paths in structured warehouse environments has become a problem of clear engineering importance for intelligent logistics and manufacturing automation.

For static path planning on known grid maps, existing approaches can be broadly divided into two categories: deterministic graph-search methods, such as A*, Dijkstra, and jump point search (JPS), which emphasize completeness and optimality or grid-search acceleration [15–17], and sampling-based methods, such as probabilistic roadways (PRMs) and rapidly-exploring random trees (RRTs), which offer probabilistic completeness and stronger scalability in high-dimensional spaces [18,19]. Graph-search methods can recover the shortest paths efficiently in regular environments, but their computational burden grows rapidly with increasing map size and resolution. By contrast, the standard PRM constructs a topological graph through random sampling and is therefore more flexible, yet it often struggles to capture narrow passages under limited sampling density. Moreover, because roadmap construction is typically not guided by scene-specific structural priors, both path quality and query efficiency can become unstable in production warehouse environments where shelves, aisles, workstations, and temporary storage zones form strong geometric constraints.

To overcome these limitations, a substantial body of work has explored either metaheuristic optimization or improved sampling strategies to enhance PRM-based planning [20–23]. Representative examples include hybrid PRM–genetic algorithm frameworks for joint path-length and smoothness optimization [24], low-dispersion PRM variants for improved sampling efficiency [25], narrow-passage-oriented PRM refinements [26], and PRM-based post-planning improvement combined with controller design [27]. Although these methods improve selected local metrics, their sampling logic still largely remains at the level of undirected exploration or posterior correction. As a result, the geometric priors embedded in highly structured manufacturing warehouses remain underexploited, leaving substantial room for improvement in path quality, scene adaptability, and engineering interpretability.

To address this issue, our previous conference work developed the heuristic-guided probabilistic roadmap (HG-PRM), which achieved a favorable trade-off between planning efficiency and path quality [28]. In large and complex maps, HG-PRM was shown to generate high-quality paths within 0.125 s, reducing planning time by 94.4% and path length by 17.2% relative to A*. These results suggest substantial engineering potential under limited computational budgets. Nevertheless, in highly structured static warehousing scenarios, HG-PRM still inherits an important limitation from generic sampling-based planners: insufficient exploitation of environment geometry during roadmap generation.

This can lead to undersampling in critical regions and redundant exploration in non-critical free space, thereby constraining further gains in path quality, structural regularity, and deployment predictability for production logistics.

A preliminary version of this work was presented at IEEE ICCASIT 2025 as a conference paper on heuristic-guided PRM [28]. The present journal version differs substantially from that conference version in both method and validation. Specifically, it extends the original HG-PRM idea from heuristic-biased sampling on a single large grid map to a structure-aware warehouse-planning framework with explicit geometric probability fields, dynamic neighborhood construction, cache-accelerated progressive refinement, four newly organized experimental campaigns, 50 paired start–goal tasks per scenario, distribution-level analysis, and statistical significance testing. These additions provide more than 50% newly generated experimental data and analytical content and make the current article a complete journal extension rather than a duplicate conference publication.

Building on HG-PRM, this study proposes a logistics-enhanced heuristic-guided probabilistic roadmap, termed LE-HG-PRM, for static path planning in production-oriented intelligent warehouses. The proposed method introduces three collaborative mechanisms: environment-feature-aware sampling, hierarchical dynamic neighbor connection, and cache-accelerated progressive optimization. Together, these mechanisms shift roadmap construction from uniform random exploration toward geometry-guided generation, enabling the planner to produce paths that are safer, smoother, and more globally efficient under limited computational resources.

The main contribution of this work is not to replace existing deterministic or sampling-based planners in a universal sense, but to embed warehouse-specific geometric priors into the sampling, connection, and refinement stages of HG-PRM. Compared with generic PRM and the previous HG-PRM, LE-HG-PRM explicitly biases sampling toward bottlenecks, traffic corridors, and task-relevant regions, adaptively increases local connectivity in constrained areas, and reuses collision-checking information during progressive path refinement. In this sense, the novelty of the work lies in converting PRM-style roadmap construction from a passive random exploration process into a geometry-conditioned planning process for structured warehouse logistics. These mechanisms make the planner particularly suitable for production warehouse layouts where regular shelves, narrow aisles, heterogeneous obstacles, and task-oriented traffic corridors coexist.

1.1. Conference Extension Statement.

This manuscript is an extended version of the ICCASIT 2025 conference paper [28]. The journal version has a substantially revised title and abstract, introduces new warehouse-structure-aware algorithmic components, adds multi-scenario and multi-scale experimental campaigns with 50 paired tasks per scenario, and includes distribution-level and statistical significance analyses, with more than 50% newly generated experimental data and analytical content.

1.2. Key Technical Contribution

The technical novelty of LE-HG-PRM is organized around three coupled mechanisms. First, geometry-aware probabilistic sampling replaces uniform or purely heuristic sampling with distance-transform-driven spatial priors, so that limited samples are preferentially allocated to bottlenecks, corridor cores, and task-relevant regions. Second, adaptive neighborhood construction dynamically increases local graph connectivity in constrained areas, improving roadmap robustness where feasible passages are most likely to be broken. Third, cache-accelerated progressive refinement reduces redundant collision checking and converts the raw graph path into a shorter and more executable route. Together,

these components form a structure-aware roadmap generation framework rather than a single-parameter modification of PRM.

The remainder of this paper is organized as follows. Section 2 presents the modeling framework and pseudocode of LE-HG-PRM. Section 3 describes the warehouse logistics environment model. Section 4 details the experimental setup and task-generation protocol. Section 5 reports the experimental results, including mean-value comparisons, distribution-level analysis, and statistical significance validation. Section 6 concludes the paper and outlines future directions for manufacturing-logistics deployment. Figure 1 summarizes the overall organization of the manuscript.

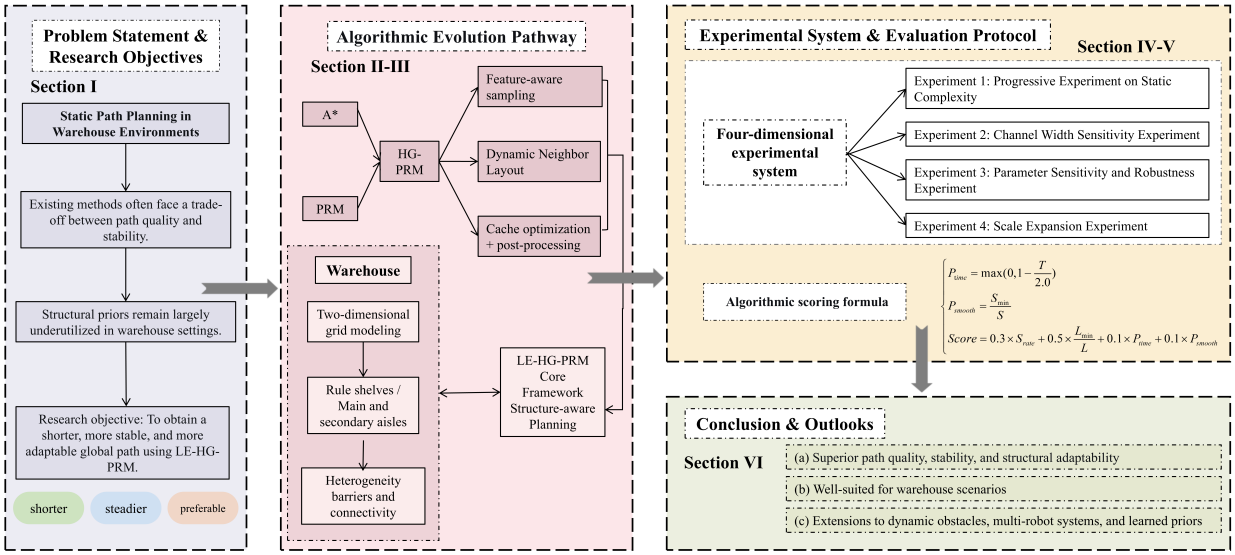


Figure 1. Overall structure of the article. Section I defines the problem statement and research objective; Sections II–III present the algorithmic evolution pathway and warehouse-environment modeling; Sections IV–V describe the experimental system, evaluation protocol, performance analysis, and statistical validation; Section VI summarizes the main conclusions and future directions.

2. LE-HG-PRM Algorithm

*2.1. A**

A* is a classical heuristic shortest-path search algorithm. It extends Dijkstra's algorithm by introducing a heuristic function to estimate the minimum movement cost from the current state to the goal state, thereby selecting the best path. The principle of the algorithm is shown in Figure 2, and the evaluation function is defined as follows:

$$f(n) = g(n) + h(n), \tag{1}$$

where $f(n)$ denotes the estimated cost from the start state to the goal state, $g(n)$ denotes the accumulated cost from the start state to the current state, and $h(n)$ denotes the estimated cost from the current state to the goal state.

2.2. PRM

Probabilistic Roadmap (PRM) is a sampling-based path planning algorithm for motion planning in high-dimensional continuous state spaces. Its core idea is to construct a roadmap offline and then query feasible paths online. The procedure mainly consists of a preprocessing stage and a query stage, as shown in Figure 3.

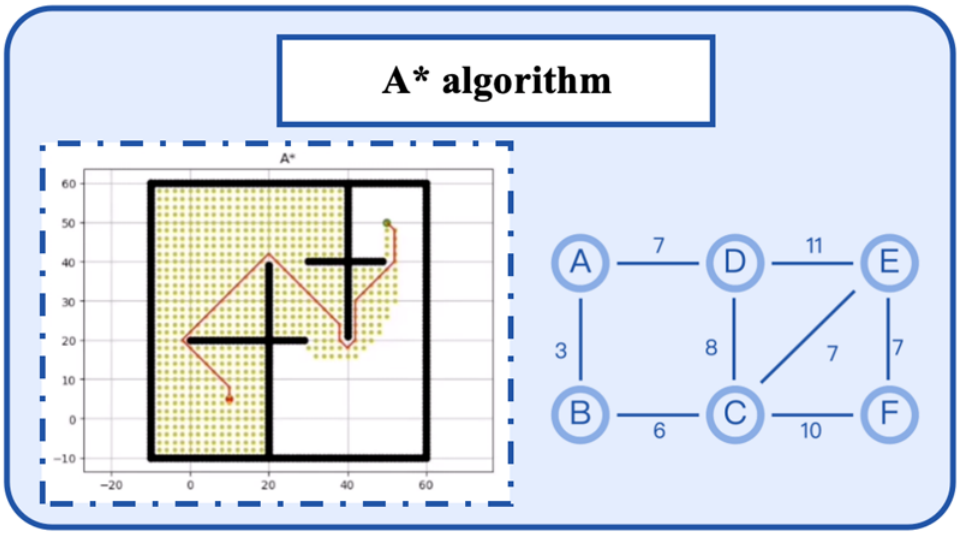


Figure 2. Schematic diagram of the A* algorithm. (Letters denote graph nodes, and numbers denote illustrative edge costs.)

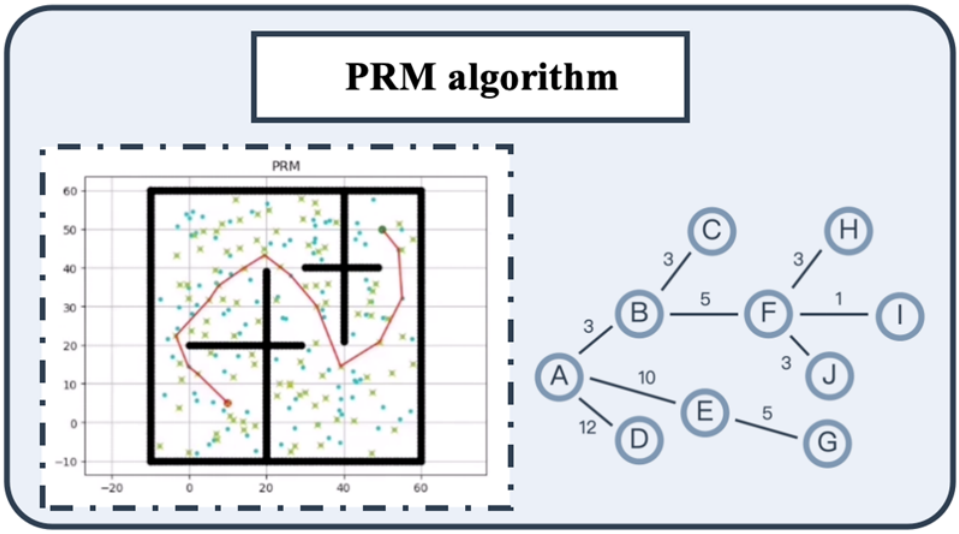


Figure 3. Schematic diagram of the PRM algorithm. (The colored points and connecting lines indicate sampled configurations and feasible roadmap edges, respectively.)

2.2.1. Preprocessing Stage

In the collision-free region C_{free} of the state space C, n collision-free configuration points are randomly sampled to form the vertex set:

$$V = \{q_i \in C_{\text{free}} \mid i = 1, \ldots, n\}. \tag{2}$$

For each vertex q_i, the algorithm attempts to connect it with neighboring vertices q_j within a given Euclidean distance r, thereby forming the edge set E. If the line segment connecting q_i and q_j lies completely in C_{free}, the edge is retained; otherwise, it is removed. The final undirected graph is written as

$$G = (V, E). \tag{3}$$

2.2.2. Query Stage

Given the start point q_{start} and the goal point q_{goal}, they are first connected to several nearest vertices in G according to distance thresholds and collision detection, yielding the extended graph G'. A graph search algorithm is then used to find the shortest path π from q_{start} to q_{goal} on G':

$$\pi = \arg\min_{\text{path} \subseteq G} \sum_{e \in \text{path}} \text{Cost}(e), \tag{4}$$

where $\text{Cost}(e)$ denotes the cost of edge e, usually defined as the Euclidean distance.

