Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Obstacle-Avoidance Path Planning for a Basket-Carrying Robotic Arm Based on an Improved APF-RRT Algorithm

Shilei Lyu
South China Agricultural University
Shengjun Shi
South China Agricultural University
Guoning Zhang
South China Agricultural University
Junxing He
South China Agricultural University
Leyuan Chen
South China Agricultural University
Xiongzhe Han
Kangwon National University
Peng Gao
gaopeng.peng@scau.edu.cn
South China Agricultural University

Research Article

Keywords: agricultural robot, material-handling robotic arm, path planning, Improved RRT algorithm, improved Artificial Potential Field method

Posted Date: July 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10165181/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Journal of Biosystems Engineering on November 4th, 2026. See the published version at https://doi.org/10.1007/s42853-026-00327-x.

Obstacle-Avoidance Path Planning for a Basket-Carrying Robotic Arm Based on an Improved APF-RRT Algorithm

Abstract

Purpose: This study addresses the challenges of robotic arm path planning in unstructured material-handling environments typical of agricultural robotics—including complex obstacle distributions, weak goal-directedness of conventional RRT algorithms, low convergence efficiency, and poor path smoothness. **Methods:** An RRT-based path planning algorithm integrated with an improved Artificial Potential Field (APF) method is proposed. A power-law decay repulsive gain coefficient eliminates the goal-unreachability problem; a three-level progressive escape mechanism combined with a penalty-weighted distance metric enables the random tree to bypass local deadlock regions; an adaptive step-size strategy based on local environmental awareness and a dynamic goal-bias probability strategy achieve a self-adaptive balance between global exploration and local goal convergence. Multi-pass greedy pruning and cubic B-spline smoothing are applied in post-processing to remove path redundancy and guarantee C^2 continuity. **Results:** Comparative simulations and real-robot grasping experiments were conducted in 2D and 3D configuration spaces and on a 6-DOF collaborative robotic arm platform. In 2D scenarios, the improved algorithm reduces average iteration counts by 85.8%–89.7% and planning time by 42.1%–56.4% relative to conventional RRT. In 3D space, planning is completed in an average of 17.1 iterations with path length shortened by

31.4%. In joint-space experiments on a 6-DOF robotic arm, the proposed algorithm reduces path cost by 38.9%, 43.9%, and 32.9% relative to RRT, RRT-Connect, and RRT*, respectively, while cutting planning time by 10.1%–20.9%.

Conclusions: Real-robot trials confirm smooth, jitter-free motion across all joints, demonstrating that the algorithm effectively satisfies engineering requirements for planning responsiveness and motion stability in complex agricultural environments.

Keywords: agricultural robot; material-handling robotic arm; path planning; Improved RRT algorithm; improved Artificial Potential Field method

1. Introduction

In modern agricultural production, rising labor costs and growing demand for automation have led to the widespread deployment of agricultural robots in tasks such as field harvesting, transfer, and palletizing (Jin & Han, 2024). As the core actuator of agricultural robotic systems, the robotic arm is responsible for the critical tasks of autonomously grasping and transporting field baskets (Liu et al., 2024). Path planning for material-handling robotic arms constitutes the central link of the entire transportation process, directly determining operational efficiency and motion stability. In field environments, irregularly distributed crop branches form random obstacles, and basket poses are highly variable and cannot be predetermined, making collisions between the robotic arm and obstacles highly likely during transport—potentially damaging the arm or causing grasping failures due to vibration-induced basket displacement (Zhao et al., 2026). Consequently, how to rapidly plan low-cost, highly

smooth, collision-free paths within a six-dimensional joint space under unstructured obstacle conditions has become a key technical challenge for automating robotic arm conveying operations in farmland settings.

The goal of robotic arm path planning is to search, within the high-dimensional configuration space formed by the joint angles, for a feasible path connecting the start configuration to the goal configuration while remaining collision-free throughout (Lozano-Pérez & Wesley, 1979). Current mainstream approaches include graph-search algorithms, metaheuristic algorithms, and sampling-based algorithms. Graph-search methods typified by A* require discrete modeling of the workspace, and their computational complexity grows exponentially with dimensionality, making them impractical for six-dimensional or higher-dimensional joint spaces (Hart et al., 1968). Metaheuristic methods such as genetic algorithms and particle swarm optimization possess strong global search capabilities but are highly sensitive to hyperparameter settings, leading to slow convergence and poor solution stability in high-dimensional complex problems (Ab Wahab et al., 2024). In recent years, data-driven planning methods based on deep reinforcement learning have demonstrated adaptive capabilities in unknown environments; some studies employ neural networks for sampling-strategy learning or collision prediction to reduce planning time (Qureshi et al., 2020). However, these approaches are constrained by their dependence on large-scale labeled training data, the difficulty of strictly guaranteeing collision safety within joint-constrained spaces, and unstable generalization when transferred to new scenarios, limiting their direct application to industrial-grade conveying tasks (Lai et al., 2025). By contrast,

random-sampling algorithms typified by the Rapidly-exploring Random Tree (RRT) require no explicit environmental modeling, are naturally suited to high-dimensional configuration spaces, and possess probabilistic completeness, earning them the widest adoption in robotic arm path planning (LaValle, 1998). Nevertheless, conventional RRT suffers from weak goal-directedness and low convergence efficiency, readily generating large numbers of invalid nodes in obstacle-dense regions, resulting in high path redundancy and pronounced turns that cannot satisfy the real-time planning and path-smoothness requirements of agricultural conveying operations (Wang et al., 2024). Accordingly, targeted improvements to the RRT algorithm for the complex working conditions of unstructured farmland conveying scenarios have become a core research direction that urgently needs to be addressed.

Numerous researchers have proposed significant improvements to address the inefficiency of the RRT algorithm from various perspectives. The RRT* algorithm proposed by Karaman and Frazzoli (Karaman & Frazzoli, 2011) introduces parent-node re-selection and rewiring mechanisms to endow the algorithm with asymptotic optimality, enabling continuous path-quality improvement as the number of iterations grows; however, it requires extensive sampling iterations before converging to a high-quality path, limiting its practicality for time-sensitive tasks. The RRT-Connect algorithm by Kuffner and LaValle (Kuffner & LaValle, 2000) expands the random tree bidirectionally from both start and goal while employing a greedy connection strategy, substantially improving planning speed, yet it lacks a path-quality optimization mechanism and produces paths with many waypoints and high cost. The

Informed RRT* algorithm proposed by Gammell et al. (Gammell et al., 2014) restricts the sampling space to an ellipsoidal domain after an initial path is found, accelerating post-hoc optimization convergence, but the quality of the initial path directly constrains the final outcome. The Quick-RRT* algorithm by Jeong et al. (Jeong et al., 2019) expands the candidate parent-node set using the triangle inequality to obtain better initial solutions with fewer nodes, but the ancestor-depth value is manually specified and lacks adaptability across different scenarios. The AEB-RRT algorithm proposed by Wang et al. (Wang et al., 2022) enhances environmental-awareness sensitivity by adaptively adjusting the bidirectional expansion sampling probability; however, its dependence on the total number of collision-detection failures still makes active escape from local minima regions difficult.

Integrating the Artificial Potential Field (APF) method with RRT represents an important research direction for further enhancing the goal-directedness of the algorithm. APF applies heuristic guidance to the node-expansion direction by constructing an attractive field toward the goal and repulsive fields from obstacles, enabling the random tree to grow more efficiently and directionally toward the target configuration, thereby reducing invalid sampling (Khatib, 1986). Ye et al. proposed the IKB-RRT algorithm for orchard mobile robot path planning (Ye et al., 2023), generating more informative guidance nodes for the bidirectional tree expansion process through an improved APF, while introducing a motion-control-parameter sampling method and a dual-tree connection path optimization strategy; their approach outperformed comparison algorithms in path length, planning time, and overall

smoothness, providing an important reference for APF-RRT integration in unstructured agricultural environments. The BPFPS-RRT algorithm proposed by Liu et al. (Liu et al., 2023) combines APF with bidirectional goal-bias and probabilistic step-size strategies, jointly determining step sizes via target angle and random values to adaptively handle regions of varying obstacle density, effectively reducing search time and path length. Duan et al. addressed the path planning problem for a plastering robotic arm in refractory brick masonry environments, proposing an improved RRT algorithm fusing a zonal Gaussian sampling strategy and an attraction-based dynamic step-size mechanism (Duan et al., 2026); by guiding the random tree toward critical regions through zone-based sampling and combining triangle-inequality pruning with cubic B-spline optimization to improve path smoothness, their approach demonstrated practical engineering utility in real masonry environments. Zhao et al. proposed the GD-RRT algorithm for apple-picking scenarios (Zhao et al., 2026), which dynamically updates a Gaussian mixture probability distribution sampling model based on collision-detection success frequency and adjusts the expansion depth value in real time, effectively resolving the insufficient adaptability of static parameter settings across regions with varying obstacle densities. The TBIRRT algorithm by Zhang et al. (Zhang et al., 2026) fuses adaptive path-expansion optimization, bidirectional search, and goal-bias mechanisms, adaptively regulating the sampling temperature parameter via the Metropolis criterion to significantly improve planning efficiency in complex multi-posture scenarios. The AS-RRT algorithm proposed by Li et al. (Li et al., 2024b) designs an accumulator-based sampling-point selection strategy and a post-collision

dynamic step-size adjustment mechanism, further optimizing planning quality through post-processing. Kang et al. improved the RRT algorithm by adjusting the expansion direction and adopting a dynamic step-size strategy (Kang et al., 2024), achieving efficient path search for multi-DOF robots in complex unstructured environments.

