

Article

NeuroJPS-A: Neural Jump Point Search with Adaptive Potential Fields for UAV Path Planning and Obstacle Avoidance in Orchard Environments

Beibei Cui *, Mingyang Wang, Pengpeng Dong, Lei Zhang , Kunpeng Zhang and Liang Zhao 

College of Electrical Engineering, Henan University of Technology, Zhengzhou 450001, China; mingyang037@163.com (M.W.); harmana007@outlook.com (P.D.); zhanglei@haut.edu.cn (L.Z.); kp.zhang@outlook.com (K.Z.); zhaoliang_270@163.com (L.Z.)

* Correspondence: beibei.cui@haut.edu.cn

Highlights

What are the main findings?

- NeuroJPS-A reduces 3D path length by 24.9% versus A*, execution time to 0.120 s ($4.5\times$ faster), and memory usage to 85.6 KB, while cutting the 2D turn count by 47.8%.
- Integrating obstacle-aware Neural Combinatorial Optimization with potential-field-guided 3D-JPS and path-guided adaptive APF eliminates local minima and tightly couples task sequencing, global search, and reactive avoidance.

What are the implications of the main findings?

- The framework enables real-time, energy-efficient autonomous navigation onboard resource-limited UAVs in semi-structured orchards, directly supporting precision tasks such as monitoring and spraying.
- The closed-loop fusion of task planning, safe path search, and reactive control establishes a robust design paradigm for UAV operations in complex, obstacle-dense agricultural environments.

Abstract

With the continuous expansion of unmanned aerial vehicle (UAV) applications, generating near-optimal paths and achieving effective obstacle avoidance in complex environments remain highly challenging tasks. To address the problems of multi-objective path planning and obstacle detection for UAV flight missions in orchard environments, this paper proposes a novel hybrid algorithmic framework named NeuroJPS-A. The main scientific contribution is the synergistic integration of neural combinatorial optimization, 3D-JPS, and adaptive APF, enabling task-aware obstacle avoidance and closed-loop trajectory adjustment. This method introduces neural combinatorial optimization from the TSP into the 3D-JPS algorithm, optimizing the search mechanism of the traditional JPS and further shortening the UAV's globally planned path length. In addition, this study integrates the proposed algorithm with the APF to solve the local dynamic obstacle avoidance problem. Quantitative results show that NeuroJPS-A reduces path length by 10% and the number of turns by 47.8% in 2D, and achieves a 24.9% shorter path and 22% of A*'s computation time in 3D. To verify the performance of the proposed method, comprehensive simulation experiments were conducted. The experimental results demonstrate that the NeuroJPS-A algorithm enables UAVs to quickly and effectively generate optimal planned routes, ensuring safe navigation in complex orchard environments and preventing collisions during flight missions.



Academic Editors: Salvatore Rosario Bassolillo, Vito Antonio Nardi and Giuseppe Tricomi

Received: 6 May 2026

Revised: 26 June 2026

Accepted: 29 June 2026

Published: 2 July 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: NeuroJPS-A; UAV path planning; obstacle avoidance; neural combinatorial optimization; 3D Jump Point Search; adaptive potential field; orchard environments

1. Introduction

With the rapid development of precision agriculture, unmanned aerial vehicles (UAVs) have been increasingly used in orchard operations, becoming an important tool for improving orchard management efficiency [1]. The orchard environment exhibits typical semi-structured characteristics: on the one hand, there are regularly arranged rows of fruit trees forming navigable corridors; on the other hand, the interlaced canopy, randomly growing branches and leaves, coupled with the appearance of dynamic obstacles such as agricultural machinery and operators, pose severe challenges to the autonomous navigation of UAVs. In complex orchard environments, the core technical issues of autonomous UAV flight can be summarized into three aspects: firstly, the ability to quickly detect various obstacles; secondly, the accurate establishment of obstacle coordinates in three-dimensional (3D) space; and finally, based on environmental perception information, the UAV must be able to plan a safe, smooth, and efficient flight path in real time, effectively avoiding unknown obstacles while completing the traversal of multiple mission points [2].