2.3. HG-PRM

This subsection summarizes the HG-PRM baseline proposed in our ICCASIT 2025 conference paper [28]. It is retained here as the methodological starting point and as the direct predecessor used for comparison with LE-HG-PRM. Traditional A* has low search efficiency in complex environments, while PRM suffers from sampling blindness and high path redundancy. Based on the complementary characteristics of A* and PRM, HG-PRM is improved in the following four aspects:

1. **Adaptive sampling**: a heuristic guidance mechanism is introduced on the basis of random sampling to generate samples more efficiently in free space.
2. **Efficient roadmap construction**: neighboring nodes are queried rapidly using a KD-tree, and feasible edges are constructed through collision detection to improve graph construction efficiency.
3. **Graph-constrained A* search**: the improved A* algorithm is run on the constructed roadmap, and the graph topology constraint is used to reduce the search space.
4. **Post-path optimization**: pruning optimization is used to eliminate redundant nodes and obtain a smoother and shorter final path.

The pseudocode of HG-PRM is given in Algorithm 1.

Algorithm 1 Heuristic-Guided Probabilistic Roadmap (HG-PRM)

Require: Map M; start configuration q_{start}; goal configuration q_{goal}; number of samples N; number of neighbors K.
Ensure: Optimized path P_{opt}; runtime T; number of vertices $|V|$; number of edges $|E|$.

Step 1: Initialization and Adaptive Sampling
$T_{\text{start}} \leftarrow \text{time}()$
$V \leftarrow \{q_{\text{start}}, q_{\text{goal}}\} \cup \text{ADAPTIVE\_SAMPLE}(M, N-2)$
$|V| \leftarrow |V|$
Step 2: Roadmap Construction with KD-tree
$(G, |E|) \leftarrow \text{BUILD\_ROADMAP}(V, K, M)$
Step 3: Graph Search on Roadmap
$P_{\text{raw}} \leftarrow \text{A\_STAR}(G, q_{\text{start}}, q_{\text{goal}})$
if $P_{\text{raw}} = \emptyset$ **then**
    $T \leftarrow \text{time}() - T_{\text{start}}$
    return $(\emptyset, T, |V|, |E|)$
end if
Step 4: Path Shortcutting and Final Metrics
$P_{\text{opt}} \leftarrow \text{PATH\_SHORTCUT}(P_{\text{raw}}, M)$
$T \leftarrow \text{time}() - T_{\text{start}}$
return $(P_{\text{opt}}, T, |V|, |E|)$

The four subroutines used in Algorithm 1 are listed in Algorithms 2–5, including adaptive sampling, roadmap construction, collision checking, and path shortcutting.

Algorithm 2 Subroutine 1: ADAPTIVE_SAMPLE

Require: Map M; required samples n.
Ensure: Sample set V_{free}.
$F \leftarrow$ set of free cells in M
$n_{\text{eff}} \leftarrow \min(n, |F|)$
$V_{\text{free}} \leftarrow \text{RANDOM\_SUBSET}(F, n_{\text{eff}})$
return V_{free}

Algorithm 3 Subroutine 2: BUILD_ROADMAP

Require: Vertex set V; neighbors K; map M.
Ensure: Graph G; number of edges $|E|$.
$G \leftarrow \text{INIT\_GRAPH}(V)$
$T_{\text{KD}} \leftarrow \text{BUILD\_KD\_TREE}(V)$
for all $v \in V$ **do**
    $N_v \leftarrow \text{K\_NEAREST}(T_{\text{KD}}, v, K+1) \setminus \{v\}$
    for all $u \in N_v$ **do**
        if $\text{COLLISION\_FREE}(v, u, M)$ **then**
            $\text{ADD\_EDGE}(G, v, u, \|v - u\|)$
        end if
    end for
end for
$|E| \leftarrow$ number of edges in G
return $(G, |E|)$

Algorithm 4 Subroutine 3: COLLISION_FREE

Require: Points a, b; map M.
Ensure: Boolean.
for all grid point p on $\text{BRESENHAM\_LINE}(a, b)$ **do**
    if p is not in M.free_space **then**
        return false
    end if
end for
return true

Algorithm 5 Subroutine 4: PATH_SHORTCUT

Require: Raw path P; map M.
Ensure: Optimized path P_{opt}.
$P_{\text{opt}} \leftarrow [P[0]]$
$i \leftarrow 0$
while $i < |P| - 1$ **do**
    success $\leftarrow$ **false**
    for $j \leftarrow |P| - 1$ **down to** $i + 1$ **do**
        if $\text{COLLISION\_FREE}(P[i], P[j], M)$ **then**
            $P_{\text{opt}}.\text{append}(P[j])$
            $i \leftarrow j$
            success $\leftarrow$ **true**
            break
        end if
    end for
    if success $=$ **false then**
        $P_{\text{opt}}.\text{append}(P[i+1])$
        $i \leftarrow i + 1$
    end if
end while
return P_{opt}

2.4. LE-HG-PRM

Building on the conference HG-PRM framework [28], this paper further improves the algorithm to address the sampling blindness that still exists in highly structured static warehousing scenarios. Because the geometric characteristics of the environment are not fully exploited, the original strategy may lead to insufficient sampling in critical regions and redundant sampling in non-critical regions, which limits further improvement in path quality. Compared with the conference method, LE-HG-PRM does not merely retune the sample size or neighbor number; instead, it introduces warehouse-specific probability-field modeling, region-adaptive neighborhood connection, and cached progressive path refinement.

Therefore, this paper proposes LE-HG-PRM. Inheriting the efficient framework of HG-PRM, the proposed method transforms sampling and roadmap construction from passive uniform exploration to active feature perception through three core mechanisms:

1. A non-uniform sampling probability field is constructed based on distance transform and region segmentation, guiding sampling points to be densely distributed in narrow bottlenecks, traffic channels, and task-related areas.
2. The node connection degree is dynamically adjusted according to the regional characteristics of the node to realize hierarchical neighbor connection.
3. A collision detection cache and a progressive path-shortening strategy based on a local look-ahead window are introduced to improve computational efficiency and generate smoother paths that better fit practical motion constraints.

The core innovation of the algorithm is to realize the collaborative enhancement of sampling, roadmap construction, and optimization by sensing the geometric structure of the environment. This design also clarifies the extension relationship with HG-PRM: HG-PRM provides the basic sampling–roadmap–search pipeline, whereas LE-HG-PRM adds warehouse-structure awareness at all three stages.

Algorithm Modeling

(a) Environmental feature extraction and sampling probability field construction.

When the algorithm is initialized, a one-time geometric feature analysis is performed on the map M to provide prior knowledge for intelligent sampling. The distance from each free grid to the nearest obstacle is calculated to obtain the distance field D:

$$D = \text{DistanceTransform}(M). \tag{5}$$

Based on the distance field D, feature regions are adaptively defined and extracted:

$$\begin{cases} \mathcal{R}_n = \{p \in C_{\text{free}} \mid D(p) < \tau_n\}, & \text{(Constricted region)}, \\ \mathcal{R}_w = \{p \in C_{\text{free}} \mid D(p) > \tau_w\}, & \text{(Broad region)}, \\ \mathcal{R}_c = \text{Erode}_{3\times3}(C_{\text{free}}), & \text{(Channel-core region)}. \end{cases} \tag{6}$$

Here, $\tau_n = \max(3, \lfloor \min(H, W)/50 \rfloor)$ and $\tau_w = \max(10, \lfloor \min(H, W)/20 \rfloor)$. The lower bounds in τ_n and τ_w prevent degenerate region extraction in small maps, while the scale-dependent terms allow the constricted and broad regions to vary with map resolution. Specifically, the divisor 50 yields a relatively small threshold for identifying narrow passages, whereas the divisor 20 yields a larger threshold for distinguishing broad open regions. $\text{Erode}_{3\times3}(\cdot)$ denotes morphological erosion using a 3×3 structuring element, which is used here to extract the channel-core region. These values are not intended as universally optimal constants, but as practical default settings for structured manufacturing warehouses. To improve faithful reimplementation, the thresholds are expressed as explicit functions of the map size rather than as hidden constants. In additional threshold-

perturbation checks, moderate variations around these default values did not change the qualitative ranking of the proposed method, indicating that the planner is primarily driven by the relative identification of constrained and broad regions rather than by a single exact threshold value.

Different base sampling probabilities are assigned to different regions to prioritize the exploration of critical areas:

$$P_b(p) = \begin{cases} 0.8, & \text{if } p \in \mathcal{R}_n, \\ 0.6, & \text{if } p \in \mathcal{R}_c \setminus \mathcal{R}_n, \\ 0.2, & \text{if } p \in C_{\text{free}} \setminus (\mathcal{R}_n \cup \mathcal{R}_c \cup \mathcal{R}_w), \\ 0.1, & \text{if } p \in \mathcal{R}_w. \end{cases} \tag{7}$$

The probability values in Equation (7) are heuristic priors assigned according to the expected planning importance of different warehouse regions rather than fitted constants. Constricted regions and channel skeletons receive higher probabilities because they are more likely to determine roadmap connectivity and path feasibility, whereas broad open regions receive lower probabilities to reduce redundant sampling. The selected ordering $0.8 > 0.6 > 0.2 > 0.1$ therefore encodes a priority hierarchy from bottleneck-critical regions to redundant open space. These values are not intended as universally optimal constants, but as a practical default setting for structured manufacturing warehouses; their influence is further examined through the parameter-sensitivity analysis in Experiment III, where the narrow_first, adaptive, and uniform strategies are compared to test whether the method relies on a narrowly tuned sampling policy.

Gaussian influence fields $I_s(p)$ and $I_g(p)$ are superimposed at the start point q_s and the goal point q_g to enhance the sampling density of task-relevant regions. By fusing the base probability field and the influence fields, the final probability distribution used to guide sampling is obtained as

$$P_f(p) = \max\left(P_b(p), I_s(p), I_g(p)\right). \tag{8}$$

(b) Intelligent sampling based on probability fields.

The generation of the sampling point set V is no longer a uniform random process. The algorithm first fixes the inclusion of q_s and q_g. Then, from all free raster cells M_{free}, weighted random sampling without replacement is performed according to the normalized probability P_f until the number of sampling points reaches the preset value.

(c) Dynamic-neighbor roadmap construction.

For the sampling point set V, the roadmap $G = (V, E)$ is constructed. The connection strategy is dynamically adjusted according to the position of each node $v_i \in V$, and the actual number of connections $k_a(i)$ is determined by the region in which the node is located:

$$k_a(i) = \begin{cases} \min(2k, |V| - 1), & \text{if } v_i \in \mathcal{R}_n, \\ \min(\lfloor 1.5k \rfloor, |V| - 1), & \text{if } v_i \in \mathcal{R}_c \setminus \mathcal{R}_n, \\ k, & \text{otherwise}, \end{cases} \tag{9}$$

where k is the baseline number of neighbors. This strategy increases connections in bottleneck regions to improve connectivity.

For any point pair, when connecting nodes, the connectivity detection result is stored in a hash cache. This avoids repeated Bresenham line scanning on the same segment and improves the efficiency of roadmap construction.

(d) Path search with progressive post-optimization.

On the constructed roadmap G, the A* algorithm is used to search for the initial path π_{raw}. Subsequently, a progressive path-shortening strategy is used for optimization. This strategy initializes the optimized path π_{opt} at the starting point and sets the maximum forward search window. From the current point, collision detection is performed from far to near within the window. If a remote node can be directly reached through the cache mechanism, it is connected and the intermediate nodes are skipped. Compared with the global backtracking shortening strategy, the local forward search method has lower computational overhead and can generate a path composed of fewer and longer straight segments, which better conforms to the motion characteristics of warehouse AGVs.

The pseudocode of LE-HG-PRM is given in Algorithm 6.

Algorithm 6 Logistics-Enhanced HG-PRM (LE-HG-PRM)

Require: Map M; start q_s; goal q_g; samples N; neighbors K.
Ensure: path π_{opt}; runtime T; $|V|$.

$T_{\text{start}} \leftarrow \text{time}()$
▷ 1. Feature extraction and probability field construction
$D \leftarrow \text{DistanceTransform}(M)$
$R_{\text{narrow}} \leftarrow \{p \mid D(p) < \max(3, \min(H, W)/50)\} \cap M_{\text{free}}$
$R_{\text{wide}} \leftarrow \{p \mid D(p) > \max(10, \min(H, W)/20)\} \cap M_{\text{free}}$
$R_{\text{channel}} \leftarrow \text{Erode}(M_{\text{free}}, 3 \times 3)$
$P_f \leftarrow \text{Compute\_Probability\_Field}(R_{\text{narrow}}, R_{\text{wide}}, R_{\text{channel}}, q_s, q_g)$
▷ 2. Smart sampling
$V \leftarrow \{q_s, q_g\} \cup \text{Weighted\_Sample}(M_{\text{free}}, P_f, N-2)$
▷ 3. Dynamic neighborhood composition
$G \leftarrow (V, \emptyset)$; Cache $\leftarrow \emptyset$
$T_{\text{kd}} \leftarrow \text{Build\_KDTree}(V)$
for v_i in V **do**
    $k_{\text{actual}} \leftarrow \text{GetDynamicK}(v_i, K, R_{\text{narrow}}, R_{\text{channel}}, |V|)$
    for v_j in $\text{K\_Nearest}(T_{\text{kd}}, v_i, k_{\text{actual}}+1) \setminus \{v_i\}$ **do**
        if $\text{Cached\_CollisionCheck}(v_i, v_j, M, \text{Cache})$ **then**
            $G.\text{add\_edge}(v_i, v_j, \|v_i - v_j\|_2)$
        end if
    end for
end for
▷ 4. Path search and optimization
$\pi_{\text{raw}} \leftarrow \text{A\_Star}(G, q_s, q_g)$
if $\pi_{\text{raw}} = \emptyset$ **then**
    return $(\emptyset, \text{time}() - T_{\text{start}}, |V|)$
end if
$\pi_{\text{opt}} \leftarrow [\pi_{\text{raw}}[0]]$; $i \leftarrow 0$
while $i < |\pi_{\text{raw}}| - 1$ **do**
    for $j = \min(i+10, |\pi_{\text{raw}}| - 1)$ **down to** $i+1$ **do**
        if $\text{Cached\_CollisionCheck}(\pi_{\text{opt}}[-1], \pi_{\text{raw}}[j], M, \text{Cache})$ **then**
            $\pi_{\text{opt}}.\text{append}(\pi_{\text{raw}}[j])$; $i \leftarrow j$; **break**
        end if
    end for
end while
return $(\pi_{\text{opt}}, \text{time}() - T_{\text{start}}, |V|)$

3. Warehouse Logistics Environment Modeling

3.1. Modeling of the Physical Warehouse Environment

In this study, the warehouse environment is modeled as a two-dimensional grid map M, in which the traversability of each grid cell is represented by a binary state variable. Specifically,

$$M(x,y) = \begin{cases} 1, & \text{if the grid cell } (x,y) \text{ belongs to free space,} \\ 0, & \text{if the grid cell } (x,y) \text{ is occupied by an obstacle.} \end{cases} \tag{10}$$

Let the start node $\mathbf{q}_s$ and the goal node $\mathbf{q}_g$ both lie in the free region of the map. The task of static path planning is then to find a collision-free path

$$\pi = \{\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_m\}, \tag{11}$$

such that

$$\mathbf{q}_0 = \mathbf{q}_s, \qquad \mathbf{q}_m = \mathbf{q}_g, \tag{12}$$

and every node on the path satisfies

$$\mathbf{q}_i \in M_{\text{free}}, \qquad i = 0, 1, \ldots, m, \tag{13}$$

where M_{free} denotes the set of all traversable cells in the map.