In practical conveying operations, the diverse placement poses of baskets require the robotic arm to reach the target configuration with a specific end-effector orientation; path smoothness requirements are particularly stringent to ensure stable joint motion after grasping the object. Meanwhile, obstacles formed by crop branches in field environments readily create numerous local-minima regions in the joint space, impeding normal expansion of the random tree. To address these challenges, this paper proposes an improved RRT robotic arm path planning algorithm tailored for farmland conveying scenarios. The algorithm deeply integrates an improved APF into the RRT node-expansion mechanism, providing heuristic expansion guidance through a multi-field coupling vector that combines goal attraction, obstacle repulsion, and random exploration forces. A power-law decay repulsive gain coefficient is adopted to eliminate goal unreachability; a three-level progressive escape mechanism combined with a penalty-weighted distance metric actively addresses local-minima traps; an adaptive step-size strategy based on local environmental awareness together with a dynamic goal-bias probability strategy coupled to iterative progress realizes self-adaptive spatial exploration during planning; and multi-pass greedy pruning combined with cubic B-spline smoothing in post-processing simultaneously ensures low path cost and motion smoothness.

2. Method

2.1. Node Expansion Mechanism Integrating the Improved APF

Traditional Random Tree Exploration (RRT) algorithms employ globally uniform random sampling to drive random tree expansion, with the expansion direction entirely dependent on the location of the random sampling points. This unguided, purely exploratory mechanism, lacking prior information about the target, is prone to generating numerous invalid nodes in a high-dimensional joint space dense with obstacles, resulting in extremely high iteration costs and very low convergence efficiency. The Artificial Potential Field (APF) method achieves real-time guidance of the configuration space by constructing a superposition of gravitational and repulsive fields; however, its inherent local minima and target unreachability problems limit the planning success rate when applied independently. To address these issues, this paper deeply integrates an improved APF into the node expansion stage of the RRT algorithm. By constructing a multi-potential field coupled resultant force vector containing target attraction, obstacle repulsion, and random exploration forces, a node expansion mechanism integrating the improved APF is proposed. The schematic diagram of new node expansion for the improved RRT algorithm fused with the artificial potential field method is shown in Fig. 1.

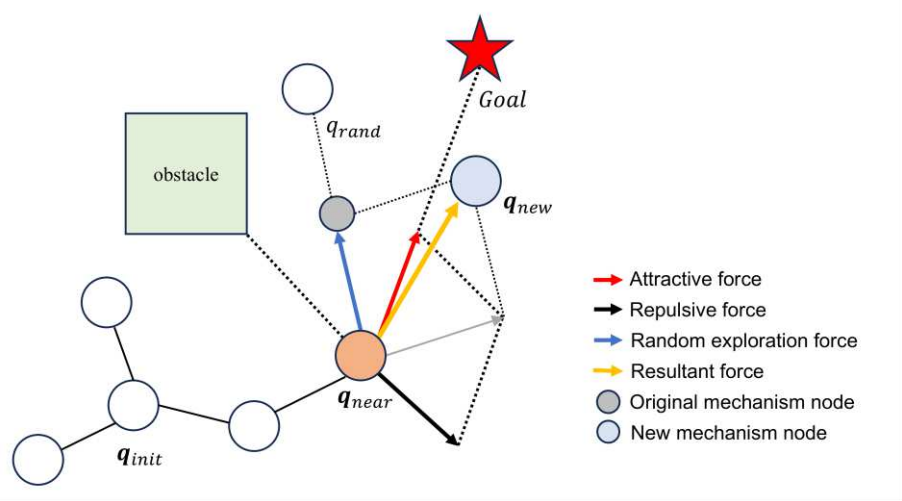


Fig.1 Schematic diagram of new node expansion mechanism.

During node expansion, three directional components—goal attraction, obstacle repulsion, and random exploration force—are simultaneously applied to the current nearest-neighbor node q_near of the random tree, and the normalized direction of their resultant is used as the expansion direction in place of the pure random direction. The resultant force vector is computed as:

$$F(n)=R_{att}(n)+R_{rep}(n)+R_{dt}(n) \quad (1)$$

where $R_{att}(n)$ is the goal attraction component; $R_{rep}(n)$ is the obstacle repulsion component; $R_{dt}(n)$ is the random exploration force component.

The goal attraction component is taken as the unit direction vector from the current node to the target configuration q_goal, weighted by the attraction gain coefficient k_{att}, to drive the random tree to grow directionally toward the goal.

$$R_{att}(n)=k_{att}\frac{q_{goal}-q_{near}}{\|q_{goal}-q_{near}\|} \quad (2)$$

The obstacle repulsive force component is established based on the classical artificial repulsive potential field gradient. Let $\rho = \rho(q_{near}, q_{obs})$ be the shortest distance between the current node and the nearest obstacle, and ρ_0 be the radius of influence of the repulsive force. When the node intrudes into the obstacle's influence domain （$\rho \le \rho_0$）, the repulsive force component is:

$$R_{rep}(n)=k_r\left(\frac{1}{\rho}-\frac{1}{\rho_0}\right)\frac{1}{\rho^2}\frac{\partial(q_{near}-q_{obs})}{\partial q_{near}} \quad (3)$$

where k_r is the repulsive gain coefficient; q_{obs} is the position of the nearest obstacle reference point.

The random exploration force component is taken as the unit direction vector from the current node to the random sampling point qrand, preserving the RRT algorithm's advantage of random coverage of the configuration space and preventing the state tree from converging prematurely to the potential field minimum:

$$R_{dt}(n)=\frac{q_{rand}-q_{near}}{\|q_{rand}-q_{near}\|} \quad (4)$$

After normalizing the resultant force vector obtained from Eq. (1), we take it as the unit vector $\hat{F}(n)$ of the expansion direction and multiply it by the adaptive step size α to obtain the new node position:

$$q_{new}=q_{near}+\alpha\hat{F}(n) \quad (5)$$

In conventional APF mechanisms, when the target configuration is located near an obstacle, the repulsive potential field retains its full strength in the goal vicinity, preventing the resultant force from converging to zero and directly causing the algorithm to become trapped in the goal-unreachability dilemma. Most existing

improvement methods introduce a decay factor related to the node-to-goal distance to modulate the repulsive gain coefficient; the typical form is exponential decay:

$$k_r = k_{r0}\left(1 - e^{-\beta d_{goal}}\right) \tag{6}$$

where $d_{goal} = \| q_{goal} - q_{near} \|$. However, the residual repulsion under this form decreases only linearly as $O(d_{goal})$ when $d_{goal} \to 0$, which can still prevent node convergence to the target region in narrow spaces. To address this, this paper constructs a repulsive force gain coefficient model based on power-law decay:

$$k_r(d_{goal}) = k_{r0} \min\left(d_{goal}^{\beta},\ 1\right) \tag{7}$$

where k_{r0} is the initial repulsive gain coefficient; β is the power-law exponent, set to $\beta = 2$ in this work; d_{goal} is the Euclidean distance between the current node and the goal point.

Compared to exponential decay, Eq. (7) causes the residual repulsion to diminish at a higher-order rate of $O\left(d_{goal}^{\beta}\right)$ as $d_{goal} \to 0$. Since $\beta > 1$, the decay rate in the goal vicinity is significantly increased, more thoroughly eliminating residual repulsion interference; in regions far from the goal ($d_{goal} \gg 1$), k_r rapidly recovers to the maximum threshold k_{r0}, preserving normal obstacle-avoidance capability. Substituting Eq. (7) into Eq. (3) yields the improved repulsion component:

$$R_{rep}(n) = k_r(d_{goal})\left(\frac{1}{\rho} - \frac{1}{\rho_0}\right)\frac{1}{\rho^2}\frac{\partial(q_{near} - q_{obs})}{\partial q_{near}},\quad \rho \le \rho_0 \tag{8}$$

After eliminating the goal-unreachability factor, stagnation at local minima caused by the local cancellation of attraction and repulsion forces still significantly undermines convergence efficiency and impedes normal growth of the random tree. To address this,

the present work introduces an expansion failure counter f for each node to quantify the depth of stagnation, and designs a three-level progressive escape mechanism that progressively increases the intensity and randomness of the escape strategy as stagnation deepens. Let the hard escape threshold be T_{esc} and the soft escape threshold be $T_{soft} = T_{esc}/2$; the node expansion states are classified as follows.

Normal Mode ($f < T_{soft}$): Node expansion was not substantially hindered and was directly expanded using the aforementioned multi-field coupling resultant force direction and adaptive step size α.

Soft Escape Mode ($T_{soft} \le f \le T_{esc}$): The node falls into a slight force field equilibrium. The algorithm superimposes a Gaussian-distributed random perturbation along the resultant force direction to deflect the expansion direction and disrupt the force equilibrium:

$$\mathrm{d}=\mathrm{normalize}\left(\hat{\mathrm{F}}_{\mathrm{APF}}+\xi\right),\ \ \xi\sim\mathrm{N}(0,\ \ \sigma^2\mathrm{I}) \tag{9}$$

where $\hat{F}_{APF}$ is the normalized unit vector of the APF resultant force; ξ is a zero-mean Gaussian random vector with independent and identically distributed components; σ is the perturbation standard deviation, which increases linearly from σ_{min} to σ_{max} as the failure count increases within the interval $[T_{soft},\ T_{esc})$, so that the perturbation intensity adaptively increases with the depth of stagnation.