Currently, UAV path planning research is mainly divided into two categories: global path planning and local path planning. Global path planning requires complete prior knowledge of the environmental information to generate the optimal or sub-optimal path from the start point to the destination [3]. Typical algorithms include the A* algorithm [4], genetic algorithms [5], particle swarm optimization [6], and ant colony optimization [7]. Local path planning focuses on perceiving environmental changes in real time and dynamically adjusting the flight trajectory. Representative methods include the Rapidly exploring Random Tree (RRT) [8], Artificial Potential Field (APF) [9], and fuzzy logic [10]. However, traditional global planning algorithms exhibit poor adaptability to dynamic environmental changes, whereas local planning methods are prone to falling into local minima, making it difficult to guarantee the global efficiency of the mission [11]. Although researchers are continuously improving algorithms to obtain globally optimal paths, this often leads to a sharp increase in computational complexity, thereby reducing the overall operational efficiency of the system [12].

For orchard operation scenarios, UAVs usually need to traverse multiple mission points (e.g., fruit tree positioning, disease sample collection points, etc.), which is essentially a Traveling Salesman Problem (TSP). At the same time, during flight between rows of fruit trees, UAVs must avoid randomly appearing obstacles. Therefore, how to organically integrate task sequence optimization, global path search, and real-time obstacle avoidance capabilities to form a collaborative and efficient path planning framework is the core scientific problem of this study.

Accurate environmental perception is the foundation of autonomous UAV navigation. Currently, UAVs are usually equipped with various sensors for environmental detection, such as LiDAR, ultrasonic sensors, monocular cameras, and binocular cameras [13]. Among them, LiDAR obtains environmental point clouds through high-frequency scanning and uses changes in spatial relationships for correction to achieve obstacle avoidance [14]. Monocular camera schemes are low-cost, but due to limited depth information, they are prone to missed detections of obstacles, leading to a decline in the reliability of the navigation system [15]. In contrast, binocular stereo vision is considered a more robust obstacle avoidance method. Its main advantage lies in its ability to accurately detect objects in the flight direction and provide reliable absolute distance estimation [16].

In known static environments, a variety of global path planning algorithms have been extensively studied. As a classic heuristic search algorithm, the A* algorithm guarantees finding the shortest path; however, its computational cost increases significantly when the search space is large [17]. Ant colony optimization algorithms find the optimal path by simulating the pheromone-laying behavior of ants, but such algorithms still face major challenges in terms of adaptability to dynamic environments and computational requirements [18].

In terms of local obstacle avoidance, the APF method guides the UAV towards the target while avoiding collisions by constructing virtual attractive and repulsive fields. However, its inherent problems, such as local minima and targets being unreachable, limit its application [19]. Although studies have shown that introducing virtual guiding points into the APF can help UAVs avoid local minima and path oscillation, this also increases the algorithm's complexity and computational load [20]. In recent years, neural networks and reinforcement learning methods have attracted much attention. They can achieve accurate and efficient decision-making in complex situations, but they often rely on massive amounts of training data and huge computing power support [21].

Planning an optimal route that satisfies multiple objectives while reducing various computational and hardware costs is highly challenging [22], and this difficulty is closely related to the TSP. The TSP is a classic combinatorial optimization problem in path planning [23]. Traditional solution methods (such as heuristic search algorithms) have certain limitations when dealing with such problems, especially lacking the capability for real-time dynamic adjustment [24]. In recent years, Neural Combinatorial Optimization (NCO) has gradually become a research hotspot in academia. This method utilizes neural network models and reinforcement learning to solve a series of combinatorial optimization problems [25]. It can quickly output high-quality solutions during the testing phase, demonstrating broad application prospects. However, NCO methods still face significant deployment challenges in real-world industrial applications, including a strong dependence on training data, limited generalization ability, edge cost definitions that are disconnected from obstacle distributions, and a lack of decision interpretability. To address these drawbacks, this study introduces potential-field-aware edge costs into the NCO layer and compensates for its insufficient environmental adaptability through global-local closed-loop synergy, thereby advancing NCO from the laboratory toward engineering applications.