In addition, for any two adjacent nodes $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ on the path, the straight-line segment connecting them must not intersect any obstacle region. This constraint can be written as

$$\text{Line}(\mathbf{q}_i, \mathbf{q}_{i+1}) \subseteq M_{\text{free}}, \qquad i = 0, 1, \ldots, m-1. \tag{14}$$

Accordingly, the warehouse path-planning problem considered in this paper can be formulated as the search for a feasible path connecting $\mathbf{q}_s$ and $\mathbf{q}_g$ while satisfying collision-avoidance constraints throughout the entire route.

3.2. Wall and Boundary Representation

The present experiments use occupancy-grid maps in which shelves, heterogeneous obstacles, corridor boundaries, and blocked regions are represented as occupied cells. In this representation, explicit wall polygons are not modeled as a separate semantic category; however, any wall or outer boundary can be encoded by the same occupied-cell mechanism used for obstacle collision checking. Recent learning-based navigation studies have shown that explicit wall and environmental-feature modeling can be useful for path-constrained robot navigation, especially when local navigation, wall-following behavior, or sensor-level interaction is considered [29]. We agree that the lack of an independent wall-semantic layer is a limitation for deployment-level navigation. In the present study, the focus is static global planning in structured warehouse layouts, and the occupancy-grid abstraction is sufficient to test roadmap connectivity, bottleneck sampling, and collision-free path generation. Future Gazebo or physical-robot validation should include explicit wall geometry, robot footprint inflation, localization uncertainty, and perception constraints.

3.3. Structural Characteristics of Warehouse Logistics Environments

Combined with the spatial layout of real warehouse systems, the environment model in this paper focuses on the following structural characteristics.

(a) Regular grid layout.

Storage shelves are typically arranged in rows and columns with relatively uniform spacing, forming a highly regular grid-like spatial pattern. As a result, the aisle widths between adjacent shelves are often approximately consistent.

(b) Hierarchical corridor structure.

Warehouse traffic spaces usually exhibit a primary–secondary corridor structure. Main corridors are wider and primarily used for backbone transportation and vehicle circulation, whereas secondary shelf aisles are narrower and mainly support picking and local-access operations.

(c) Obstacle heterogeneity.

In addition to regular shelf structures, real warehouse environments may also contain heterogeneous obstacles, such as columns, workstations, temporary storage areas, and other irregular blocking elements. These obstacles increase local geometric complexity and may generate bottlenecks or fragmented feasible regions.

(d) Spatial connectivity.

To ensure the existence of a feasible path-planning task, the environment must preserve at least one traversable passage connecting the start and goal regions. In other words, the free space should maintain global connectivity at the task level, even when local narrow passages or obstacle clusters are present.

These structural characteristics together define the geometric prior of warehouse logistics environments. They also provide the basis for the structure-aware sampling and roadmap-construction strategy adopted in LE-HG-PRM.

4. Experimental Environment and Evaluation Metrics

4.1. Experimental Design

To systematically evaluate the comprehensive performance of LE-HG-PRM, this study constructs an experimental framework from four complementary dimensions: environment complexity, corridor width, parameter sensitivity, and map-scale expansion. Specifically, Experiment I is designed to examine basic path quality under progressively increasing structural complexity; Experiment II evaluates the ability of the algorithm to preserve path regularity under corridor contraction; Experiment III identifies the effective operating interval of key hyperparameters; and Experiment IV investigates the degradation pattern of the algorithm under scale expansion.

Taken together, these four experiments are intended to assess not only whether LE-HG-PRM can generate shorter paths, but also whether it can maintain structural stability, deployment consistency, and engineering interpretability across diverse manufacturing-warehouse scenarios.

4.2. Experiment I: Static-Complexity Progression Scenario

Experiment I considers three representative warehouse maps with progressively increasing complexity, as shown in Figure 4a(i)–a(iii). The first map is a simple regular warehouse environment, with an obstacle density of approximately 20%, regularly arranged shelf structures, and a clear corridor layout. This scenario is mainly used to evaluate the basic planning capability of each algorithm in structured environments.

The second map corresponds to a medium-complexity warehouse environment, with an obstacle density of approximately 35%. Random rectangular obstacles and lo-

cal main corridors are introduced into the regular layout to simulate the coexistence of heterogeneous obstacles and local bottlenecks in realistic warehouse scenes.

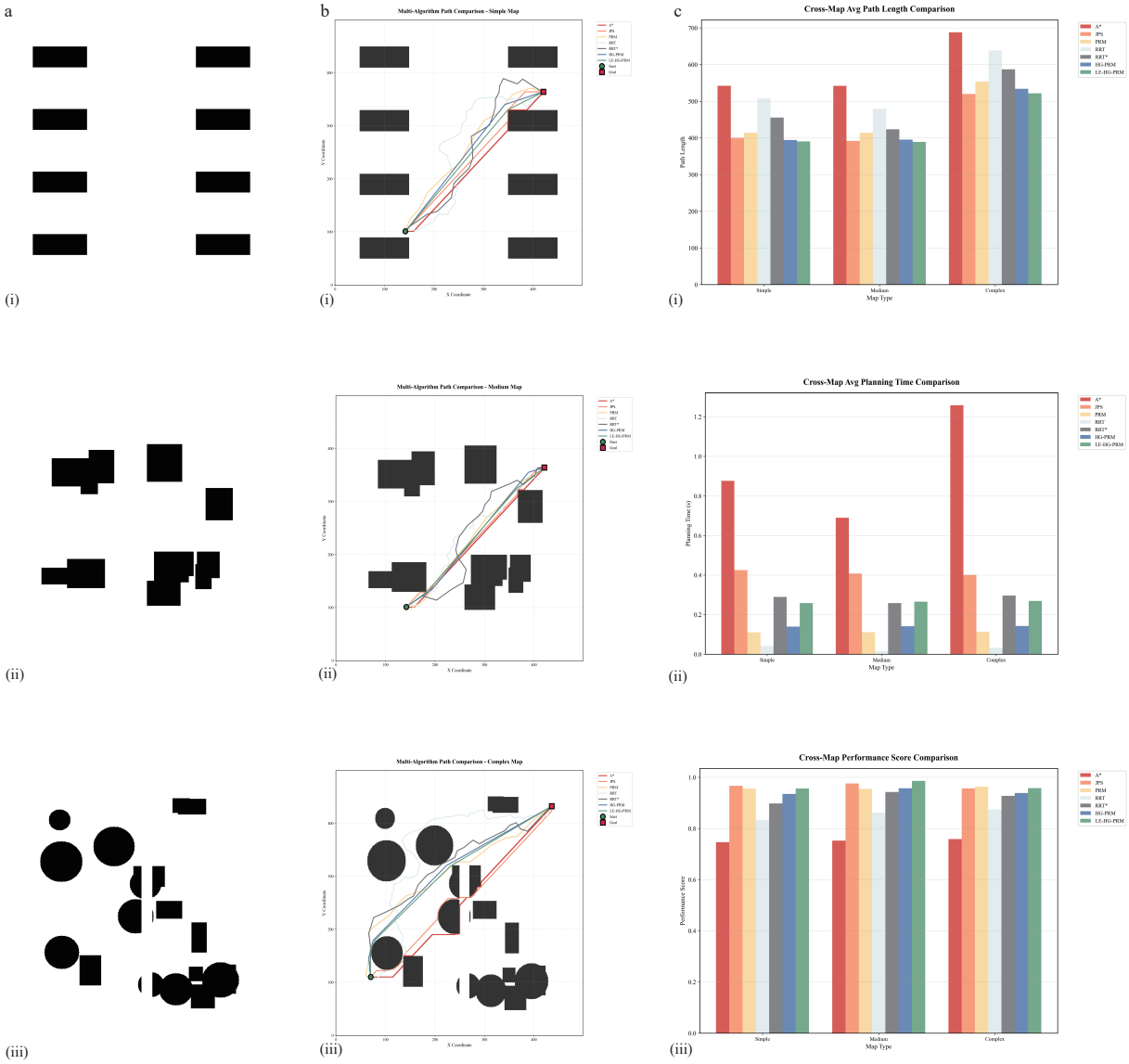


Figure 4. Experimental environments, representative paths, and cross-scenario statistics under different static-complexity settings. (**a(i)–a(iii)**) Simple regular, medium-complexity, and high-density complex warehouse maps. (**b(i)–b(iii)**) Representative path-planning results of the compared algorithms in the corresponding environments. (**c(i)–c(iii)**) Cross-map comparisons of average path length, average planning time, and overall performance score.

The third map is a high-density complex environment, with an obstacle density of approximately 50%. It includes rectangular, circular, and L-shaped obstacles, together with narrow passages, and is used to test whether the algorithm can effectively identify and exploit critical feasible regions under strong geometric constraints and high levels of occlusion.

Representative path-planning results for these three scenarios are presented in Figure 4b(i)–b(iii), while the corresponding cross-scenario statistics, including success rate, average planning time, and overall performance score, are summarized in Figure 4c(i)–c(iii).

4.3. Experiment II: Corridor-Width Sensitivity Scenario

Experiment II focuses on the effect of corridor width on path-planning behavior. Three corridor environments are considered: a wide main corridor, a standard storage corridor, and a narrow picking corridor.

Unlike Experiment I, which emphasizes structural complexity, this experiment focuses on whether the planned path can still maintain centerline conformity, corner control, and overall executability as the feasible corridor is progressively compressed.

Representative planning results and cross-scenario comparisons are analyzed in Section 5.3.

4.4. Experiment III: Parameter-Sensitivity Scenario and Variable Settings

Experiment III investigates the response of LE-HG-PRM to key hyperparameters under a fixed test scenario.The purpose of using a fixed environment is to isolate parameter effects from environmental variability and thereby obtain a clearer understanding of algorithmic sensitivity.

The tested variables include:

1. The number of sampling points, denoted by n_{points};
2. The number of neighbor connections, denoted by $k_{\text{neighbors}}$;
3. The sampling strategy.

This experiment is intended to determine whether LE-HG-PRM exhibits a sufficiently wide interval of effective parameter settings, rather than relying on a narrowly tuned optimal point. The mean-value and distribution-level results are analyzed in Section 5.4.

4.5. Experiment IV: Scale-Expansion Scenario

Experiment IV is used to evaluate the scalability of the proposed method under different map sizes. Three map scales are considered: 200×200, 400×400, and 600×600.

All three maps adopt random obstacle layouts while preserving a traversable main-corridor structure, so as to simulate the scale expansion from a local picking unit to a larger regional logistics network. This experiment is designed to reveal whether the path-quality advantage of LE-HG-PRM can remain stable as the workspace increases and whether its computational cost exhibits a controllable degradation trend.

Representative planning results and cross-scale statistical comparisons are analyzed in Section 5.5.

4.6. Task Generation and Evaluation Metrics

For each scenario, a fixed set of paired planning trials was generated in the free space and shared by all compared algorithms. To avoid overly trivial tasks, the start and goal nodes were required to maintain sufficient geometric separation. Unless otherwise specified, each scenario contained $N_{\text{total}} = 50$ paired start–goal tasks. For sampling-based algorithms, each task was repeated 12 times using independent random seeds, while deterministic algorithms were evaluated on the same task set. The same map, start–goal pairs, and evaluation pipeline were used for all compared methods. The final statistics were averaged over valid repeated runs to reduce the influence of individual randomness on the comparison results.

The performance of each algorithm is evaluated using the following core metrics.

(a) Success rate.

Success rate measures the proportion of independent runs in which a collision-free path is successfully found. It reflects the basic usability and robustness of the algorithm:

$$S_{\text{rate}} = \frac{N_{\text{success}}}{N_{\text{total}}}, \tag{15}$$

where N_{success} is the number of successful planning runs and N_{total} is the total number of runs.

(b) Average path length.

Average path length characterizes the global efficiency of the generated path and is defined as the average accumulated Euclidean length of successful paths:

$$L = \frac{1}{N_{\text{success}}} \sum_{k=1}^{N_{\text{success}}} \sum_{i=0}^{m_k-1} \left\| \mathbf{q}_{i+1}^{(k)} - \mathbf{q}_i^{(k)} \right\|_2, \tag{16}$$

where $\mathbf{q}_i^{(k)}$ denotes the ith node on the kth successful path.

(c) Average planning time.

Average planning time measures the real-time cost of the algorithm and is defined as the average CPU time from algorithm initialization to final path output:

$$T = \frac{1}{N_{\text{total}}} \sum_{k=1}^{N_{\text{total}}} t_k, \tag{17}$$

where t_k denotes the planning time of the kth run.

(d) Average path smoothness.

Average path smoothness is used to characterize path continuity and motion friendliness. It is defined by the mean turning-angle variation between adjacent path segments:

$$S = \frac{1}{N_{\text{success}}} \sum_{k=1}^{N_{\text{success}}} \frac{1}{m_k - 1} \sum_{i=1}^{m_k-1} \left| \theta_i^{(k)} - \theta_{i-1}^{(k)} \right|, \tag{18}$$

where $\theta_i^{(k)}$ denotes the direction angle of the ith segment on the kth path. A smaller value indicates a smoother path.