Soft Escape Mode retains the dominant potential-field guidance direction and only breaks the force equilibrium through random perturbations, providing the node with a perturbation increment that deviates from the local-minima region without losing the goal direction.

Hard Escape Mode ($f > T_{esc}$): The node is trapped in a deep local trap. The algorithm completely suppresses potential-field guidance and instead randomly generates N_{try} candidate directions in the configuration space, evaluates the effective collision-free traversal span $\triangle_i$ along each direction via local collision detection, and selects the direction with the greatest safe-space depth as the final escape direction:

$$d^* = \arg\max_{i \in \{1,\cdots,N_{try}\}} \triangle_i \quad (10)$$

where $\triangle_i$ is the actual traversable distance along the i-th candidate direction as evaluated by collision detection; N_{try} is the number of candidate directions.

This multi-candidate best-selection strategy significantly improves the single-step escape success rate compared to pure random walk. Once expansion along the selected direction succeeds, the system immediately resets the failure counter f to zero, allowing the algorithm to quickly return to attraction-dominated mode.

Furthermore, to prevent the nearest-neighbor search (which conventionally relies on pure Euclidean distance) from repeatedly selecting deadlocked nodes already trapped in local minima, a failure-count penalty term is incorporated into the spatial distance metric to construct a penalty-weighted distance:

$$d_{pen}(q_i, q_{rand}) = \|q_i - q_{rand}\| + f_i \cdot s_{step} \quad (11)$$

where f_i is the cumulative expansion failure count of node i; s_{step} is the current expansion step size (nominal step size); $\| q_i - q_{rand} \|$ is the Euclidean distance between node i and the random sample point.

Eq. (11) unifies the failure penalty and spatial distance under the same metric scale

using the expansion step size as the unit. Each expansion failure is equivalent to pushing the node further away by one step size in the configuration space. As the number of failures f_i accumulates, the weighted distance of the node increases monotonically. The nearest neighbor search will adaptively bias the expansion starting point towards the node with a higher historical success rate, effectively improving the overall expansion efficiency without changing the tree structure. Furthermore, when f_i exceeds the ultimate escape threshold, the node is permanently discarded, fundamentally eliminating the continued consumption of computational resources by ineffective regions.

2.2. Adaptive Step-Size Strategy

In conventional RRT, step size is a critical factor directly affecting spatial exploration efficiency. The fixed step-size strategy faces a fundamental trade-off: excessively large steps in obstacle-dense regions lead to frequent expansion failures, while excessively small steps in free space reduce exploration speed. To address this, the present work proposes an adaptive step-size strategy based on local environmental awareness, using the distance from the current node to the nearest obstacle d_{obs} and the distance to the goal d_{goal} as dual inputs to dynamically regulate the expansion step size.

An obstacle proximity factor s_o and a goal proximity factor s_g are defined respectively as:

$$s_o=\min\left(\frac{d_{obs}}{d_{obs_margin}},1\right),\quad s_g=\min\left(\frac{d_{goal}}{d_{goal_margin}},1\right) \tag{12}$$

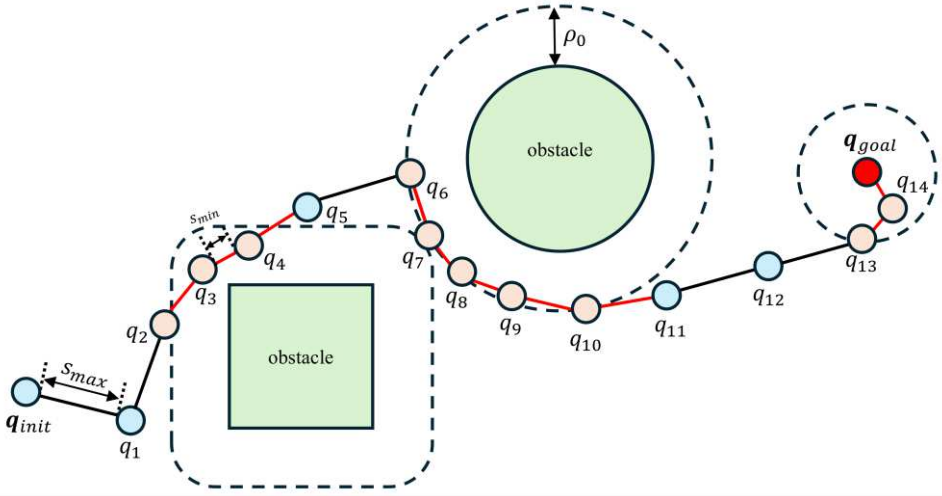
where d_{obs_margin} is the obstacle safety distance threshold; d_{goal_margin} is the goal neighborhood radius threshold; s_o, $s_g \in [0, 1]$ reflect the proximity of the current node to the obstacle and to the goal, respectively.

The adaptive step size is taken as a linear mapping of the smaller of the two factors:

$$\alpha = s_{min} + (s_{max} - s_{min}) \min(s_o, s_g) \quad (13)$$

where s_{min} and s_{max} are the lower and upper bounds of the step size.

When a node resides in free space and is far from the goal, both factors approach 1 and the step size approaches the upper bound s_max, ensuring rapid coverage of the configuration space; when a node approaches an obstacle or the goal, the environmental constraint factors activate and the step size decreases sharply, improving collision-detection pass rates in narrow corridors and enhancing goal-connection accuracy. This strategy enables the step size to vary continuously according to the local environment throughout the planning process, balancing exploration efficiency and motion safety. The principle of this strategy is shown in Fig. 2.



311 Fig.2 Schematic diagram of the adaptive step size strategy.

312

313 **2.3. Dynamic Goal-Bias Probability Strategy Based on Iterative Progress**

314 Goal-biased sampling is a commonly used technique to improve convergence,
315 wherein the goal configuration is used as the sample point with a certain probability to
316 reinforce the tendency of the random tree to grow toward the goal. Existing methods
317 either employ a fixed bias probability or set a static threshold based on the volumetric
318 proportion of obstacles in the environment. A fixed bias cannot adapt to the dynamic
319 changes of the planning process, and a static threshold only maps the structural features
320 of the environment while decoupling from the current planning state; neither can
321 adequately satisfy the ideal sampling requirement of "broad exploration in early stages,
322 directed convergence in later stages." To address this, the present work proposes a
323 dynamic goal-bias probability strategy coupled to iterative progress.

324 Let the current iterative progress be $\tau = iter/iter_{max} \in [0,1]$; the goal-bias
325 probability p_{dt} is updated according to the following piecewise function:

326 $$p_{dt}(\tau)=p_{min}+(p_{max}-p_{min})\phi(\tau) \quad (14)$$

327 where p_{min} and p_{max} are the lower and upper bounds of the bias probability;
328 $\phi(\tau)$ is a strictly monotonically increasing mapping function of the iterative progress
329 τ, with $\phi(\tau) \in [0,1]$.

330 To eliminate the destabilizing effect of abrupt probability changes on planning
331 stability, the mapping function $\phi(\tau)$ employs a piecewise smooth design: a cubic
332 Hermite interpolation function is used during the early planning phase ($\tau \leq \tau_0$) to

achieve a smooth, cliff-free rise at the beginning of planning; a linear function is used during the later phase ($\tau > \tau_0$) to steadily increase toward the upper bound. Here τ_0 is the piecewise breakpoint; the cubic Hermite interpolation function satisfies zero first-derivative boundary conditions at both endpoints, guaranteeing that the probability curve is continuously differentiable at $\tau = 0$ and $\tau = \tau_0$:

$$\phi(\tau)=\phi_0 \cdot h(\tau/\tau_0),\ \ h(u)=3u^2-2u^3,\ \ \tau \le \tau_0 \tag{15}$$

$$\phi(\tau)=\phi_0+(1-\phi_0)\frac{\tau-\tau_0}{1-\tau_0},\ \ \tau>\tau_0 \tag{16}$$

where τ_0 is the piecewise breakpoint (set to $\tau_0 = 0.3$ in this work); ϕ_0 is the function value at the endpoint of the early-phase mapping (set to $\phi_0 = 0.75$ in this work); $h(u) = 3u^2 - 2u^3$ is the cubic Hermite smooth step function, satisfying $h(0) = 0$、$h(1) = 1$, and zero derivatives at both endpoints.

Equations (14)-(16) together ensure that p_{dt}t is strictly monotonically increasing and continuously differentiable throughout the planning process. In the early stage of planning, τ is small and p_{dt} is at a low level. The algorithm mainly uses uniform random sampling to fully cover the configuration space. As the planning progresses, p_{dt} gradually increases, and the algorithm concentrates the sampling resources in the target direction to accelerate convergence. Compared to bias strategies that rely solely on static environmental parameters, the proposed method introduces planning progress into the sampling decision, enabling sampling behavior to adaptively evolve with the iterative process and conferring broader environmental applicability.