The limitations of a single algorithm have prompted researchers to continuously explore hybrid path planning methods. Common approaches include combining global planning with local planning (e.g., the fusion of the A* algorithm and the DWA algorithm [26]), as well as combining geometric methods with intelligent optimization algorithms. To some extent, these hybrid methods balance global optimality and local real-time performance. However, they generally suffer from drawbacks such as loose coupling between modules and insufficient information interaction, making it difficult to achieve true collaborative optimization.

Based on a comprehensive analysis of current domestic and international research, UAV path planning oriented toward orchard environments currently faces four main problems: (1) Separation between global planning and local obstacle avoidance: Existing hybrid methods mostly adopt a hierarchical architecture, where the local obstacle avoidance module runs independently after the global planning module generates a reference path. This loose coupling leads to the global path not fully considering the feasibility of local obstacle avoidance, while the displacement adjustments of local obstacle avoidance can easily deviate from the globally optimal path, lacking an effective closed-loop feedback mechanism between the two. (2) Separation of multi-mission point sequence optimization

and path planning: In multi-mission point operation scenarios in orchards, the order of task visits directly determines the length of the total flight path. However, existing research usually treats TSP solving and specific path planning as two independent stages. The TSP often calculates node costs based solely on simple Euclidean distances, ignoring the existence of feasible paths in the actual 3D space and the impact of obstacle distribution on real path costs. (3) Insufficient adaptability to dynamic obstacles: Dynamic obstacles such as agricultural machinery and operators frequently appear in orchard environments. Traditional methods like APF have fixed parameters, respond slowly to dynamic obstacles, and are difficult to adapt to complex and changing environments; while deep learning-based methods, although promising, rely highly on annotated data and long training times, making it difficult to guarantee real-time onboard computation. (4) Shallow level of algorithmic fusion: Current hybrid algorithms are mostly simple stacking of functional modules, lacking deep collaborative mechanism design. How to enable the three core modules—TSP solving, global path search, and local obstacle avoidance—to share information and enhance each other to form an organic whole is the key to breaking through the performance bottlenecks of existing algorithms.

To address the above problems and challenges, this paper proposes a novel hybrid path planning framework named NeuroJPS-A. This framework deeply integrates NCO for TSP task sequencing, the 3D Jump Point Search (3D-JPS) algorithm for efficient global path searching, and the APF for real-time local obstacle avoidance and trajectory fine-tuning, truly realizing the collaborative operation of global path optimization and local real-time obstacle avoidance. The main research contents and innovations of this paper include:

1. Task Sequence Optimization: APF-informed NCO (Innovation 1)

A neural combinatorial optimizer based on a Graph Attention Network is designed to learn the implicit structural features between orchard task points, enabling fast inference while obtaining a low-cost optimal visit sequence for subsequent global path planning. The core innovation lies in integrating artificial potential field information into the edge cost c_{ij} of the NCO (i.e., APF-informed NCO), so that task sequencing proactively accounts for obstacle density, thereby minimizing downstream collision-avoidance costs at the source.

2. Fast Global Path Search: Potential-aware 3D-JPS (Innovation 2)

After obtaining the optimal visit sequence of task points, the 3D-JPS algorithm is adopted to quickly generate an initial global path in the semi-structured orchard environment. The 3D-JPS fully utilizes the symmetry and jump point characteristics of the 3D grid map, substantially reducing redundant search nodes. The core innovation is incorporating the APF into the JPS edge cost (i.e., potential-aware JPS), which prioritizes the selection of “low-potential” (i.e., clearer) corridors during global search, thereby reducing the need for later local corrections.