To avoid any single metric dominating the evaluation, this study further defines an overall performance score that jointly considers feasibility, path quality, time efficiency, and smoothness. First, the planning-time score is defined as

$$P_{\text{time}} = \max\left(0.1,\ 1 - \frac{T}{2.0}\right), \tag{19}$$

and the smoothness score is defined as

$$P_{\text{smooth}} = \frac{S_{\min} + \varepsilon}{S + \varepsilon}, \tag{20}$$

where $S_{\min}$ is the minimum average path smoothness among the compared algorithms in the same experimental group, and $\varepsilon = 10^{-6}$ is a small constant used to avoid division by zero when the minimum smoothness is zero.

The final comprehensive score is then calculated as

$$\text{Score} = 0.3 \times S_{\text{rate}} + 0.5 \times \frac{L_{\min}}{L} + 0.1 \times P_{\text{time}} + 0.1 \times P_{\text{smooth}}, \tag{21}$$

where $L_{\min}$ denotes the minimum average path length among all compared algorithms in the same experimental setting.

The weights in Equation (21) are selected according to the engineering priority of static warehouse path planning. Feasibility and path quality are given dominant weights because a planner must first generate a valid and globally efficient route before secondary indicators become meaningful. Planning time and smoothness are included with smaller weights to reflect real-time applicability and motion friendliness, while preventing them from overwhelming the primary path-quality objective. Therefore, the score is used as a compact descriptive index for cross-method comparison rather than as the sole criterion for declaring algorithmic superiority.

5. Experimental Results and Analysis

5.1. Experimental Setup

5.1.1. Computing Environment

All experiments were conducted on a platform running Windows 11. The algorithms were implemented and evaluated in Python 3.8 using PyCharm 2024.2.1, and the hardware configuration consisted of an Intel(R) Core(TM) i9-14900HX processor at 2.20 GHz with 32 GB of memory. To ensure strict fairness across methods, all maps were stored as raster images and NumPy binary files, so that each algorithm was tested under exactly the same environmental inputs and start–goal tasks.

5.1.2. Baseline Algorithms

To comprehensively assess the performance of LE-HG-PRM, six representative baseline methods are selected for comparison: A*, JPS, PRM, RRT, RRT*, and HG-PRM. A* is included as a classical heuristic graph-search algorithm and provides a deterministic reference for shortest-path search. JPS is adopted as an efficient grid-based acceleration strategy. PRM serves as the standard sampling-based roadmap baseline without structural priors. RRT and RRT* represent rapidly exploring random sampling and asymptotically optimal tree expansion, respectively. HG-PRM is included as the direct predecessor of the proposed method, enabling a clearer examination of the actual gain introduced by LE-HG-PRM. For fairness, all non-deterministic comparison methods were tuned through grid search over their key parameters, including sampling density, neighbor number, connection radius or step size, and iteration budget where applicable. The final baseline settings were selected from the best-performing stable configurations under the same maps and task sets, so the reported comparisons reflect reasonably tuned baselines rather than default or under-optimized implementations.

5.2. Experiment I: Static-Complexity Progression

Figure 4 shows the environment configurations, representative path results, and cross-scenario statistical comparisons for the three static-complexity scenarios. As can be seen from Tables 1–3 and Figure 4c(i)–c(iii), as the environment gradually evolves from a simple regular warehouse to a high-density complex scene, the average path length of all algorithms increases overall. This trend is consistent with the progressive shrinkage of feasible regions and the increasing difficulty of obstacle avoidance.

Table 1. Performance comparison in the simple regular warehouse environment.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	542.0	0.876	0.988	0.7463
JPS	100	400.2	0.425	0.295	0.9666
PRM	100	413.9	0.111	0.328	0.9559
RRT	100	507.6	0.041	0.579	0.8334
RRT*	100	455.5	0.290	0.354	0.8974
HG-PRM	100	394.5	0.139	0.629	0.9349
LE-HG-PRM	100	390.5	0.258	0.428	0.9559

Table 2. Performance comparison in the medium-complexity warehouse environment.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	542.0	0.690	1.166	0.7525
JPS	100	392.0	0.408	0.327	0.9758
PRM	100	414.2	0.111	0.362	0.9545
RRT	100	479.5	0.017	0.574	0.8619
RRT*	100	423.5	0.258	0.341	0.9423
HG-PRM	100	395.4	0.141	0.454	0.9571
LE-HG-PRM	100	389.1	0.265	0.331	0.9855

Table 3. Performance comparison in the high-density complex warehouse environment.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	688.0	1.258	0.840	0.7584
JPS	100	519.9	0.401	0.479	0.9563
PRM	100	553.8	0.114	0.366	0.9637
RRT	100	638.4	0.032	0.534	0.8740
RRT*	100	587.1	0.297	0.368	0.9274
HG-PRM	100	534.0	0.142	0.622	0.9385
LE-HG-PRM	100	521.8	0.269	0.504	0.9573

However, LE-HG-PRM remains among the best-performing methods in terms of average path length across all three scenarios, with values of 390.5, 389.1, and 521.8 pixels, respectively. It achieves the shortest mean path in the simple and medium-complexity environments and remains statistically comparable to JPS in the high-density complex environment, while still outperforming HG-PRM in mean path length. This indicates that its structure-aware sampling strategy can preserve a competitive path-quality level as complexity increases, especially compared with the direct predecessor HG-PRM. In particular, in the medium-complexity scenario, LE-HG-PRM achieves the highest comprehensive score of 0.9855 while simultaneously producing the shortest path. This suggests that when regular corridors, heterogeneous obstacles, and local bottlenecks coexist, the geometric prior of the environment can be effectively converted into observable overall benefit.

By contrast, JPS still benefits from regular grid structures in simple scenarios and therefore attains a relatively high score there. In the high-density complex environment, JPS obtains a slightly shorter mean path, whereas LE-HG-PRM remains close to the best result while producing a higher overall score than HG-PRM. This observation suggests that superiority in one metric does not necessarily imply dominance in every evaluation dimension.

Representative path results further illustrate this behavior. As shown in Figure 4b(i), in the simple regular scenario, the path generated by LE-HG-PRM is most closely aligned with the dominant traffic direction and exhibits the least redundant redirection. As shown in Figure 4b(ii), in the medium-complexity scenario, the path still preserves a clear global skeleton and does not display obvious oscillation caused by local obstacle perturbations. Even in the fragmented and strongly constrained environment of Figure 4b(iii), LE-HG-PRM maintains a relatively clear global travel direction.

Figure 5 further reveals the distribution-level differences behind these mean values. As shown in Figure 5a(i),b(i),c(i), the path-length distribution of LE-HG-PRM is generally shifted to the left across all three scenarios and is more concentrated than those of RRT, RRT*, and standard PRM. This indicates that its advantage does not arise from a few favorable cases, but instead reflects a broader improvement over most tasks. As shown in Figure 5a(ii),b(ii),c(ii), the planning-time distribution of LE-HG-PRM is always substantially better than that of A*, while remaining in a relatively stable medium-to-low latency range rather than exhibiting the extreme leftward shift seen in RRT. This suggests that the proposed method adopts a deliberate trade-off in which moderate roadmap-construction overhead is exchanged for stable path quality. In addition, as shown in Figure 5a(iii),b(iii), the path-smoothness distribution of LE-HG-PRM is visibly left-shifted and contracted relative to HG-PRM in the simple and medium-complexity scenarios, indicating that the proposed improvement enhances not only path length but also path regularity. Although its smoothness advantage is less absolute in the complex scenario of Figure 5c(iii), it still remains in the leading group.

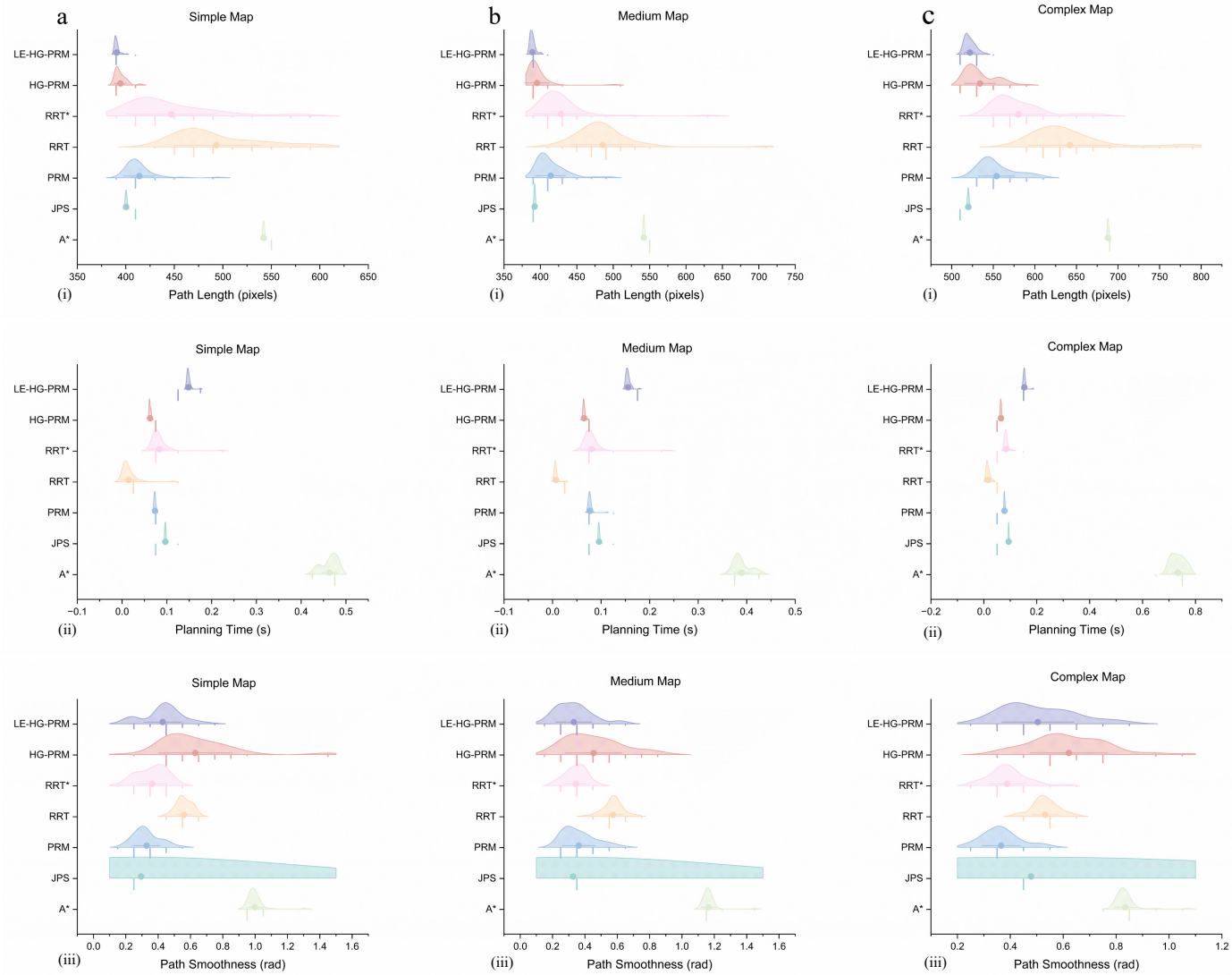


Figure 5. Cloud–rain distribution comparison of algorithmic performance under different static-complexity scenarios. (**a(i)–a(iii)**) Path length, planning time, and path smoothness distributions in the simple regular warehouse environment. (**b(i)–b(iii)**) Corresponding distributions in the medium-complexity environment. (**c(i)–c(iii)**) Corresponding distributions in the high-density complex environment. Different colors denote different algorithms as indicated by the legend.

Overall, Experiment I shows that the core competitiveness of LE-HG-PRM does not lie merely in a shorter mean path length, but in its ability to continuously generate more stable, more regular, and more globally structured paths under progressively increasing environmental complexity.

5.3. Experiment II: Corridor-Width Sensitivity

Figure 6 shows the environment configurations, representative planning results, and cross-scenario statistics under different corridor-width conditions. According to Tables 4–6 and Figure 6c(i)–c(iii), as the corridor width progressively decreases from a wide main corridor to a standard storage aisle and then to a narrow picking aisle, the average path length of all algorithms decreases overall. This is because the geometric feasible region is increasingly compressed, reducing path freedom and causing the optimal solution to converge toward the corridor medial axis.