2.4. Path Optimization Methods

The initial paths produced by sampling-based path planning algorithms typically contain numerous redundant nodes, resulting in elevated path cost and unnecessary detours that cannot directly satisfy the trajectory-quality requirements of robotic arm motion execution. The present work employs a multi-pass greedy pruning strategy for iterative post-processing of the initial path. The core idea is: starting from the current node, the algorithm successively attempts to establish direct connections to successor nodes as close to the path end as possible; if the collision check passes, all intermediate redundant nodes are deleted and the simplified path is retained. Let the path node sequence be $\{q_0, q_1, \cdots, q_N\}$; the execution steps of a single-pass greedy pruning are as follows:

Set the current node index to $\mathrm{i} = 0$, and search backward from the path end for the farthest index $j(j > i + 1)$ satisfying the following condition:

$$\|q_i - q_j\| \ll \mathrm{D_{max}} \text{ and } \mathrm{CollisionFree}(q_i, q_j) = \mathrm{Ture} \quad (17)$$

where D_{max} is the maximum direct-connection distance threshold; $\mathrm{CollisionFree}(\cdot)$ is the collision detection function for the path between two nodes.

If a qualifying index j exists, q_j is added to the new path and i is set to j to continue searching; otherwise i is incremented to $i + 1$ and q_i is added to the path. After a single-pass scan is complete, if the node count does not decrease compared to the previous pass, the pruning process has converged; otherwise the above scan is repeated on the resulting path until convergence. The path optimization results of the multi-pass greedy pruning strategy are shown in Fig. 3. Given the original path node sequence $\{q_{init}, q_1, \cdots, q_{goal}\}$, after the first pass of corrupt pruning, redundant node

q_4 is removed, and after the second pass of pruning, redundant node q_7 is removed. After the next single pass of pruning, the number of path nodes is reduced, resulting in the final path node sequence $\{q_{init}, q_1, q_2, q_3, q_5, q_6, q_{goal}\}$.The purpose of introducing the maximum direct-connection distance threshold D_{max} is to control the computational cost of collision detection and avoid unnecessary kinematic verification for node pairs that are too far apart. Through iterative refinement of pruning results, the multi-pass scanning strategy more thoroughly eliminates path redundancy compared to a single-pass greedy strategy.

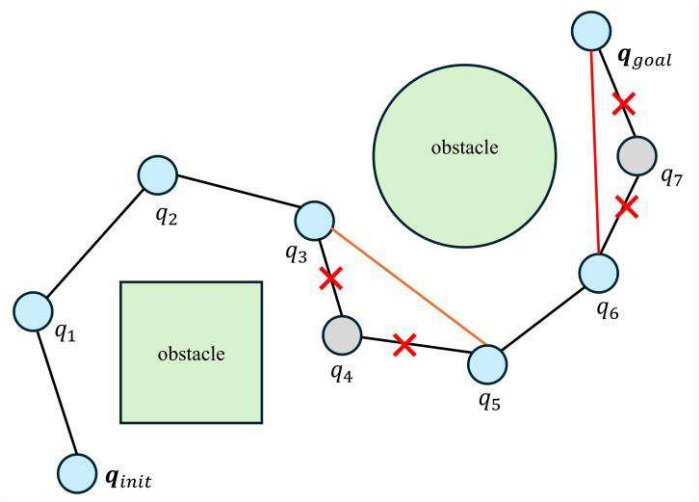


Fig.3 Diagram of multi-pass greedy pruning strategy.

The path produced after greedy pruning consists of piecewise linear segments with discontinuous tangent vectors at the waypoints, failing to satisfy the smoothness requirements of robotic arm motion trajectories. To address this, uniform cubic B-spline smoothing is applied to the pruned path, elevating the piecewise linear path to a C^2-continuous parametric curve, eliminating curvature discontinuities at the

waypoints, and keeping the angular accelerations of all robotic arm joints continuous.

Let the pruned path node sequence $\{p_0, p_1, \cdots, p_m\}$ serve as B-spline control points. To ensure the generated curve strictly passes through the start and end points of the path, the first and last control points are each repeated twice, constructing the extended control point sequence $\{p_0, p_0, p_0, p_1, \cdots, p_m, p_m, p_m\}$. The i-th spline segment is defined by the following parametric equation:

$$P_i(u)=\sum_{k=0}^{3} B_k(u)\, p_{i+k}, \quad u\in[0,1] \tag{18}$$

The uniform cubic B-spline basis functions adopt the standard form derived from the de Boor recursion:

$$B_0(u)=\frac{(1-u)^3}{6}, \quad B_1(u)=\frac{3u^3-6u^2+4}{6} \tag{19}$$

$$B_2(u)=\frac{-3u^3+3u^2+3u+1}{6}, \quad B_3(u)=\frac{u^3}{6} \tag{20}$$

The above basis functions satisfy non-negativity and the partition-of-unity property; adjacent segments possess C^2 continuity at the common knots, guaranteeing everywhere-continuous curvature of the synthesized path curve. By evaluating Eq. (18) segment by segment over the control point sequence and uniformly sampling several interpolation points per segment, the smoothed dense path-point sequence is obtained.

To ensure that the smoothing operation does not introduce collisions, collision detection is performed on all adjacent point pairs of the smoothed path. If any adjacent segment is found to interfere with an obstacle, the smoothing result is discarded and the piecewise linear path from greedy pruning is used directly as the final planning output to guarantee path safety. By simultaneously reducing path length and improving

smoothness, the proposed path post-processing method effectively enhances the trajectory quality of robotic arm motion.

3. Experiments and results

3.1. Algorithm Simulation Experiments and Analysis

To systematically evaluate the planning performance of the proposed algorithm under different obstacle distribution environments, comparative simulation experiments were conducted in both 2D planar scenes and 3D spatial scenes, with comparisons against RRT, RRT*, and RRT-Connect. The experimental platform was built on an Intel Core i9-10900K CPU, 32 GB RAM, RTX 3080 GPU, Windows 10 OS, and MATLAB R2022a; all experiments were conducted under identical hardware and software conditions. To eliminate statistical fluctuations introduced by random sampling, each algorithm was independently run 100 times in each test scenario, and the mean of each metric was used as the final evaluation basis.

3.1.1. 2D Simulation Experiments and Analysis

The 2D simulation map was set to 1000 mm × 1000 mm, with the start point at coordinates (50, 50) and the goal point at (950 mm, 950 mm). To comprehensively investigate the adaptability of the algorithm under different obstacle distribution conditions, three representative planning scenarios were constructed: Environment 1 is a sparse-obstacle scene, primarily examining path quality and planning efficiency at low obstacle density; Environment 2 is a narrow-corridor scene, focused on evaluating

the algorithm's traversal capability in highly constrained spaces and its planning success rate; Environment 3 is a complex, dense-obstacle scene, comprehensively testing the robustness and convergence of the algorithm in high obstacle density environments with multiple local minima. Unified parameter settings for all algorithms are as follows: maximum iteration count 50,000; step size 0.10 rad; in the proposed algorithm, attraction gain coefficient $k_{att} = 1.5$, repulsive influence radius $\rho_0 = 0.40$, power-law exponent $\beta = 2$, and escape threshold $T_{esc} = 8$.

The planned paths of each algorithm in the three 2D environments are shown in Figs. 4–6, and the detailed simulation statistics are summarized in Table 1.

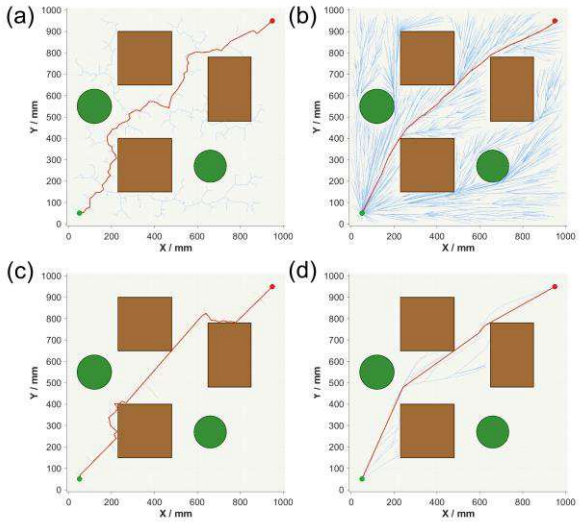
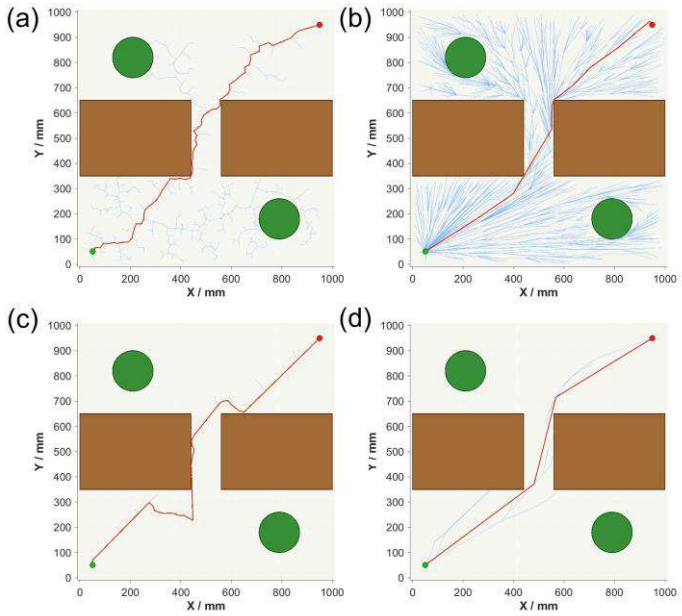