3. Real-time Obstacle Avoidance: Path-guided APF Integration (Innovation 3)

An improved adaptive potential field method is introduced for local real-time obstacle avoidance. It dynamically adjusts key parameters such as the attractive and repulsive coefficients by fusing the output of the neural optimizer with real-time perception information from onboard sensors, constructing an efficient closed-loop feedback mechanism. The core innovation is introducing a path-guidance potential term U_{guide} hybridized with the classical APF (i.e., path-guided APF integration), which mitigates the local minima problem inherent to traditional APF and maintains global trajectory consistency by anchoring the local controller to the JPS global path.

4. Deep Fusion of Multiple Algorithms and Multi-UAV Deconfliction (Innovation 4 and Framework Synergy)

The core advantage of NeuroJPS-A lies in the deep coupling among its modules: the output of the NCO not only determines the task sequence but is also used to initialize the parameter preferences of the APF; simultaneously, the global path generated by 3D-JPS provides an accurate guiding potential field for the APF; during local obstacle avoidance, the dynamic adjustment information of the APF is fed back to the NCO in real time for online fine-tuning of subsequent unexecuted tasks. Furthermore, another core innovation is resolving inter-UAV conflicts using local occupancy maps and incremental JPS replanning (i.e., efficient multi-UAV deconfliction), bypassing the computational overhead of global recalculations.

The proposed NeuroJPS-A framework is not a simple cascade of independent modules but a co-designed system in which each core algorithm is adapted for cross-layer synergy. The NCO module incorporates an APF-aware edge cost, the 3D-JPS planner uses a potential-field correction in node expansion, the APF local controller adds a path-guidance potential anchored to the JPS trajectory, and multi-UAV conflicts are resolved by segment-wise replanning. These components interact via bi-directional information: NCO output influences APF parameters, the global JPS path guides local control, and online APF adjustments are fed back to refine the remaining tasks. This closed-loop design differs from earlier hybrid approaches (e.g., A^* +DWA or A^* +APF without feedback), which typically connect global and local planners in a one-way, loose manner and lack correction of local deviations from the global optimum. By maintaining continuous alignment between planning layers, NeuroJPS-A can achieve a more balanced trade-off among path optimality, computational efficiency, and reactive safety.

These four innovations work synergistically: the NCO provides an obstacle-aware task sequence, the JPS generates a low-risk global path, and the APF executes locally adaptive adjustments while receiving closed-loop feedback. This design represents a departure from hybrid methods that merely stack independent modules. The following section presents experimental evaluations of this integrated framework.

The proposed collaborative framework of NeuroJPS-A is illustrated in Figure 1. The framework first establishes a typical orchard environment model (Orchard Environment Modeling) and then deeply integrates NCO, 3D-JPS, and the APF method. This deep integration breaks the module fragmentation inherent in traditional hierarchical architectures, achieving an information closed loop and collaborative optimization. The specific innovations and advantages of the framework are reflected in the following aspects:

First, a parameter adaptive mechanism is designed. The system dynamically adjusts the key parameters of the APF based on the output of the Task Allocation and NCO Solver and real-time environmental perception information, endowing the local obstacle avoidance behavior with both task awareness and environmental adaptability.

Second, a closed-loop feedback path is constructed. The state information from the local obstacle avoidance process is fed back to the global planning layer in real time. This mechanism enables the algorithm with online learning and adaptive adjustment capabilities, thereby significantly enhancing its robustness in dynamic and changing environments.

Third, specific optimizations for orchard scenarios are implemented. Fully considering the semi-structured characteristics of the orchard environment, prior knowledge of the fruit tree row directions is incorporated into the 3D-JPS Global Planner, and a dedicated canopy avoidance strategy is designed within the APF Local Controller. These optimizations greatly improve the applicability and safety of the algorithm in real-world orchard operation scenarios.