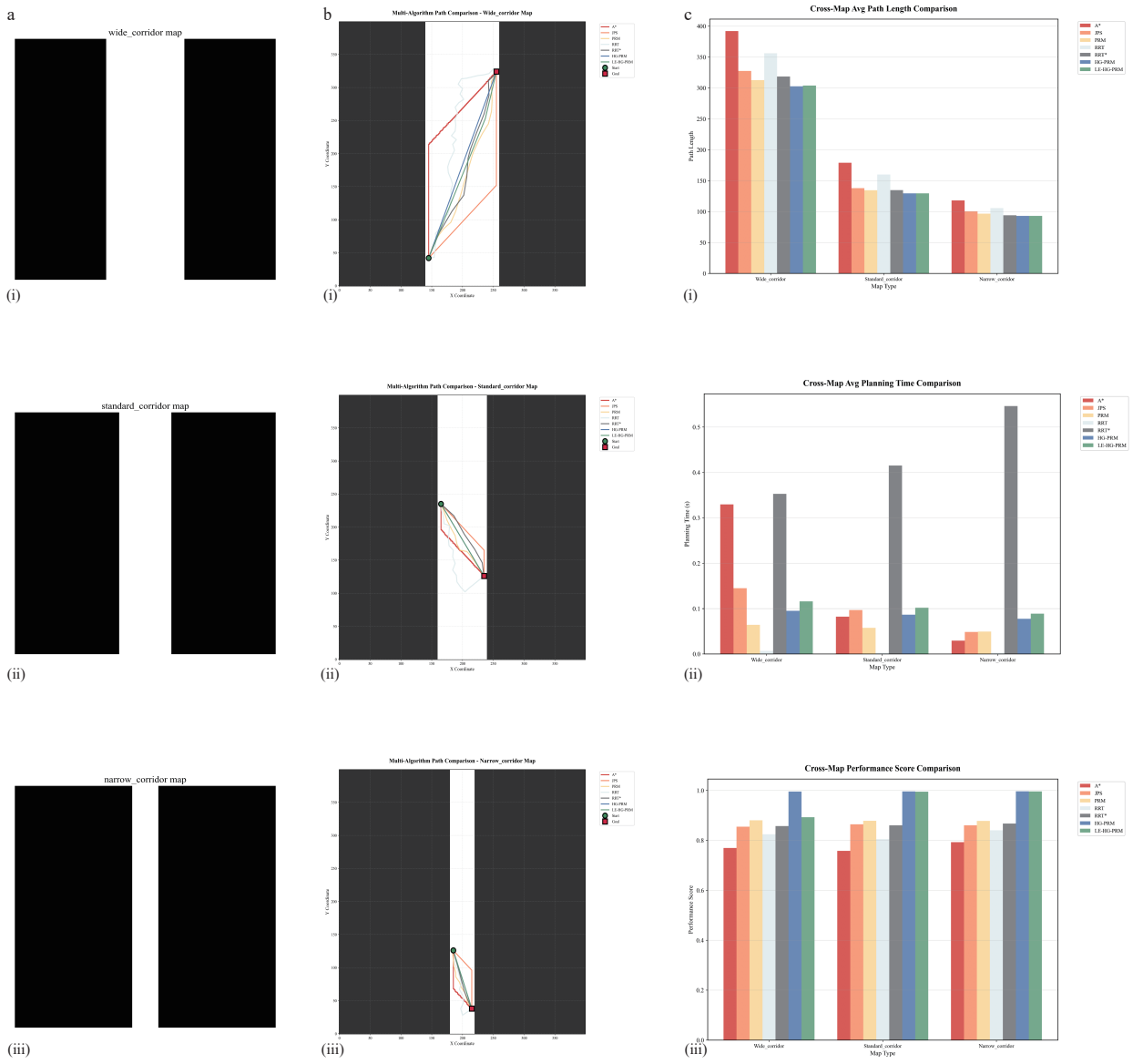


Figure 6. Experimental environments, representative paths, and cross-scenario statistics under different corridor-width settings. (**a(i)–a(iii)**) Wide main corridor, standard storage corridor, and narrow picking corridor. (**b(i)–b(iii)**) Representative path-planning results in the corresponding corridor environments. (**c(i)–c(iii)**) Cross-scenario comparisons of average path length, average planning time, and overall performance score.

Table 4. Performance comparison in the wide main-corridor environment.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	291.1	0.197	0.757	0.6967
JPS	40.00	178.5	0.090	0.785	0.7155
PRM	100	229.6	0.061	0.290	0.7857
RRT	100	267.6	0.005	0.494	0.7333
RRT*	100	250.5	0.869	0.308	0.7128
HG-PRM	100	221.0	0.091	0.000	0.7993
LE-HG-PRM	100	221.7	0.114	0.199	0.7969

Table 5. Performance comparison in the standard storage-corridor environment.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	179.0	0.082	0.926	0.7577
JPS	100	138.0	0.097	0.785	0.8645
PRM	100	134.5	0.057	0.294	0.8786
RRT	100	159.9	0.003	0.498	0.8049
RRT*	100	134.7	0.415	0.267	0.8601
HG-PRM	100	129.5	0.087	0.000	0.9957
LE-HG-PRM	100	129.5	0.102	0.000	0.9949

Table 6. Performance comparison in the narrow picking-corridor environment.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	118.0	0.029	0.611	0.7925
JPS	100	100.4	0.048	0.785	0.8605
PRM	100	96.7	0.049	0.291	0.8782
RRT	100	105.6	0.002	0.459	0.8400
RRT*	100	94.0	0.546	0.184	0.8675
HG-PRM	100	93.0	0.077	0.000	0.9961
LE-HG-PRM	100	93.0	0.089	0.000	0.9956

Within this process, HG-PRM and LE-HG-PRM remain consistently in the optimal or near-optimal group. In the wide-corridor scenario, HG-PRM attains the shortest average path of 221.0 pixels, followed closely by LE-HG-PRM at 221.7 pixels. In the standard and narrow corridor scenarios, LE-HG-PRM shares the shortest average path lengths with HG-PRM, at 129.5 and 93.0 pixels, respectively, while also attaining an average path smoothness of 0.000 rad in both cases. These results indicate that the proposed method can stably approximate the geometrically ideal centerline-like solution in highly constrained corridor environments.

By contrast, although A* and JPS retain certain advantages in a regular deterministic search, these advantages do not translate into sustained leadership in path quality under corridor scenarios. RRT preserves very low planning time, but this comes at the expense of longer paths and lower regularity. RRT* exhibits a more pronounced increase in planning time after corridor narrowing.

The representative planning results further support these observations. As shown in Figure 6b(i), LE-HG-PRM already maintains good conformity to the dominant travel direction in the wide-corridor environment. As shown in Figure 6b(ii),b(iii), its path unfolds almost along the centerline in both the standard and narrow corridors, with little obvious lateral offset or unnecessary oscillation.

Figure 7 further reveals the stability of this behavior at the distribution level. As shown in Figure 7a(i),b(i),c(i), as the corridor becomes narrower, the path-length distribution of LE-HG-PRM not only remains on the left side, but also contracts in width, indicating that its advantage extends from a shorter mean path to a more stable improvement over most tasks. As shown in Figure 7a(ii),b(ii),c(ii), the planning-time distribution of LE-HG-PRM remains relatively compact across all corridor-width scenarios and does not display the obvious dispersion observed for RRT* in highly constrained cases. Moreover, as shown in

Figure 7b(iii),c(iii), the path-smoothness distribution of LE-HG-PRM is almost compressed into an extremely narrow band in the standard and narrow corridor scenarios, implying that the method can consistently output an approximately medial-axis trajectory rather than doing so only occasionally.

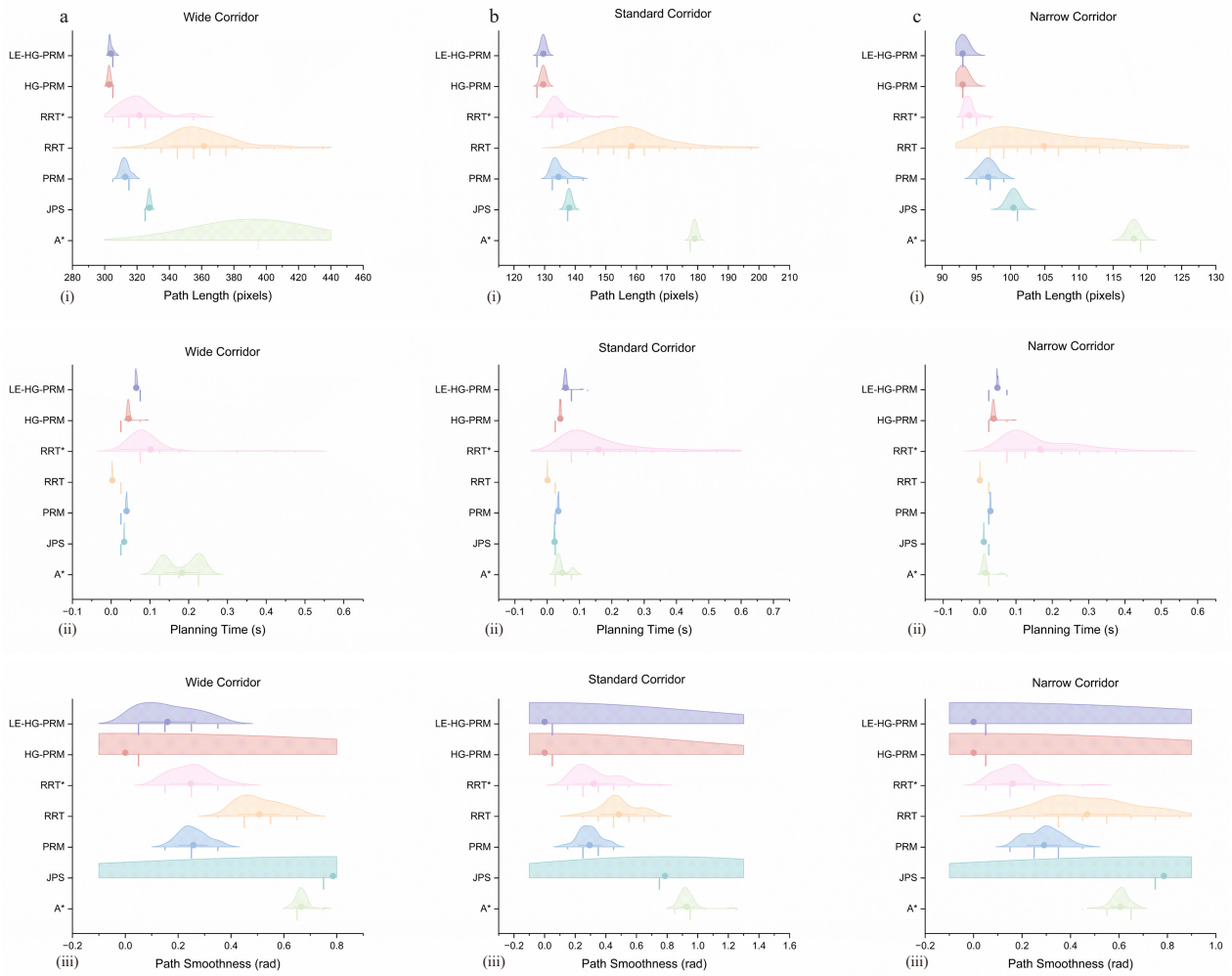


Figure 7. Cloud–rain distribution comparison of algorithmic performance under different corridor-width scenarios. (**a(i)–a(iii)**) Path length, planning time, and path smoothness distributions in the wide main-corridor environment. (**b(i)–b(iii)**) Corresponding distributions in the standard storage-corridor environment. (**c(i)–c(iii)**) Corresponding distributions in the narrow picking-corridor environment. Different colors denote different algorithms as indicated by the legend.

Therefore, Experiment II indicates that LE-HG-PRM preserves near-optimal path quality and high output consistency under corridor contraction. Its advantage is most evident when compared with external baselines, while its performance remains comparable to HG-PRM in highly constrained corridor settings.

5.4. Experiment III: Parameter-Sensitivity Analysis

As shown in Figure 8, a fixed test scenario is adopted for parameter-sensitivity analysis so that environmental variation does not interfere with the observed trends. Figure 9 presents the mean-value changes under different parameter settings, and Figure 10 further shows the corresponding distribution-level behavior.

First, the number of sampling points mainly affects the local geometric quality of the path, while its effect on the global path skeleton is relatively limited. As reported in Table 7 and shown in Figure 9b(i), when the number of sampling points increases from 200 to 1000,

the average path length decreases only slightly, from 353.0 to 352.0 pixels, indicating a limited influence on global path efficiency. By contrast, as shown in Figure 9b(ii), the average planning time increases continuously from 0.154 s to 0.267 s. Meanwhile, the overall performance score in Figure 9b(iii) shows a gradual upward trend, suggesting that the gain comes less from path shortening and more from improvements in local path quality and regularity.

This interpretation is further supported by the distributions in Figure 10. As shown in Figure 10a(i), the right tail of the path-length distribution is more pronounced when the number of sampling points is low, indicating that insufficient coverage can lead to obvious detours in individual tasks. Once the sampling number reaches approximately 400 or above, the path-length distributions begin to overlap substantially, indicating that the marginal gain in global path length becomes small. At the same time, Figure 10c(i) shows that the smoothness distribution continues to shift leftward and contract, suggesting that denser sampling mainly improves local geometric resolution and hence path regularity.

Second, the number of neighbor connections $k_{\text{neighbors}}$ directly affects the topological quality of the roadmap and therefore has a stronger influence on connectivity and path length. As reported in Table 8 and shown in Figure 9a(i), when the number of neighbors increases from 4 to 12, the average path length decreases from 360.7 to 351.3 pixels, indicating that higher local connectivity substantially improves graph quality. However, Figure 9a(ii) shows that the average planning time also increases, from 0.151 s to 0.219 s, implying that this topological improvement is obtained at the cost of additional computation. In Figure 9a(iii), the overall score rises as the number of neighbors increases, but the gain becomes much smaller after 8.

The corresponding distributions in Figure 10 show the same pattern. When the number of neighbor connections is small, the path-length distribution is wider and has a longer right tail, indicating that insufficient connectivity amplifies task-to-task variation. Once the number of neighbors reaches 8 or 10, the distribution becomes noticeably more concentrated. At the same time, the planning-time distribution in Figure 10b(ii) shifts steadily to the right as the number of neighbors increases. Therefore, the effect of neighbor number follows a typical pattern of rapid initial improvement followed by diminishing returns.

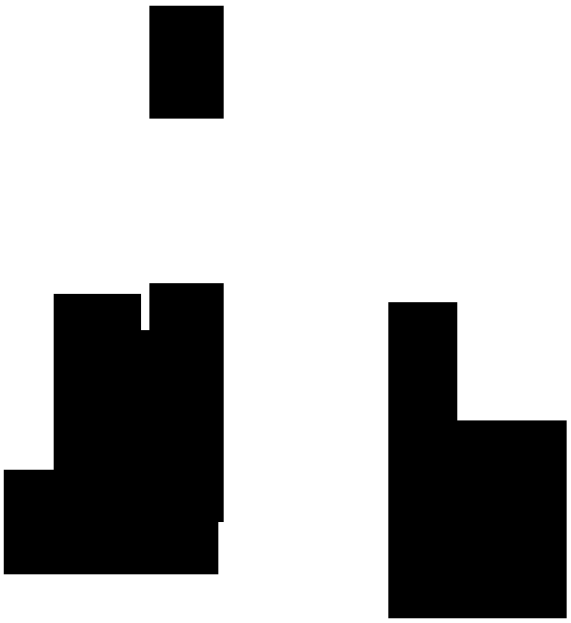


Figure 8. Fixed test scenario used for the parameter-sensitivity experiment. The same map is adopted throughout Experiment III to isolate the effects of sampling density, neighbor connection number, and sampling strategy from environmental variability.