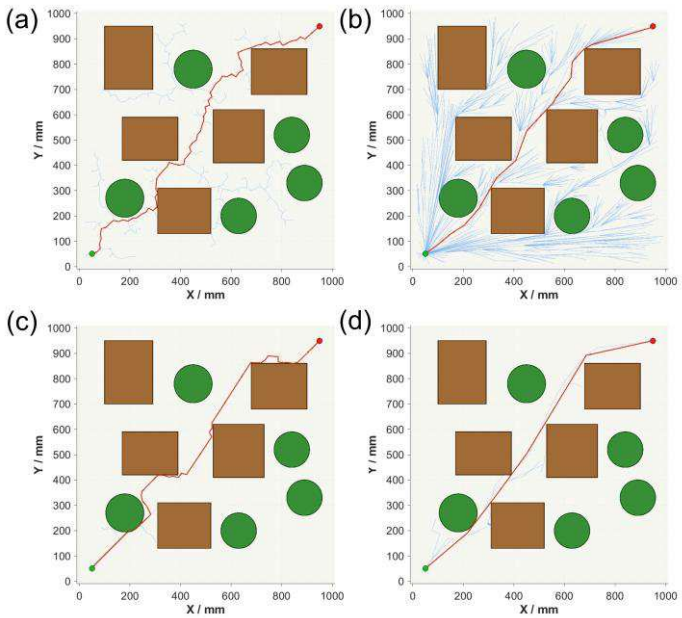
Fig.4 Paths planned by various algorithms in environment 1.



447

448 Fig.5 Paths planned by various algorithms in environment 2.

449



450

451 Fig.6 Paths planned by various algorithms in environment 3.

452

453 Table 1. Simulation data statistics of each algorithm in different 2D environments.

Environment	Algorithm	Path length/mm	Cohen's d [95% CI]	Planning time/s	Number of iterations	Significance
Environment 1 (Simple obstacles)	RRT	1622.24±99.50	4.27[−329.46, −274.11]	0.019±0.004	434.60±96.87	***
	RRT*	1319.32±9.82	0.11[−2.64, 4.91]	0.594±0.179	1095.74±440.17	ns/***/***
	RRT-Connect	1486.62±65.21	3.52[−183.13, −145.20]	0.006±0.001	88.32±22.09	***
	Improved APF-RRT	1320.45±10.36	—	0.011±0.004	61.54±28.57	—
Environment 2 (Narrow passage)	RRT	1666.29±119.86	3.82[−359.18, −291.17]	0.015±0.005	443.58±179.26	***
	RRT*	1331.72±12.76	0.76[4.58, 14.21]	0.403±0.154	984.12±448.34	***
	RRT-Connect	1550.76±123.16	2.40[−245.66, −173.61]	0.006±0.003	69.08±48.18	***/**/***
	Improved APF-RRT	1341.12±12.11	—	0.008±0.002	45.78±18.64	—
Environment 3 (Complex obstacles)	RRT	1654.35±78.79	5.75[−347.32, −302.33]	0.039±0.010	557.76±141.49	***
	RRT*	1339.26±14.70	0.70[−15.39, −4.08]	1.172±0.299	1245.20±441.61	**/***/***
	RRT-Connect	1484.91±60.43	3.56[−173.28, −137.48]	0.012±0.003	107.90±46.35	***
	Improved APF-RRT	1329.53±12.97	—	0.017±0.004	72.20±23.94	—

Notes:

[a] Cohen's d represents the absolute effect size, used to quantify the magnitude of performance differences between algorithms. Effect size interpretation follows Cohen (1988): $d < 0.2$ negligible, 0.2–0.5 small, 0.5–0.8 medium, $d > 0.8$ large. The 95% CI is the confidence interval for the mean path-length difference ("comparison algorithm − proposed algorithm"), computed using paired two-tailed t-tests ($\alpha = 0.05$); positive values indicate the comparison algorithm outperforms the proposed algorithm, negative values indicate the opposite.

[b] Significance markers correspond, in order, to path length / iteration count / planning time; when all three agree, a single marker is given. ***: $p < 0.001$; **: $0.001 \le p < 0.01$; *: $0.01 \le p < 0.05$; ns: $p \ge 0.05$. The proposed algorithm is the reference baseline and carries no significance marker.

As shown in Table 1, in terms of average path length, the proposed improved algorithm is compared with RRT*, which focuses specifically on path-length improvement: in the complex-obstacle environment, the proposed algorithm further reduces path length by 0.7% relative to RRT*, with comparable results in the other two environments but substantially improved planning time and iteration counts. Across all environments, the proposed algorithm achieves shorter path lengths than all other algorithms—reductions of 18.6% and 11.2% versus RRT and RRT-Connect in Environment 1; 19.5% and 13.5% in Environment 2; and 19.6% and 10.5% in Environment 3—further confirming the effectiveness of the path optimization strategy.

Regarding iteration counts, the proposed algorithm demonstrates optimal search efficiency across all three test environments. Specifically, iteration counts in Environments 1, 2, and 3 are only 61.54, 45.78, and 72.20 respectively, comprehensively outperforming all three comparison algorithms. Relative to RRT, iteration counts are reduced by as much as 85.8%, 89.7%, and 87.1%; relative to RRT-Connect, reductions of 30.3%, 33.7%, and 33.1% are achieved. Regarding planning time, the proposed algorithm also exhibits outstanding and stable performance: overall planning efficiency is improved by 42.1%, 46.7%, and 56.4% versus RRT across the three environments, and reduced by more than 98.0% versus RRT*, which focuses on path optimization. Although the proposed algorithm's planning time is slightly higher than that of RRT-Connect, this marginal millisecond-level time cost is exchanged for shorter path lengths and fewer iterations, achieving a good balance between path quality and search efficiency.

Furthermore, from the perspective of statistical distribution and significance testing, the proposed algorithm also demonstrates extremely high reliability. In terms of algorithmic stability, the standard deviations (SD) of path length under all three test environments are kept within very small ranges, with far lower dispersion than RRT and RRT-Connect, indicating that the proposed optimization strategy possesses strong robustness and consistently produces high-quality paths. Regarding statistical significance, paired t-test results show that the proposed algorithm's improvements over RRT and RRT-Connect on key metrics such as path length and iteration count all reach highly significant levels. Meanwhile, the effect sizes (Cohen's d) comparing these two algorithms range from 2.40 to 5.75, far exceeding the "large effect" threshold of 0.8, statistically confirming the substantive nature of the proposed algorithm's performance gains. In comparison with RRT*, although the effect size on path length is negligible between the two—and no significant difference is observed in the simple environment—the proposed algorithm's reductions in iteration count and planning time both reach highly significant levels. This further demonstrates statistically that, while maintaining path quality equivalent to RRT*'s optimum, the proposed algorithm significantly remedies RRT*'s deficiencies in computational time and search redundancy.

3.1.2. 3D Simulation Experiments and Analysis

To further verify the spatial scalability of the algorithm, a simulation environment was constructed in a 3D configuration space. The map dimensions were set to 10 mm ×

10 mm × 10 mm, with the start point at (0, 0, 0) and the goal point at (10 mm, 10 mm, 10 mm); step size was set to 2 mm, maximum iterations to 10^5, and spherical obstacles were randomly distributed in the 3D space. All other algorithm parameters were kept consistent with the 2D simulations, and each algorithm was independently run 100 times with results averaged.

The search trees and final paths of each algorithm in the 3D environment are shown in Fig. 7, and the quantitative statistics are presented in Table 2. As can be seen from Fig. 7, the proposed algorithm's search tree contains noticeably fewer nodes than those of the other algorithms, and the resulting path is more concise and smooth, demonstrating the effectiveness of the APF resultant-force guidance and path post-processing strategy in 3D space.

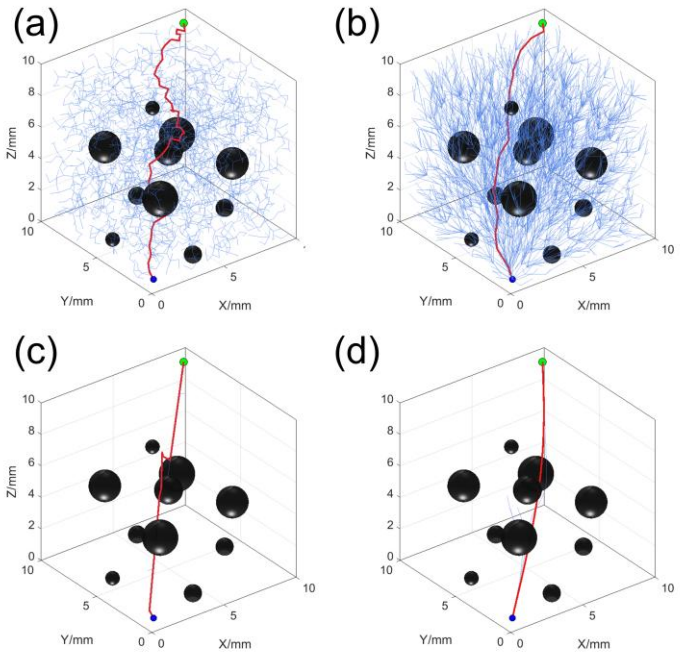


Fig.7 Path planning algorithms in three-dimensional space.