Finally, through the Multi-UAV Coordinator, seamless cooperation and efficient information exchange among the core modules are achieved, ultimately establishing a highly efficient and reliable collaborative path planning system for UAVs.

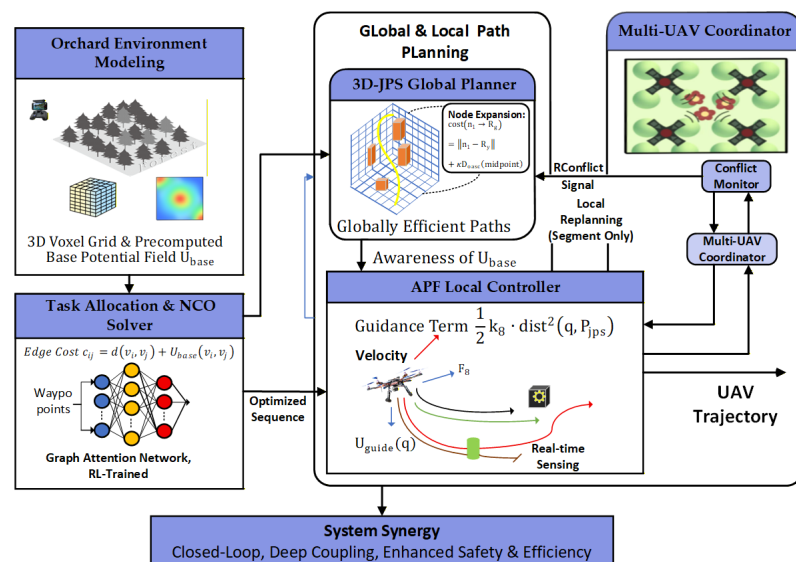


Figure 1. NeuroJPS-A framework.

The detailed technical description is provided in Section 2.

2. Materials and Methods

2.1. Network Framework

The core objective of UAV path planning is to generate a safe and optimal flight trajectory while satisfying the conditions for dynamic threat avoidance [27]. To accomplish UAV flight missions in complex backgrounds, this paper proposes a hybrid path planning framework oriented towards orchard environments, named NeuroJPS-A. This framework effectively balances global task optimality, search efficiency, and online obstacle avoidance capabilities, making it particularly suitable for structured yet obstacle-dense orchard scenarios and multi-UAV collaborative tasks.

As shown in Figure 2, the NeuroJPS-A framework is primarily composed of the following five core functional modules: (1) Orchard Environment Modeling Module (Orchard Map): This module generates a 3D voxel grid based on geographic information or pre-mapped data. The trunks and canopies of fruit trees are rasterized separately and subjected to a safety buffer (inflation operation), ultimately outputting an occupancy grid map with safety margins and a basic APF potential field. (2) Task Allocation and NCO Module (Task Allocation and NCO): Target points are first clustered or regionally partitioned to achieve load balancing among multiple UAVs. Subsequently, NCO is utilized at either the individual UAV or global level to generate the optimal visit sequence. The NCO uses a weighted edge cost incorporating APF components as the optimization objective for both training and inference. (3) Global Path Planning Module (3D-JPS Global Planner): For the task sequence output by the NCO, the 3D-JPS algorithm, incorporating APF cost corrections, is employed segment by segment to calculate discrete waypoint paths in 3D space. (4) Local Obstacle Avoidance and Execution Module (APF Local Controller): During actual flight, an online APF guided by the global path is used to generate real-time velocity and position correction commands. If a trajectory conflict with another UAV is detected, a local replanning mechanism is triggered (calling 3D-JPS only for the conflicting segment and temporarily marking the conflicting grid as “occupied”). (5) Coordination and Monitoring Module (Multi-UAV Coordinator): This module is responsible for the temporal simulation and conflict detection of the multi-UAV system, monitoring task completion status, and the dynamic switching of control strategies (e.g., increasing safety distance parameters in narrow areas to make the system behave more conservatively).