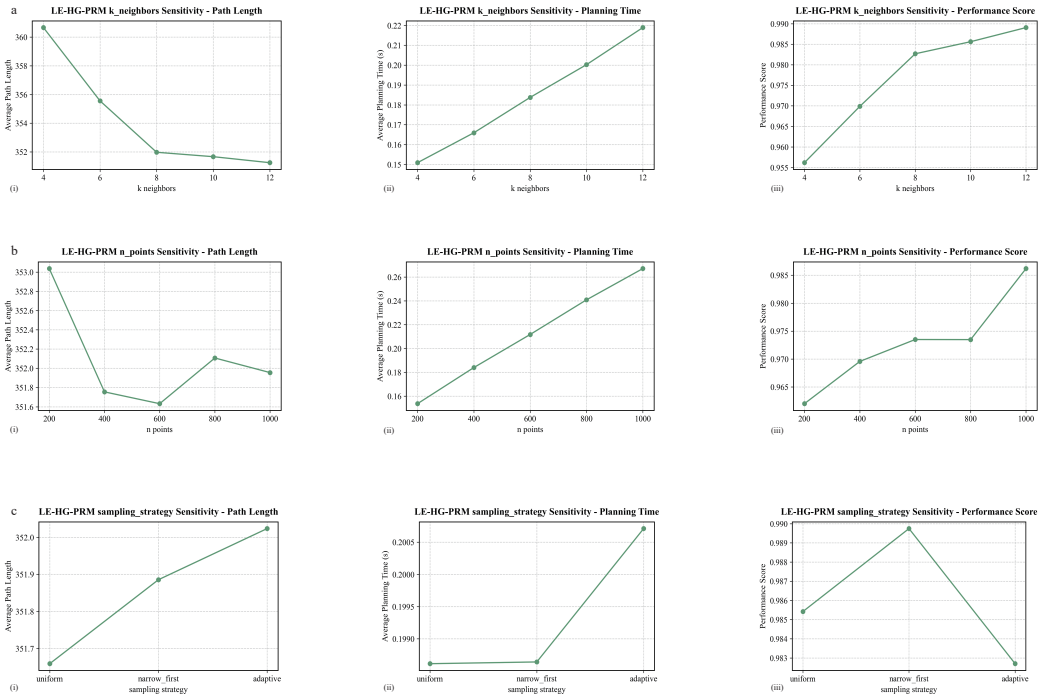


Figure 9. Mean-value trend analysis for LE-HG-PRM parameter sensitivity. (**a(i)–a(iii)**) Effects of the number of neighbor connections on average path length, average planning time, and overall score. (**b(i)–b(iii)**) Effects of the number of sampling points on the same metrics. (**c(i)–c(iii)**) Effects of different sampling strategies on the same metrics.

Table 7. Effect of the number of sampling points on the performance of LE-HG-PRM.

Number of Sampling Points	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
200	100	353.0	0.154	0.384	0.9620
400	100	351.8	0.184	0.349	0.9696
600	100	351.6	0.212	0.327	0.9735
800	100	352.1	0.241	0.319	0.9735
1000	100	352.0	0.267	0.275	0.9862

Table 8. Effect of the number of neighbor connections on the performance of LE-HG-PRM.

Number of Neighbor Connections	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
4	100	360.7	0.151	0.440	0.9562
6	100	355.6	0.166	0.401	0.9699
8	100	352.0	0.184	0.363	0.9827
10	100	351.7	0.200	0.351	0.9856
12	100	351.3	0.219	0.338	0.9891

Third, the sampling strategy determines how the limited sampling budget is allocated across different spatial regions. As shown in Figure 9c(i), the differences in average path length among the `uniform`, `narrow_first`, and `adaptive` strategies are not large, indicating that sampling strategy does not fundamentally rewrite the global feasible-path skeleton. However, differences emerge more clearly in planning time and overall score, as shown in Figure 9c(ii),c(iii). The distributions in Figure 10a(iii),c(iii) show that `narrow_first` yields more concentrated path-length and smoothness distributions, whereas `adaptive` displays a more right-shifted time distribution. This suggests that, in the current warehouse scenarios characterized by local bottlenecks and narrow corridors, prioritizing the sampling budget toward critical constrained regions is more effective than globally uniform allocation or broader adaptive diffusion.

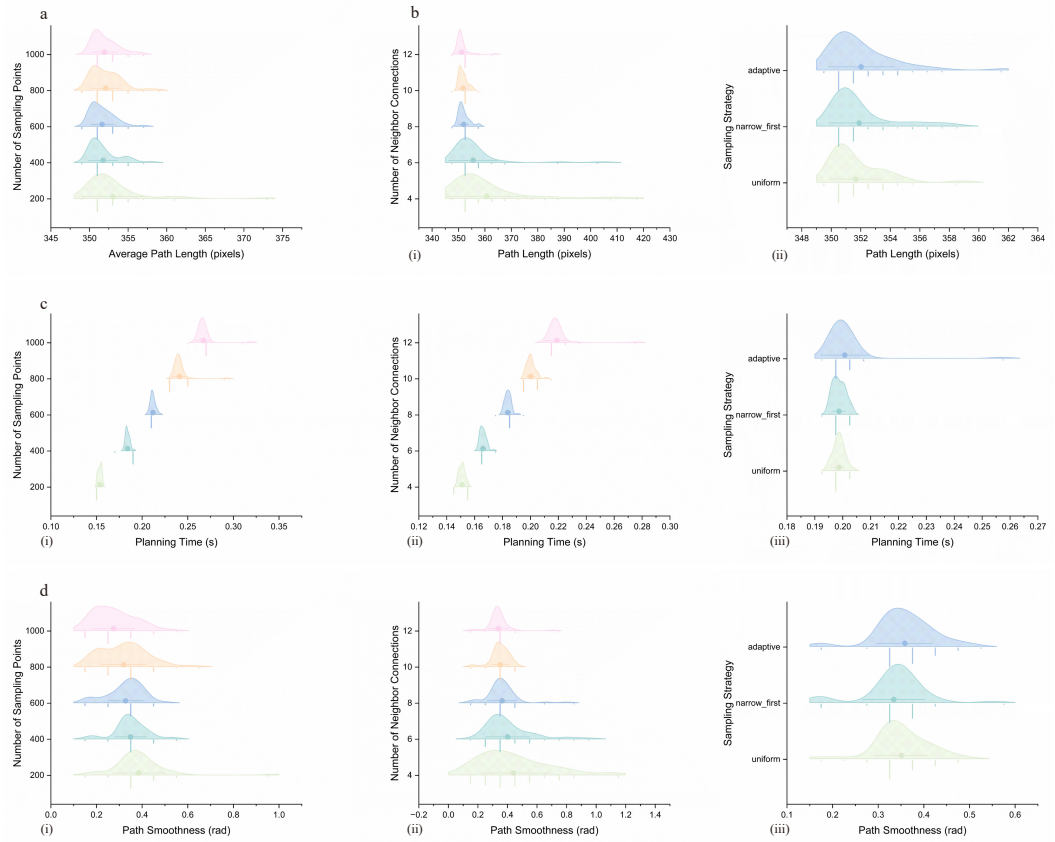


Figure 10. Distribution-level analysis of LE-HG-PRM parameter sensitivity. (**a(i)–a(iii)**) Path-length distributions under different sampling-point numbers, neighbor-connection numbers, and sampling strategies. (**b(i)–b(iii)**) Corresponding planning-time distributions. (**c(i)–c(iii)**) Corresponding path-smoothness distributions. Different colors denote different parameter settings or sampling strategies as indicated by the legend.

In addition to the three explicit parameter groups, the threshold definitions in Equation (6) were checked by perturbing the scale-normalized thresholds around their default values. The purpose of this check was not to identify a new optimal pair of τ_n and τ_w, but to examine whether the roadmap construction collapses when the constrained-region and broad-region boundaries are moderately shifted. The observed trends remained consistent under moderate perturbations: success rate was preserved, and the relative advantage of structure-aware sampling over uniform sampling remained stable. This supports the use of the scale-dependent formulas in Equation (6) as reproducible engineering defaults.

Taken together, Experiment III shows that LE-HG-PRM remains stable and usable over a relatively wide parameter range and does not exhibit severe performance degradation under mild parameter perturbations. Different parameters affect different layers of the algorithm: the number of sampling points mainly influences local geometric quality, the number of neighbor connections primarily affects graph connectivity, and the sampling strategy determines how efficiently the limited budget is allocated to critical regions. Therefore, parameter tuning should not seek only an isolated optimum, but rather an effective operating interval that balances path length, smoothness, and computational cost. According to the present sensitivity trends, $n_{\text{points}} = 400$–600, $k_{\text{neighbors}} = 8$–10, and the `narrow_first` strategy provide a practical default configuration from an engineering perspective, although they should be regarded as a recommended operating range rather than a statistically unique optimum.

5.5. Experiment IV: Scale-Expansion Analysis

Figure 11 presents the environment configurations, representative planning results, and cross-scale statistical comparisons under different map sizes. According to Tables 9–11 and Figure 11a(iii),b(iii),c(iii), LE-HG-PRM achieves the shortest average path in the 200 × 200 scenario, with a value of 114.5 pixels. In the 400 × 400 scenario, it remains shorter than PRM, RRT, RRT*, HG-PRM, and A*, although JPS obtains a shorter mean path. In the 600 × 600 scenario, LE-HG-PRM continues to outperform PRM, HG-PRM, RRT, RRT*, and A* in path length, while JPS remains the shortest method. This indicates that, as map scale increases, the path-quality performance of LE-HG-PRM does not collapse abruptly, but its advantage becomes more scenario-dependent and is no longer dominant over all deterministic search baselines.

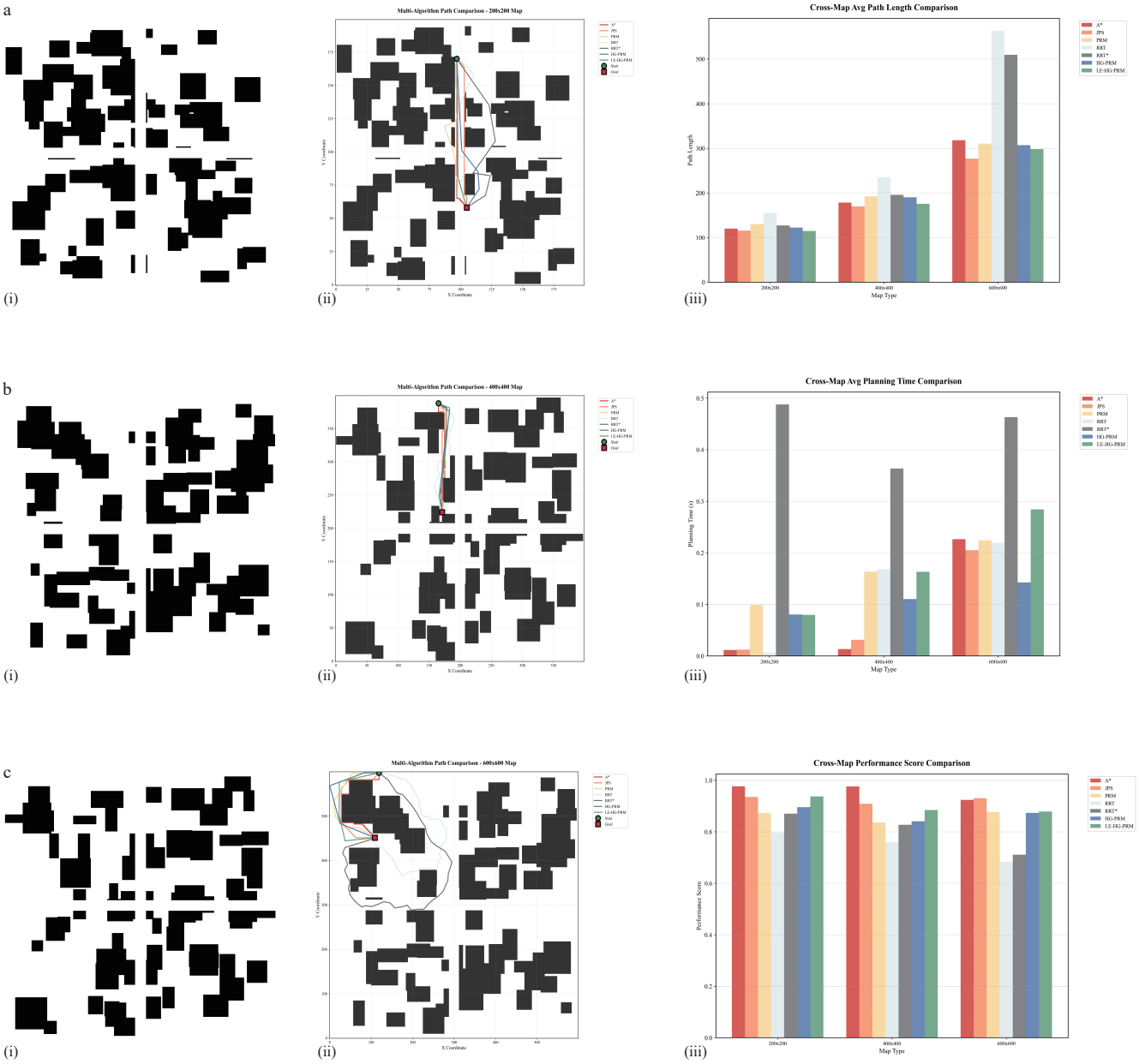


Figure 11. Experimental environments, representative paths, and cross-scale statistics under different map sizes. (**a(i)–a(iii)**) Environment, representative planning result, and average path-length comparison for the 200 × 200 map. (**b(i)–b(iii)**) Corresponding results for the 400 × 400 map. (**c(i)–c(iii)**) Corresponding results for the 600 × 600 map, including the cross-scale overall performance-score comparison.