Table 2. Simulation data statistics of each algorithm in 3D environment.

Algorithm	Path length/mm	Cohen's d [95% CI]	Planning time/s	Number of iterations	Significance
RRT	22.96±1.41	5.136[−7.50,−6.89]	0.062±0.036	2330.9±1062.9	***
RRT*	17.37±0.56	2.822[−1.79,−1.42]	0.246±0.166	2560.5±1135.8	***
RRT-Connect	17.59±0.66	2.790[−1.96,−1.70]	0.001±0.000	18.7±7.6	***/***/ns
Improved APF-RRT	15.76±0.08	—	0.016±0.006	17.1±6.2	—

As shown in Table 2, the proposed algorithm achieves the dual advantage of the shortest path length and the fewest iteration counts in the 3D environment. In terms of path length, the proposed algorithm's average planned path length is 15.76 mm, representing reductions of 31.4% versus RRT, 10.4% versus RRT-Connect, and 9.3% versus RRT*. The proposed algorithm surpasses RRT* in path quality because the APF resultant force direction in 3D space can equally suppress invalid expansion of the random tree away from the goal, enabling the path to connect the start and goal configurations more directly, thereby achieving lower path cost.

Regarding iteration count, the proposed algorithm requires only an average of 17.1 iterations to complete planning, representing reductions of 99.3% versus both RRT and RRT*. Notably, RRT* requires an average of 2560.5 iterations, indicating extremely high search redundancy in the current 3D scenario, whereas the proposed algorithm completes planning in fewer than 20 iterations on average, fully validating the search convergence advantage of artificial potential field guidance in high-dimensional spaces. In terms of planning time, the proposed algorithm averages 0.0016 s, representing reductions of 74.2% versus RRT and 93.5% versus RRT*. Although its planning time

is slightly higher than that of RRT-Connect, the proposed algorithm outperforms RRT-Connect in both path quality and iteration efficiency, yielding a more balanced overall performance. The 3D simulation results demonstrate that the proposed algorithm also possesses excellent path-optimization capability and search convergence in high-dimensional spaces, laying an algorithmic foundation for subsequent high-dimensional joint-space planning for robotic arms.

3.2. Robotic Arm Conveying Experiments and Analysis

3.2.1. Experimental Setup

To validate the effectiveness of the proposed algorithm in practical robotic arm path planning tasks, a 6-DOF robotic arm motion-planning simulation environment was constructed on the ROS platform. The Siasun GCR25-1800 collaborative robotic arm was selected, equipped with a wide-opening gripper; the joint motion ranges cover the arm's full workspace, with a repeat positioning accuracy of ±0.05 mm, a maximum reach of 1800 mm, and an end-effector payload capacity of 25 kg. The planning scenario was constructed with reference to farmland conveying operations, and tree-shaped obstacles simulating crop branches were placed within the robotic arm workspace, as shown in Fig. 8. Obstacle pose information is managed uniformly by the ROS parameter server and queried in real time via the FCL collision-detection library.

The proposed algorithm was integrated into the ROS motion-planning framework MoveIt! as an OMPL plugin. MoveIt! provides kinematics solving, inverse kinematics (IK) computation, and mapping between joint space and task space, with real-time

visualization of the planned path and robotic arm motion state in RViz. To comprehensively evaluate the algorithm's performance in a true high-dimensional joint space, the robotic arm's initial joint angle configuration was set to (−126°, 68°, -76°, 54°, 89°, -80°). Each algorithm independently executed 100 planning tasks in this environment, with joint-space trajectory cost and planning time recorded.

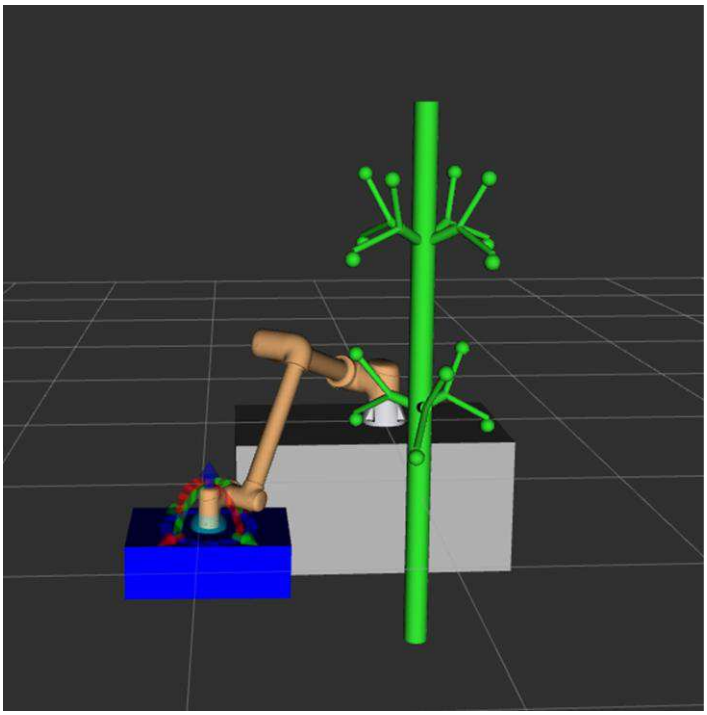


Fig.8 Simulation test transportation environment.

3.2.2. Simulation Results and Analysis

The planned path trajectories of each algorithm in the robotic arm conveying simulation scenario are shown in Fig. 9, where the orange model represents the initial arm pose, the grey model represents the target pose, and the green curve represents the end-effector motion trajectory. The corresponding motion process of the robotic arm under each algorithm is shown in Fig. 10, where the white models represent a series of

arm motion states. The joint angle variation curves during arm operation under each algorithm are shown in Fig. 11, and quantitative performance statistics are presented in Table 3.

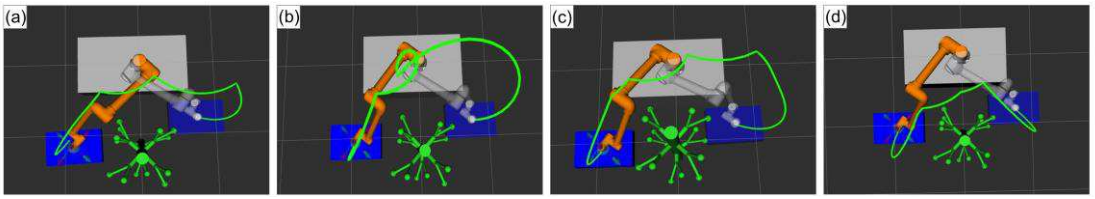


Fig. 9 Simulated path trajectories for each algorithm.

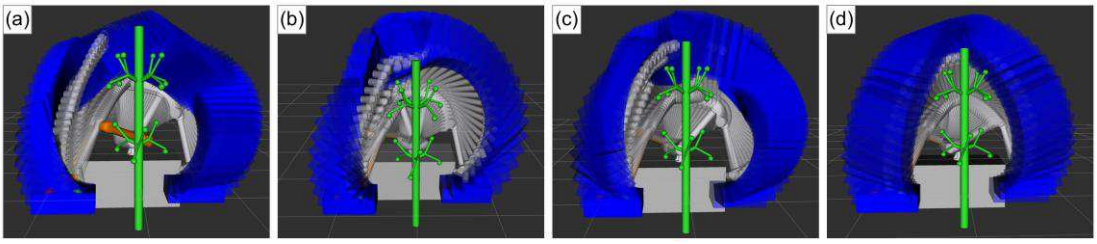


Fig.10 Simulated path planning results of each algorithm.

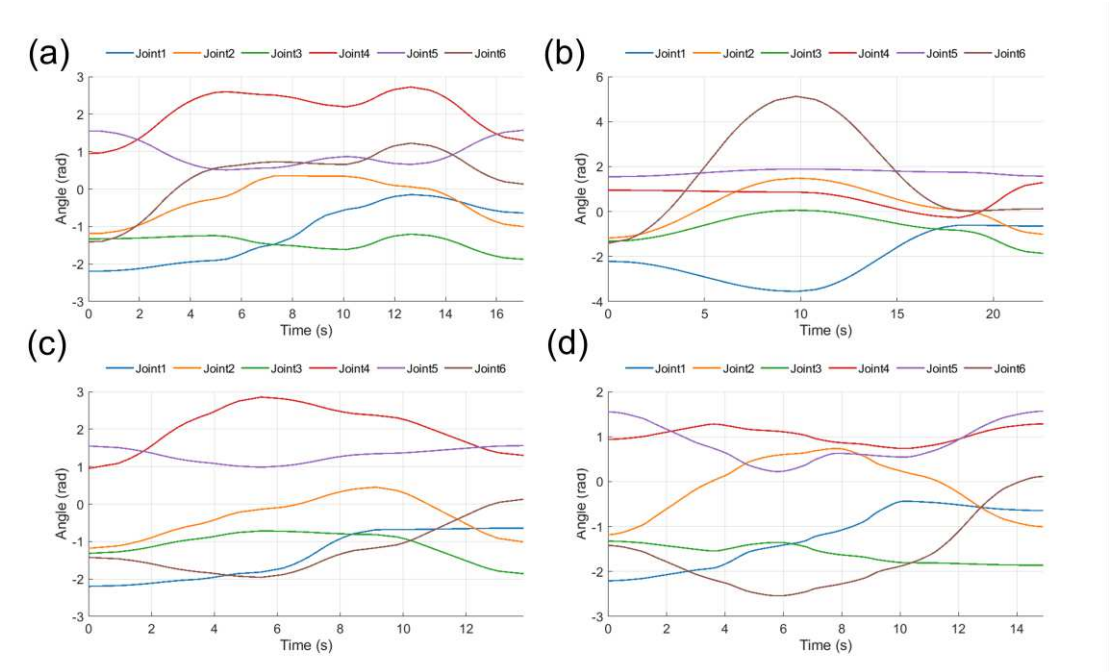


Fig.11 Angle variation curves of each joint of manipulator.