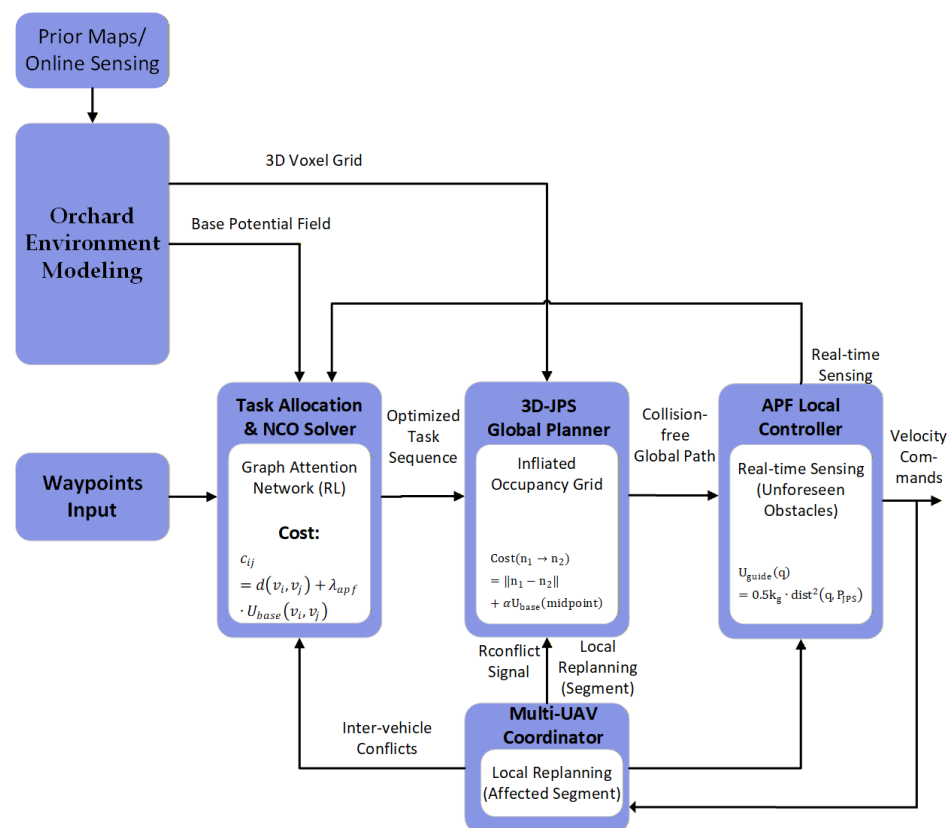


Figure 2. Structure of the NeuroJPS-A model.

2.2. Neural Combinatorial Optimization for TSP

Neural Network Architecture and Training. The neural model for task sequencing employs a graph attention network (GAT) as the encoder, consisting of 3 attention heads, 2 layers, and an embedding dimension of 128, followed by a pointer network decoder; no alternative backbone such as Transformer or generic GNN is used. The input to the model is a set of 3D waypoint coordinates (optionally augmented with task weights such as priorities or time windows), and the output is a permutation of node indices representing the visiting order. For training, we adopt the REINFORCE algorithm with a greedy rollout baseline to reduce variance; the baseline is updated every 20 epochs using an exponential moving average with a decay factor of 0.9, and reward normalization is applied per mini-batch. The learning rate is set to 1×10^{-4} with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and no weight decay; the batch size is 64, and the network is trained for up to 200 epochs, with early stopping triggered if the validation reward does not improve for 20 consecutive epochs. Training data consist of 1,000,000 randomly generated instances, each containing 10 to 30 waypoints sampled within the orchard grid, with 95% used for training and 5% for validation. For each sampled waypoint set, the corresponding APF is pre-computed from the occupancy map, and the edge costs used in the loss function are defined as in Equation (3). Both the training and validation reward/loss curves are recorded at each epoch for monitoring, and the final model checkpoint is selected based on the highest average reward on the validation set. This configuration enables the model to learn obstacle-aware task sequencing that minimizes the expected weighted path cost. The total training time is approximately 6 h on a single NVIDIA RTX 4090 GPU.