Table 9. Performance comparison in the small-scale map.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	120.0	0.012	0.155	0.9765
JPS	100	115.3	0.013	0.393	0.9354
PRM	100	130.2	0.099	0.414	0.8723
RRT	100	154.4	0.006	0.581	0.7971
RRT*	100	127.0	0.488	0.355	0.8703
HG-PRM	100	122.2	0.080	0.504	0.8953
LE-HG-PRM	100	114.5	0.080	0.379	0.9370

Table 10. Performance comparison in the medium-scale map.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	178.0	0.013	0.048	0.9763
JPS	100	169.8	0.031	0.471	0.9087
PRM	98	192.3	0.164	0.530	0.8365
RRT	100	235.4	0.169	0.623	0.7600
RRT*	100	195.2	0.363	0.467	0.8270
HG-PRM	98	190.1	0.111	0.823	0.8409
LE-HG-PRM	100	175.1	0.163	0.595	0.8848

Table 11. Performance comparison in the large-scale map.

Algorithm	Success Rate (%)	Path Length (Pixels)	Planning Time (s)	Smoothness (rad)	Score
A*	100	318.0	0.226	0.266	0.9242
JPS	100	277.0	0.205	0.654	0.9304
PRM	100	310.3	0.224	0.635	0.8771
RRT	100	563.4	0.220	0.546	0.6835
RRT*	100	509.8	0.463	0.425	0.7111
HG-PRM	100	306.8	0.143	0.924	0.8731
LE-HG-PRM	100	298.1	0.284	0.938	0.8788

At the same time, the ordering of overall performance scores changes noticeably across scales. In small-scale scenarios, A* attains the highest score because of its extremely low time cost and relatively favorable smoothness. In the largest-scale scenario, JPS obtains the shortest path and the highest overall score. LE-HG-PRM is therefore not dominant in every metric at every scale, but its path length remains consistently among the best-performing methods. In this sense, Experiment IV reveals what kind of advantage each algorithm can still preserve after scale expansion.

Figure 12 further explains the source of this rearrangement. As shown in Figure 12a(i), the path-length distribution of LE-HG-PRM is already located to the left in the 200 × 200 scenario. This leftward advantage remains in the 400 × 400 scenario, as shown in Figure 12a(ii). When the scale expands to 600 × 600, the distribution shifts rightward overall, but it does not rapidly develop the broader and heavier right tail typically associated with generic random-sampling methods. This indicates that environment-aware priors continue to exert a persistent constraining effect even in larger workspaces.

Meanwhile, Figure 12b(i),b(ii) show that A* and JPS retain more compact and left-shifted planning-time distributions at small and medium scales, which helps explain their advantage in overall score under those conditions. However, as shown in Figure 12b(iii), although the time distribution of LE-HG-PRM expands moderately with increasing map size, it does not exhibit uncontrolled growth. In addition, Figure 12c(i),c(ii) show that the path smoothness of LE-HG-PRM remains in the first echelon in small- and medium-scale scenarios. In the large-scale environment shown in Figure 12c(iii), the smoothness distribution becomes more dispersed, suggesting that while environment geometry can still help maintain a good global path skeleton, preserving highly regular local path shape at a very large scale requires additional construction and post-optimization effort.

Therefore, Experiment IV shows that the value of LE-HG-PRM does not lie in dominating all metrics at all scales, but in the continuity of its path-quality advantage and in its relatively smooth, interpretable, and controllable degradation behavior as the planning scale increases.

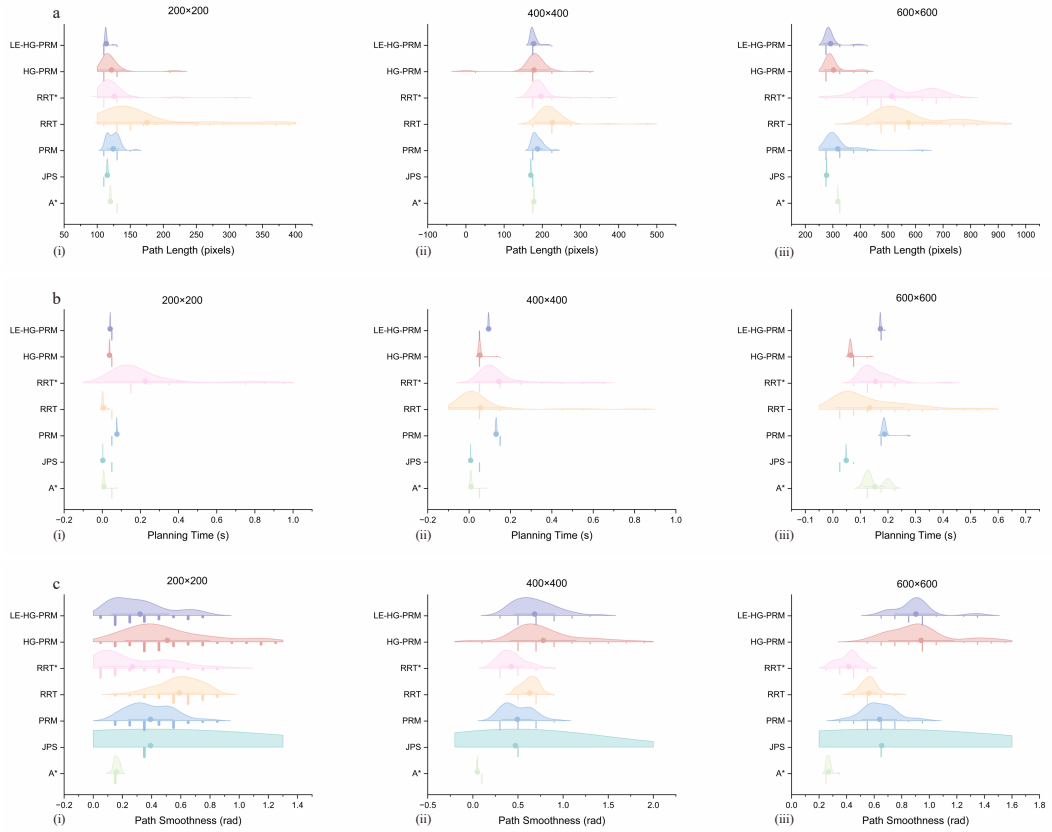


Figure 12. Cloud–rain distribution comparison of algorithmic performance at different map scales. (**a(i)–a(iii)**) Path-length distributions in the 200 × 200, 400 × 400, and 600 × 600 maps. (**b(i)–b(iii)**) Corresponding planning-time distributions. (**c(i)–c(iii)**) Corresponding path-smoothness distributions. Different colors denote different algorithms as indicated by the legend.

5.6. Statistical Significance Analysis

To further support the mean-value and distribution-level comparisons reported above, a statistical significance analysis was conducted for the key path-quality results. Since all compared algorithms were evaluated under the same scenarios, map conditions, and repeated planning trials, their path-length results could be matched on a paired basis. Therefore, a two-sided paired t-test was used to examine whether the observed path-length differences between LE-HG-PRM and representative comparison methods were statistically significant. It should be noted that the paired t-test and Holm–Bonferroni correction play different roles in the analysis: the paired t-test evaluates the significance of each individual algorithm comparison, whereas the Holm–Bonferroni procedure controls the false-positive risk caused by multiple comparisons.

5.6.1. Statistical Testing Protocol

For each scenario, paired observations were constructed by matching the path-length records of LE-HG-PRM and a comparison algorithm under the same scenario and repeated-run condition. Since path length is meaningful only when a collision-free path is successfully generated, only valid paired records in which both algorithms successfully produced a feasible path were retained for the path-length test. For sampling-based methods, the 12 independent repeated runs associated with the same start–goal task were first averaged

at the task level, and the paired statistical test was then performed over the task-level paired observations. This prevents artificial inflation of the sample size by treating repeated random seeds of the same task as independent tasks. Let $L_{i,\mathrm{LE}}$ denote the path length obtained by LE-HG-PRM in the i-th valid paired trial, and let $L_{i,\mathrm{B}}$ denote the corresponding path length obtained by a comparison baseline B. The paired difference was defined as

$$d_i = L_{i,\mathrm{LE}} - L_{i,\mathrm{B}}, \tag{22}$$

where a negative value of d_i indicates a shorter path generated by LE-HG-PRM, whereas a positive value indicates a shorter path generated by the baseline. The statistical hypotheses were formulated as

$$H_0 : \mu_d = 0, \qquad H_1 : \mu_d \neq 0, \tag{23}$$

where μ_d is the population mean of the paired differences. The null hypothesis H_0 assumes that there is no statistically significant path-length difference between the two methods, while the alternative hypothesis H_1 indicates that a significant difference exists. A two-sided test was adopted rather than a one-sided test so that both possible directions of difference could be evaluated without assuming in advance that LE-HG-PRM must outperform the baseline in every scenario. The paired t-test statistic was calculated as

$$t = \frac{\bar{d}}{s_d / \sqrt{n}}, \tag{24}$$

where $\bar{d}$ and s_d are the mean and standard deviation of the paired differences, respectively, and n is the number of valid paired samples. The corresponding 95% confidence interval of the mean paired difference is estimated as $\bar{d} \pm t_{0.975,n-1} s_d / \sqrt{n}$ and is used together with the adjusted p-value to interpret the robustness of the observed difference.

5.6.2. Tested Metric and Comparison Groups

The statistical analysis in the main text focuses on path length. Path length is the primary indicator of global path quality because it directly reflects whether a planner can generate a shorter collision-free route under the same environment and task conditions. Planning time, path smoothness, success rate, and overall score have already been systematically reported in the preceding performance tables, and path length is also a dominant component of the overall score. Therefore, to avoid an over-expanded main-text statistical table, Table 12 reports only the core significance results for path length rather than repeating significance tests for all metrics.

Table 12. Statistical significance analysis of key path-length comparisons.

Experiment	Scenario	Vs HG-PRM: ΔL, p_{adj}	Strongest External Baseline	Vs Strongest Baseline: ΔL, p_{adj}
Exp.I	Simple regular warehouse	-3.9, $p_{\mathrm{adj}} < 0.001$ ***	JPS	-9.7, $p_{\mathrm{adj}} < 0.001$ ***
Exp.I	Medium-complexity warehouse	-6.3, $p_{\mathrm{adj}} = 0.088$ n.s.	JPS	-3.0, $p_{\mathrm{adj}} < 0.001$ ***
Exp.I	High-density complex warehouse	-12.2, $p_{\mathrm{adj}} = 0.001$ **	JPS	$+1.9$, $p_{\mathrm{adj}} = 0.159$ n.s.
Exp.II	Wide main corridor	$+1.2$, $p_{\mathrm{adj}} < 0.001$ ***	PRM	-8.8, $p_{\mathrm{adj}} < 0.001$ ***
Exp.II	Standard storage corridor	0.0, $p_{\mathrm{adj}} = 1.000$ n.s.	PRM	-5.0, $p_{\mathrm{adj}} < 0.001$ ***
Exp.II	Narrow picking corridor	0.0, $p_{\mathrm{adj}} = 1.000$ n.s.	RRT*	-1.0, $p_{\mathrm{adj}} < 0.001$ ***
Exp.IV	200×200 map	-7.9, $p_{\mathrm{adj}} = 0.083$ n.s.	JPS	-1.6, $p_{\mathrm{adj}} = 0.005$ **
Exp.IV	400×400 map	-8.4, $p_{\mathrm{adj}} = 0.088$ n.s.	JPS	$+7.5$, $p_{\mathrm{adj}} < 0.001$ ***
Exp.IV	600×600 map	-10.8, $p_{\mathrm{adj}} = 0.331$ n.s.	JPS	$+14.6$, $p_{\mathrm{adj}} = 0.003$ **

Note: ΔL denotes the mean path-length difference calculated as $\overline{L}_{\mathrm{LE-HG-PRM}} - \overline{L}_{\mathrm{baseline}}$. Negative values indicate shorter paths for LE-HG-PRM, and positive values indicate shorter paths for the comparison baseline. p_{adj} denotes Holm–Bonferroni-adjusted p-values obtained from two-sided paired t-tests. Significance levels are marked as n.s. for not significant, ** for $p_{\mathrm{adj}} < 0.01$, and *** for $p_{\mathrm{adj}} < 0.001$. The paired sample size is $n = 50$ for all comparisons except the 400×400 map comparison between LE-HG-PRM and HG-PRM, where $n = 48$.

For each scenario, two types of comparisons were pre-defined. The first comparison was between LE-HG-PRM and HG-PRM. Since HG-PRM is the direct predecessor of the proposed method, this comparison evaluates whether the logistics-enhanced sampling, dynamic-neighbor connection, and progressive optimization mechanisms provide statistically verifiable path-length improvement over the previous method. The second comparison was between LE-HG-PRM and the strongest external baseline in the corresponding scenario. The external baseline was selected from A*, JPS, PRM, RRT, and RRT* according to the shortest mean path length among methods with reliable task success in that scenario. This avoids treating a method with a substantially lower success rate as the strongest baseline merely because its successful subset has a short mean path. This comparison examines whether LE-HG-PRM remains competitive not only against its direct predecessor but also against the best-performing representative external method.

Statistical testing was applied to Experiment I, Experiment II, and Experiment IV, corresponding to static-complexity progression, corridor-width sensitivity, and scale expansion, respectively. Experiment III was not included in Table 12 because it investigates the internal parameter sensitivity of LE-HG-PRM rather than direct algorithm-to-algorithm performance comparisons.

5.6.3. Multiple-Comparison Correction and Reporting Format

Because Table 12 reports multiple scenarios and multiple algorithm comparisons, judging significance directly from raw p-values may increase the risk of false-positive conclusions. Therefore, Holm–Bonferroni correction was applied to the p-values obtained from paired t-tests. In this procedure, the paired t-test provides the raw significance level for each individual comparison, while Holm–Bonferroni correction adjusts these p-values to make the final significance judgment more robust.