Table 3. Performance statistics of each algorithm in manipulator conveying test.

Algorithm	Path length/rad	Cohen's d [95% CI]	Running time/s	Significance
RRT	9.38±2.36	1.53[−5.09,−0.54]	11.68±2.04	*/**
RRT*	8.54±5.74	0.87[−8.27,2.57]	10.37±4.07	ns
RRT-Connect	10.21±2.12	2.85[−7.50,−3.46]	11.78±2.36	***/*
Improved APF-RRT	5.73±1.17	—	9.32±1.33	—

As shown in Table 3, the proposed algorithm achieves the best results on both path cost and planning time. In terms of path cost, the joint-angle cost of the proposed algorithm's planned path is 5.73 rad, representing reductions of 38.9% versus RRT, 43.9% versus RRT-Connect, and 32.9% versus RRT*. RRT-Connect has the highest path cost (10.21 rad). Although RRT* has a relatively lower average path cost, its standard deviation is large; this is because its rewiring strategy optimizes the cumulative cost function, which converges slowly in high-dimensional joint spaces and cannot sufficiently reduce path cost within a limited number of iterations—reflecting the inherent limitations of purely probabilistically optimal algorithms on high-dimensional problems. The proposed algorithm provides effective heuristic guidance in the high-dimensional joint space through the improved APF, combined with a greedy pruning strategy to directly eliminate redundant joint-angle changes, thereby significantly reducing path cost.

Regarding planning time, the proposed algorithm averages 9.32 s, representing

reductions of 20.2% versus RRT, 20.9% versus RRT-Connect, and 10.1% versus RRT*. Compared with the 2D simulations where planning time differences among algorithms are relatively small, the gap in planning time widens markedly in the 6D joint space; this is directly attributable to the higher frequency of collision checks and the greater computational overhead of inverse kinematics solving in high-dimensional spaces.

Regarding motion smoothness, as shown by the joint angle variation curves in Figure 4, the joint angle motion changes corresponding to the planned path of the algorithm in this paper are gradual, and the joint angle variation curves are continuous without obvious abrupt changes, which meets the execution requirements of smooth movement of the robotic arm and can greatly reduce redundant motion when the robotic arm grasps and transports baskets in different poses.

Integrating the three metrics of path cost, planning time, and motion smoothness, the proposed algorithm demonstrates comprehensive superiority in 6-DOF robotic arm joint-space planning tasks, validating the effectiveness and engineering applicability of the proposed method in real robotic arm motion planning scenarios.

3.2.3. Real-Robot Validation

Building on the foregoing simulation planning results, real-robot validation experiments were conducted on the physical Siasun GCR25-1800 collaborative robotic arm platform. The obstacle avoidance motion process of the robotic arm in the laboratory environment, obtained by sending the joint angle trajectories from the simulation planning to the controller, is shown in Fig. 12. The arm executed the

conveying task along the planned path, and the end-effector gripper successfully completed the entire obstacle-avoidance conveying process. Motion was smooth throughout, with all joint movements continuous and free of jitter, and no collisions with obstacles occurred. The real-robot experimental results are in close agreement with the simulation planning results, further demonstrating the reliability and practicality of the proposed algorithm in unstructured obstacle environments and laying the foundation for subsequent application in real agricultural conveying scenarios.

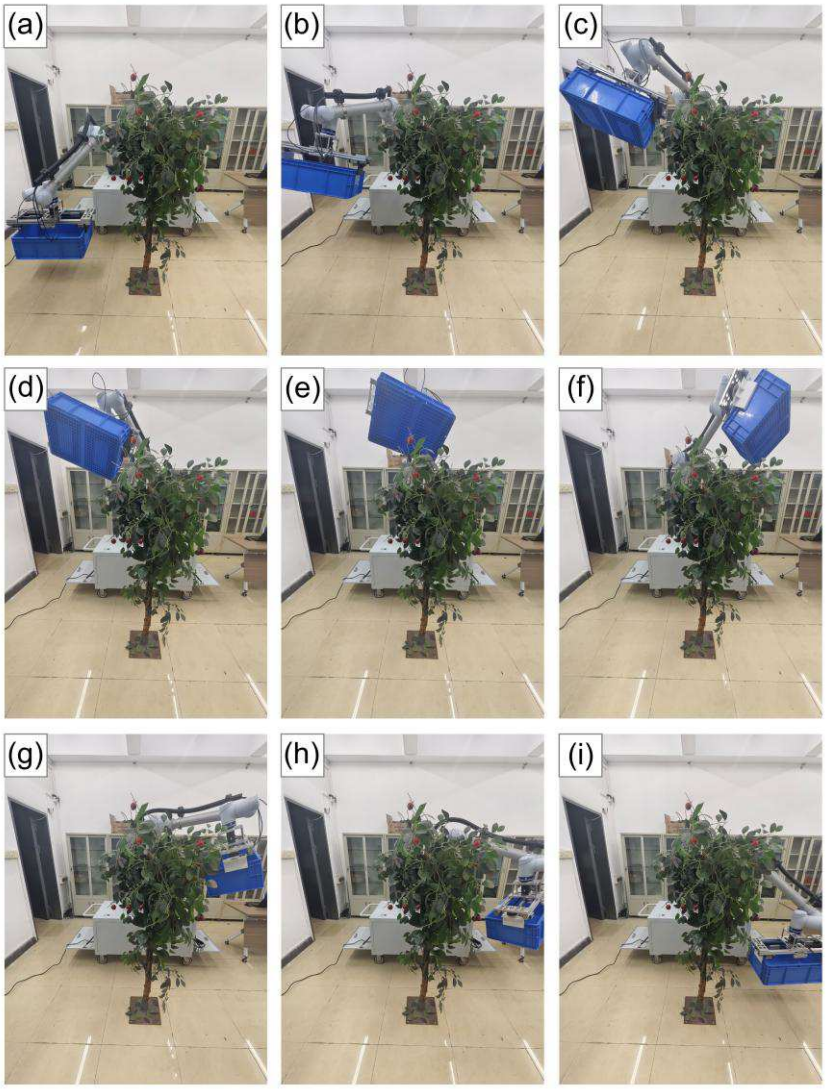


Fig.12 Obstacle avoidance movement of laboratory carrying robot arm.

4. Discussion

4.1. Qualitative Comparative Analysis with Agriculture-Specific Improved Algorithms

Since the source code of agriculture-specific improved algorithms such as IKB-RRT (Ye et al., 2023), BPFPS-RRT (Liu et al., 2023), GD-RRT (Zhao et al., 2026), and AS-RRT (Li et al., 2024b) has not yet been publicly released, direct quantitative comparisons on a unified experimental platform are objectively limited. To this end, this section conducts a qualitative analysis based on the publicly available literature data of each algorithm from two dimensions—design methodology and magnitude of performance improvement—to clarify the positioning of the proposed method within the same category of research.

Comparative analysis of algorithm mechanisms. The proposed algorithm and the above methods differ significantly in their core improvement directions. IKB-RRT (Ye et al., 2023) improves kinematic constraints for bidirectional expansion of orchard mobile robots; its APF guidance focuses on navigation in the external workspace and does not address local-minima escape in the 6D joint space. AS-RRT (Li et al., 2024b) proposes an accumulator-based sampling-point selection strategy and post-collision dynamic step-size adjustment, optimizing planning quality for tea-picking arms; however, its step-size adaptation mechanism is triggered post-collision and lacks proactive escape capability in local-minima regions. BPFPS-RRT (Liu et al., 2023) jointly determines step size via target angle and random values, achieving good results

under fixed obstacle density, but its design does not explicitly handle deep local-trap problems. GD-RRT (Zhao et al., 2026) updates a Gaussian mixture distribution in real time based on collision-detection success frequency, demonstrating environmental adaptability in apple-picking scenarios, but likewise lacks a dedicated escape mechanism. By contrast, the proposed algorithm directly confronts the core challenge of dense local minima in the joint space of conveying scenarios through a systematic mechanism of "multi-potential-field coupling guidance + three-level progressive escape + penalty-weighted distance metric," while simultaneously ensuring path cost and C^2 smoothness through multi-pass greedy pruning and cubic B-spline smoothing—addressing the relatively weak aspects of the aforementioned methods at the design level.

Cross-reference to literature-reported performance data. It should be noted that the following performance data are drawn from the results reported by each algorithm's paper under its respective experimental conditions. Due to differences in test platform, scenario scale, obstacle density, and metric definitions, direct absolute-value comparisons across different sources are not appropriate; the data are provided for trend-reference purposes only.