NCO can use deep learning to solve the TSP. Currently, the more advanced TSP methods, using the neural combinatorial optimization pipeline method [28], have given this paper great inspiration.

The neural model is perfected in the following three steps, as shown in Figure 3. In the orchard inspection task, a set of 3D checkpoints (waypoints) needs to be visited. Let the target point set be:

$$V = \{v_1, v_2, \dots, v_n\} \quad (1)$$

where each $v_{\{i\}}$ represents spatial coordinates. The objective of the TSP is to find a permutation π that minimizes the total path length:

$$\min_{\pi} L(\pi) = \sum_{i=1}^{n-1} d(v_{\pi(i)}, v_{\pi(i+1)}) \quad (2)$$

where $d(\cdot, \cdot)$ is the Euclidean distance.

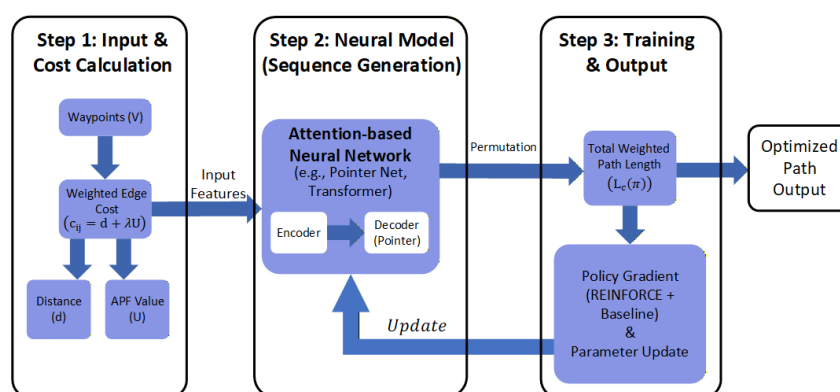


Figure 3. Structure of the Neural model.

Based on the APF-weighted edge cost, in order to improve the sorting quality sensitive to obstacles (so that the path does not frequently cross in dense obstacle areas), this study uses the weighted edge cost as the training and inference target in NCO. The edge cost is defined as:

$$c_{ij} = d(v_i, v_j) + \lambda_{\text{apf}} U(v_i, v_j) \quad (3)$$

where $U(v_i, v_j)$ represents the average potential field value (APF value) at the midpoint of the connecting line or sampling points, and $\lambda_{\text{apf}} > 0$ is the weight coefficient. The sequence trained in this way can avoid high-potential field areas at the global level, reducing the subsequent path correction overhead—this is one of the important innovations of this work to traditional NCO. The neural model adopts an attention mechanism-based sequence generator, expressed as:

$$\pi = f_{\theta}(V) \quad (4)$$

The training objective is to minimize the expected cost:

$$\min_{\theta} \mathbb{E}_{\pi \sim_{\theta}(\cdot|V)} [L_c(\pi)], \quad L_c(\pi) = \sum_i c_{\pi(i), \pi(i+1)} \quad (5)$$

The training uses policy gradient (REINFORCE) or baseline-controlled policy gradient variants, and the gradient estimation is:

$$\nabla_{\theta} \mathbb{E}_{\pi} [L_c(\pi)] \approx \mathbb{E}_{\pi} [(L_c(\pi) - b) \nabla_{\theta} \log p_{\theta}(\pi|V)] \quad (6)$$

where b is the baseline used for variance suppression. Module input/output (I/O): the input is a node set V (which can include location information, task weights, time windows, etc.), and the output is the visit sequence π for each UAV. In the multi-UAV scenario, the initial task allocation can use regional division or clustering, after which the NCO is run separately for each UAV or jointly before decoupling the allocation.