In Table 12, the mean path-length difference is defined as

$$\Delta L = \overline{L}_{\mathrm{LE}} - \overline{L}_{\mathrm{B}}, \tag{25}$$

where $\overline{L}_{\mathrm{LE}}$ and $\overline{L}_{\mathrm{B}}$ are the mean path lengths of LE-HG-PRM and the comparison algorithm, respectively. Consistent with the path-length format used in the preceding performance tables, ΔL is reported in pixels to one decimal place. Because ΔL is calculated from the valid paired records retained for each statistical test, it may differ slightly from the direct subtraction of rounded aggregate means in the preceding performance tables. The adjusted p-value is denoted as p_{adj} and is reported to three decimal places unless $p_{\mathrm{adj}} < 0.001$, in which case it is reported as $p_{\mathrm{adj}} < 0.001$. Statistical significance markers are assigned according to adjusted p-values rather than raw p-values.

5.6.4. Interpretation of Statistical Results

The results in Table 12 show that the path-length advantage of LE-HG-PRM is scenario-dependent. In Experiment I, LE-HG-PRM significantly reduces path length compared with HG-PRM in the simple regular warehouse and high-density complex warehouse environments. It also significantly outperforms JPS in the simple and medium-complexity environments. These results indicate that when regular warehouse structures coexist with local bottlenecks and heterogeneous obstacles, the proposed structure-aware sampling and dynamic connection mechanisms can be effectively converted into observable path-quality gains.

In Experiment II, the positive ΔL in the wide main-corridor scenario indicates that HG-PRM is statistically shorter than LE-HG-PRM by a small margin. In the standard storage corridor and narrow picking corridor scenarios, LE-HG-PRM shows no statistically significant difference from HG-PRM because the two methods produce identical mean

path lengths. This result does not indicate degradation of the proposed method; rather, it suggests that both planners approach a centerline-like solution under sufficiently compressed corridor geometry. Meanwhile, LE-HG-PRM remains significantly shorter than the strongest external baselines in all three corridor-width scenarios, confirming its stable competitiveness in corridor-constrained warehouse layouts.

In Experiment IV, LE-HG-PRM maintains shorter mean paths than HG-PRM across all tested map scales, although these differences are not statistically significant after Holm–Bonferroni correction. Compared with JPS, LE-HG-PRM shows a statistically shorter path in the 200×200 map, whereas JPS achieves significantly shorter paths in the 400×400 and 600×600 maps. This result reveals the scenario boundary of the proposed method: LE-HG-PRM is not intended to unconditionally outperform deterministic search methods at all map scales, but it shows its clearest path-quality advantage in structured warehouse environments where corridor and bottleneck priors can be effectively exploited.

6. Conclusions and Future Work

This paper proposed LE-HG-PRM for static path planning in production-oriented intelligent warehouse environments and systematically evaluated its performance under four complementary conditions: complexity progression, corridor contraction, parameter perturbation, and scale expansion. The experimental results show that the main advantages of the proposed method can be summarized at three levels: path-quality preservation, adaptation to structural constraints, and stability for engineering deployment.

First, LE-HG-PRM demonstrates competitive path-quality performance. Across the three static-complexity scenarios, it achieves the shortest average path length in the simple and medium-complexity environments and remains in the leading group in the high-density complex environment. In the standard and narrow corridor settings, the generated paths further approach the geometrically ideal centerline solution. Under scale expansion, although its comprehensive score is no longer ranked first in every case, its path quality remains stably within the leading group. These results indicate that the proposed method can preserve a competitive and statistically supported path-quality advantage in representative structured warehouse scenarios, particularly when local bottlenecks and corridor constraints are present.

Second, the advantage of LE-HG-PRM is reflected not only in better mean performance, but also in more stable distributional behavior. As shown by the cloud–rain plots in Figures 5, 7, 10 and 12, the method repeatedly exhibits a leftward shift in path-length distribution, reduced dispersion, and suppressed long-tail risk across most experiments. This suggests that the observed gain is not limited to isolated favorable cases, but is associated with a broader tendency toward shorter and more regular paths in the tested structured scenarios.

Third, the results also reveal clear scenario boundaries for the superiority of LE-HG-PRM. Its most significant gains appear in structured warehousing environments characterized by regular global skeletons, local bottlenecks, and strengthened geometric constraints, rather than through unconditional dominance across all scenarios and all metrics. This observation further supports the engineering positioning of the proposed method as a high-quality global planning module for structured logistics environments.

In addition, the parameter-sensitivity analysis shows that LE-HG-PRM has good engineering robustness. The number of sampling points mainly influences the local geometric quality of the generated path, the number of neighbor connections directly affects the topological quality of the roadmap, and the sampling strategy determines how efficiently the limited sampling budget is allocated to critical constrained regions. Overall, the method does not exhibit strong dependence on a single parameter point. According to the present

results, when the number of sampling points is set to 400–600, the number of neighbor connections to 8–10, and the `narrow_first` sampling strategy is adopted, a more favorable balance among path length, path smoothness, and computational cost can be achieved. This indicates that LE-HG-PRM is not only reasonable in algorithmic design, but also practically adjustable and reusable.

Overall, this study demonstrates that LE-HG-PRM is well suited to serve as a high-quality global planner in complex static production warehouse environments, and can provide a reliable basis for subsequent online replanning, multi-robot cooperative scheduling, and intelligent manufacturing-logistics system integration.

Although the performance of LE-HG-PRM has been systematically verified in this work, several directions remain open for further development. First, this study focuses on static environments and does not incorporate time-varying factors such as dynamic obstacles, temporary congestion, or local traffic fluctuation into a unified modeling framework. Future work may couple the proposed method with dynamic obstacle avoidance, local replanning, or spatiotemporal search mechanisms to construct a hybrid planner that balances global path quality with online responsiveness. Second, the current prior mainly originates from explicit geometric rules and does not yet exploit higher-level information such as historical task distributions, congestion hotspots, or semantic functional regions. In the future, learning-based priors or data-driven probability-field modeling could be introduced so that sampling weights can be adaptively updated from long-term operational data. Third, this paper does not address conflict avoidance or task-level coupling in multi-robot systems. As warehouse systems continue to scale up, single-robot global planning should be further co-designed with multi-vehicle scheduling, time-slot allocation, and local traffic-control strategies to improve system-level throughput. Fourth, because the relative advantages of A*, JPS, and LE-HG-PRM vary across scenarios, future research may explore an adaptive switching mechanism between deterministic search and structure-aware roadmap planning, thereby forming a warehouse path-planning framework with stronger scenario awareness and engineering adaptability. Finally, because the present study primarily focuses on algorithmic exploration under controlled static occupancy-grid maps, future work will include validation in more realistic simulation platforms, such as Gazebo-based manufacturing warehouse environments with explicit wall geometry, inflated robot footprints, and sensor constraints, and further tests on physical AGV or AMR platforms to examine deployment-level robustness under sensing, actuation, and localization uncertainties.

Author Contributions: Conceptualization, S.W. and Z.T.; methodology, S.W.; software, S.W.; validation, S.W., G.W. and F.Y.; formal analysis, S.W. and F.Y.; investigation, S.W., G.W., D.P. and X.Y.; resources, S.W., Z.T. and S.Z.; data curation, S.W. and X.L.; writing—original draft preparation, S.W.; writing—review and editing, G.W., F.Y., S.Z. and Z.T.; visualization, S.W. and X.L.; supervision, Z.T.; project administration, S.W. and Z.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request. The original contributions presented in the study are included in the article.

Acknowledgments: The authors thank the editors and reviewers for their constructive comments, which helped improve the clarity, reproducibility, and statistical rigor of this work.

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

Abbreviations

The following abbreviations are used in this manuscript:

AGV	Automated guided vehicle
AMR	Autonomous mobile robot
JPS	Jump point search
PRM	Probabilistic roadmap
RRT	Rapidly exploring random tree
HG-PRM	Heuristic-guided probabilistic roadmap
LE-HG-PRM	Logistics-enhanced heuristic-guided probabilistic roadmap

References

1. Tubis, A.A.; Rohman, J. Intelligent Warehouse in Industry 4.0—Systematic Literature Review. *Sensors* **2023**, *23*, 4105. [CrossRef] [PubMed]
2. Wahab, S.; Safian, S.S.; Othman, N.; Mohamad Azhar, N.A.Z. Motivations To Implement Sustainable Warehouse Management: A Literature Review. *Int. J. Account.* **2021**, *6*, 117–124.
3. Mason, S.J.; Mauricio Ribera, P.; Farris, J.A.; Kirk, R.G. Integrating the warehousing and transportation functions of the supply chain. *Transp. Res. Part E Logist. Transp. Rev.* **2003**, *39*, 141–159. [CrossRef]
4. Liu, X.; Cao, J.; Yang, Y.; Jiang, S. CPS-Based Smart Warehouse for Industry 4.0: A Survey of the Underlying Technologies. *Computers* **2018**, *7*, 13. [CrossRef]
5. Wang, Y.J.; Liu, X.Q.; Leng, J.; Wang, J.J.; Meng, Q.; Zhou, M. Study on scheduling and path planning problems of multi-AGVs based on a heuristic algorithm in intelligent manufacturing workshop. *Adv. Prod. Eng. Manag.* **2022**, *17*, 505–513. [CrossRef]
6. Yi, G.; Feng, Z.; Mei, T.; Li, P.; Jin, W.; Chen, S. Multi-AGVs path planning based on improved ant colony algorithm. *J. Supercomput.* **2019**, *75*, 5898–5913. [CrossRef]
7. Meysami, A.; Cuillière, J.C.; Franois, V.; Kelouwani, S. Investigating the Impact of Triangle and Quadrangle Mesh Representations on AGV Path Planning for Various Indoor Environments: With or Without Inflation. *Robotics* **2022**, *11*, 50. [CrossRef]
8. Bai, Z.; Pang, H.; He, Z.; Zhao, B.; Wang, T. Path Planning of Autonomous Mobile Robot in Comprehensive Unknown Environment Using Deep Reinforcement Learning. *IEEE Internet Things J.* **2024**, *11*, 22153–22166. [CrossRef]
9. Hai, X.; Zhu, Z.; Liu, Y.; Khong, A.W.H.; Wen, C. Resilient Real-Time Decision-Making for Autonomous Mobile Robot Path Planning in Complex Dynamic Environments. *IEEE Trans. Ind. Electron.* **2025**, *72*, 11551–11562. [CrossRef]
10. Song, Q. Semi-Open Multi-Distribution Center Path Planning with Time Windows. *Sustainability* **2010**, *15*, 14.
11. Lei, X.; Chen, J.; Chen, X. Research on Path Planning of Warehouse Robot with Improved Harris Hawks Algorithm. *J. Syst. Simul.* **2024**, *36*, 1081–1092.
12. Pikulin, P.; Lishunov, V.; Kułakowski, K. Optimizing path planning for automated guided vehicles in constrained warehouse environments: Addressing the challenges of non-rotary platforms and irregular layouts. *Robotics* **2025**, *14*, 39. [CrossRef]
13. Chen, X.; Li, Y.; Liu, L. A coordinated path planning algorithm for multi-robot in intelligent warehouse. In *Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6-8 December 2019*; IEEE: New York, NY, USA, 2019; pp. 2945–2950.
14. Li, J.; Tinka, A.; Kiesel, S.; Durham, J.W.; Kumar, T.S.; Koenig, S. Lifelong multi-agent path finding in large-scale warehouses. In *Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021*; Association for the Advancement of Artificial Intelligence (AAAI): Washington, DC, USA , 2021; Volume 35, pp. 11272–11281.
15. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
16. Dijkstra, E.W. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*; ACM Digital Library: New York, NY, USA, 2022; pp. 287–290.
17. Harabor, D.; Grastien, A. Online Graph Pruning for Pathfinding on Grid Maps. In *Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 7–11 August 2011*; Association for the Advancement of Artificial Intelligence (AAAI): Washington, DC, USA, 2011; Volume 25, pp. 1114–1119. [CrossRef]
18. Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **2002**, *12*, 566–580.
19. LaValle, S. Rapidly-exploring random trees: A new tool for path planning. In *Research Report 9811*; Iowa State University: Ames, IA, USA, 1998.
20. De Jong, K. Learning with genetic algorithms: An overview. *Mach. Learn.* **1988**, *3*, 121–138. [CrossRef]

21. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1996**, *26*, 29–41. [CrossRef]
22. Hutter, M. Fitness uniform selection to preserve genetic diversity. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Honolulu, HI, USA, 12–17 May 2002*; IEEE: New York, NY, USA, 2002; Volume 1, pp. 783–788.
23. Nssibi, M.; Manita, G.; Korbaa, O. Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey. *Comput. Sci. Rev.* **2023**, *49*, 100559. [CrossRef]
24. Jathunga, T.; Rajapaksha, S. Improved Path Planning for Multi-Robot Systems Using a Hybrid Probabilistic Roadmap and Genetic Algorithm Approach. *J. Robot. Control* **2025**, *6*, 715–733. [CrossRef]
25. Khaksar, W.; Hong, T.S.; Khaksar, M.; Motlagh, O. A low dispersion probabilistic roadmaps (LD-PRM) algorithm for fast and efficient sampling-based motion planning. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 397.
26. Huang, Y.; Wang, H.; Han, L.; Xu, Y. Robot path planning in narrow passages based on improved PRM method. *Intell. Serv. Robot.* **2024**, *17*, 609–620. [CrossRef]
27. Chen, S.; Yang, G.; Cui, G.; Yi, S.; Wu, L. Improved path planning and controller design based on PRM. *IEEE Access* **2025**, *13*, 44156–44168. [CrossRef]
28. Wang, S.; Zhang, S.; Tian, Z.; Yao, Y.; Wang, G.; Zhao, Y. Heuristic-guided PRM: A Dynamic Weighting Approach for Optimal Path Planning in Complex Environments. In *Proceedings of the 2025 IEEE 7th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), Yinchuan, China, 22–24 October 2025* ; IEEE: New York, NY, USA , 2025; pp. 715–723. [CrossRef]
29. Pico, N.; Montero, E.; Amirbek, A.; Auh, E.; Jeon, J.; Alvarez-Alvarado, M.S.; Jamil, B.; Algabri, R.; Moon, H. Human and environmental feature-driven neural network for path-constrained robot navigation using deep reinforcement learning. *Eng. Sci. Technol. Int. J.* **2025**, *64*, 101993. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.