Regarding path quality improvement: IKB-RRT (Ye et al., 2023) reports a maximum path-length reduction of 13.94% relative to comparison algorithms; AS-RRT (Li et al., 2024b) reports a reduction of 14.18%. The proposed algorithm reduces path length by 18.6%–19.6% versus RRT in 2D scenes, 31.4% in 3D scenes, and path cost by 38.9% versus RRT in the 6D joint space. Given that the conveying task

in this paper operates in a high-dimensional joint space with more complex obstacle geometries, the magnitude of path-cost improvement has a reasonable scenario-specific basis.

Regarding planning efficiency improvement: AS-RRT (Li et al., 2024b) reports planning times below 1 s (tea-garden environment, optimized relative to RRT*). The improved RRT algorithm proposed by Liang et al. integrating dynamic cone constraints and adaptive step size (Liang et al., 2026) reports planning-time reductions of 84.76% (2D) and 73.03% (3D), and path-length reductions of 20.13% (2D) and 25.20% (3D), relative to standard RRT. The proposed algorithm reduces planning time by 42.1%–56.4% versus RRT in 2D scenes, by 74.2% in 3D scenes. The absolute difference in planning time is primarily determined by the experimental platform and spatial dimensionality; the magnitudes of improvement achieved by the proposed algorithm in the high-dimensional robotic arm joint space are in the same order as those reported by comparable methods in the literature.

In aggregate, the proposed algorithm holds a clear advantage over existing agriculture-specific methods in terms of mechanism completeness and adaptability to high-dimensional joint spaces. Direct quantitative comparison requires code reproduction on a unified platform, which constitutes one of the primary limitations of this paper; future work will conduct systematic comparative validation on open benchmark platforms.

4.2. Analysis of Algorithm Limitations

The proposed algorithm demonstrates good performance in both simulation and real-robot experiments under laboratory conditions; however, its engineering deployment still faces the following limitations that need to be addressed in subsequent research.

(1) Limitation of the static-obstacle assumption. The current algorithm requires obstacle pose information to be perceived by sensors and pre-modeled before planning. In real farmland, crop branches may sway randomly due to wind, and personnel may enter the robotic arm workspace, creating unpredictable dynamic obstacles. The proposed method does not incorporate an online re-planning mechanism and therefore cannot respond to real-time obstacle changes during plan execution. This limitation means the current method is better suited to semi-structured, relatively static agricultural conveying scenarios rather than fully open dynamic field environments.

(2) Perception dependency and deployment cost. The algorithm relies on precise 3D position and shape information of obstacles as input. In actual farmland, constructing high-accuracy obstacle models requires additional LiDAR sensors as well as the corresponding calibration and registration procedures, introducing additional hardware costs and system complexity.

(3) Engineering transfer gap from laboratory to field. The current real-robot experiments were conducted in a controlled laboratory environment, and the obstacles in the scene were fixed simulated-branch models. Real farmland also involves disturbance factors not addressed in this paper, including illumination variation (direct sunlight causing perception errors), dust and mud (affecting sensor accuracy),

mechanical vibration (affecting control precision), and temperature changes (affecting actuator performance). The combined effects of these factors need to be systematically evaluated in subsequent field trials.

5. Conclusions

(1) An RRT robotic arm path planning algorithm integrating an improved Artificial Potential Field method is proposed. By deeply embedding the improved APF into the RRT node-expansion mechanism and constructing a power-law decay repulsive gain coefficient, the inherent goal-unreachability defect of conventional potential-field methods is effectively overcome. The designed three-level progressive escape mechanism and penalty-weighted distance metric endow the random tree with the ability to actively bypass local deadlock regions. Simultaneously, an environment-aware adaptive step-size strategy and a dynamic goal-bias probability strategy achieve a self-adaptive balance between global spatial exploration and local goal convergence, supported by multi-pass greedy pruning and cubic B-spline smoothing to ensure low path cost and C^2 continuity at the algorithmic level.

(2) Simulation results in 2D and 3D configuration spaces demonstrate that the improved algorithm possesses significant search advantages in complex multi-constraint environments. In three representative 2D scenarios covering sparse obstacles, narrow corridors, and complex dense distributions, the improved algorithm reduces average iteration counts by 85.8%–89.7% and planning time by 42.1%–56.4% relative to conventional RRT. In the 3D optimization tests, the algorithm completes

reliable planning in an average of only 17.1 iterations, with path length shortened by 31.4% versus conventional RRT, validating the strong robustness and high execution efficiency of the multi-field coupling guidance mechanism in high-dimensional spaces.

(3) Taking the collaborative robotic arm joint-space simulation and real-robot grasping experiments as examples, results show that, compared to conventional RRT, RRT-Connect, and RRT*, the improved algorithm significantly reduces joint-space path cost by 38.9%, 43.9%, and 32.9%, respectively, and shortens planning time by 10.1%–20.9%. Joint angular velocity and acceleration variations are smooth and continuous throughout; the real-robot obstacle-avoidance grasping measurements are in close agreement with the simulation analysis, validating the algorithm's good planning responsiveness and motion stability under laboratory conditions and providing a feasible algorithmic foundation for robotic arm conveying operations in complex agricultural environments.

References

Ab Wahab, M. N., Nazir, A., Khalil, A., et al. (2024). Improved genetic algorithm for mobile robot path planning in static environments. Expert Systems with Applications, 249, 123762. https://doi.org/10.1016/j.eswa.2024.123762

Duan, M., Zheng, Z., Gao, Q., et al. (2026). An improved path planning method for RRT spreading robotic arm in refractory brick masonry. Journal of Mechanical Transmission. https://link.cnki.net/urlid/41.1129.TH.20260414.0925.002

Gammell, J. D., Srinivasa, S. S., & Barfoot, T. D. (2014). Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2997–3004). IEEE.

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 4(2), 100–107.

Jeong, I. B., Lee, S. J., & Kim, J. H. (2019). Quick-RRT*: Triangular inequality-based implementation of RRT* with improved initial solution and convergence rate. Expert Systems with Applications, 123, 82–90. https://doi.org/10.1016/j.eswa.2018.12.029

Jin, T., & Han, X. (2024). Robotic arms in precision agriculture: A comprehensive review of the technologies, applications, challenges, and future prospects. Computers and Electronics in Agriculture, 221, 108938. https://doi.org/10.1016/j.compag.2024.108938

Kang, M., Chen, Q., Fan, Z., et al. (2024). A RRT based path planning scheme for multi-DOF robots in unstructured environments. Computers and Electronics in Agriculture, 218, 108707. https://doi.org/10.1016/j.compag.2024.108707

Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research, 30(7), 846–894. https://doi.org/10.1177/0278364911406761

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research, 5(1), 90–98.

Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. In Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA) (pp. 995–1001). IEEE.

Lai, L., Yu, H., Li, S., et al. (2025). A novel hybrid RRT and metaheuristic algorithm for path planning in complex environments. Expert Systems with Applications.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning (Technical Report TR 98-11). Iowa State University.

Li, X., Yang, J., Wang, X., et al. (2024). Adaptive step RRT*-based method for path planning of tea-picking robotic arm. Sensors, 24(23), 7759. https://doi.org/10.3390/s24237759

Liang, X., Yu, F., Gao, W., et al. (2026). An improved RRT-based path planning approach with dynamic cone angle guidance for robotic manipulator obstacle avoidance. Intelligent Service Robotics, 19, 21. https://doi.org/10.1007/s11370-025-00688-w

Liu, L., Yang, F., Liu, X., et al. (2024). A review of the current status and common key technologies for agricultural field robots. Computers and Electronics in Agriculture, 227, 109630. https://doi.org/10.1016/j.compag.2024.109630

Liu, Y., Tao, W., Li, S., et al. (2023). A path planning method with a bidirectional potential field probabilistic step size RRT for a dual manipulator. Sensors, 23(11), 5172. https://doi.org/10.3390/s23115172

Lozano-Pérez, T., & Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. Communications of the ACM, 22(10), 560–570.

Qureshi, A. H., Miao, Y., Simeonov, A., et al. (2020). Motion planning networks: Bridging the gap between learning and planning. IEEE Robotics and Automation Letters, 5(2), 2022–2029.

Wang, X., et al. (2022). AEB-RRT*: An adaptive extension bidirectional RRT* algorithm. Autonomous Robots, 46(6), 685–704. https://doi.org/10.1007/s10514-022-10049-4

Wang, Y., Jiang, W. S., Luo, Z., et al. (2024). Path planning of a 6-DOF measuring robot with a direction guidance RRT method. Expert Systems with Applications, 238, 122057. https://doi.org/10.1016/j.eswa.2023.122057

Ye, L., Wu, F., Zou, X., et al. (2023). Path planning for mobile robots in unstructured orchard environments: An improved kinematically constrained bi-directional RRT approach. Computers and Electronics in Agriculture, 215, 108453. https://doi.org/10.1016/j.compag.2023.108453

826 Zhang, B., Xia, Y., Gu, Y., et al. (2026). Application of improved RRT algorithm by
827 multi-strategy fusion in path planning for robotic manipulator: A case study of
828 multi-posture dragon fruit picking. Computers and Electronics in Agriculture,
829 242, 111285. https://doi.org/10.1016/j.compag.2025.111285
830 Zhao, X., Wang, K., Zhang, P., et al. (2026). An improved RRT* path planning
831 algorithm combining Gaussian distributed sampling and depth strategy for
832 robotic arm of fruit-picking robot. Computers and Electronics in Agriculture, 240,
833 111244. https://doi.org/10.1016/j.compag.2025.111244
834