2.3. 3D Jump Point Search (3D-JPS) for Global Path Planning

The environment is represented by a 3D grid map (voxel map):

$$G = (N, E) \quad (7)$$

where N is the set of voxels and represents the adjacency relationship (up to 26-neighborhood). Traditional A* expands a large number of nodes on large voxel grids. This study adopts 3D-JPS, using “jumping” and “forced neighbor” rules to prune symmetrical nodes, thereby significantly reducing the number of expansions.

The evaluation function of A* remains:

$$f(n) = g(n) + h(n) \quad (8)$$

where $g(n)$ is the actual cost from the starting point to node n , and the heuristic function $h(n)$ takes the 3D Euclidean distance:

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2 + (z_n - z_g)^2} \quad (9)$$

The jump operation definition (recursive) of JPS is:

$$JP(n, d) = \begin{cases} n, & \text{if } n = \text{goal} \\ n, & \text{if forced_neighbor}(n, d) = \text{true} \\ JP(n + d, d), & \text{otherwise} \end{cases} \quad (10)$$

where d is the direction vector (one of 26 possible directions), identifying forced points that must be expanded due to obstacles.

In order to penalize high-risk corridors in advance during the global search stage, this study introduces APF into JPS cost evaluation and extends the voxel cost to:

$$\cos t(n_1 \rightarrow n_2) = \|n_1 - n_2\| + \alpha U\left(\frac{n_1 + n_2}{2}\right) \quad (11)$$

where $U(\cdot)$ is the APF value applied to the voxel, and α is the weighting coefficient. This modification causes JPS to prefer traversing low-potential field corridors when selecting jump directions and sorting, thereby reducing the frequency of subsequent local obstacle avoidance adjustments. I/O: The input is the start point, end point, and occupancy grid with safety buffers; the output is a discrete sequence of path nodes. JPS is particularly efficient under the regular row-and-column structure of orchards (inter-row corridors will form long jump segments), which is the engineering advantage of this work in combining scene geometry.

Since the APF cost breaks the grid uniformity, the modified JPS no longer guarantees global optimality. However, as a risk-aware heuristic search, it can still quickly identify low-potential corridors, significantly reducing the overhead of subsequent local corrections.

2.4. Artificial Potential Field (APF) Is Used for Local Obstacle Avoidance and Trajectory Fine-Tuning

Although 3D-JPS provides a globally feasible path, unexpected obstacles or interactions with other UAVs still occur in real-world scenarios, requiring online fine-tuning. APF is adopted as a low-cost local obstacle avoidance strategy. The basic configuration of APF is:

Attractive potential:

$$U_{\text{att}}(q) = \frac{1}{2} k_{\text{att}} \|q - q_g\|^2 \quad (12)$$

Repulsive potential:

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2} k_{\text{rep}} \left(\frac{1}{d(q, q_o)} - \frac{1}{d_0} \right)^2, & d(q, q_o) \leq d_0 \\ 0, & d(q, q_o) > d_0 \end{cases} \quad (13)$$

Total potential field:

$$U(q) = U_{\text{att}}(q) + \sum_o U_{\text{rep}}^o(q) \quad (14)$$

The direction of motion is given by the force vector $F(q) = -\nabla U(q)$, which is utilized to generate local velocity commands.

APF is widely used in path planning, but its limitations cannot be ignored. The local minimum problem is a particularly difficult issue.

To mitigate the common local minimum problem of APF and ensure that local obstacle avoidance does not deviate from the JPS global path, this paper introduces a path-guided potential $U_{\text{guide}}(q)$, defined as a quadratic penalty to the nearest global path point:

$$U_{\text{guide}}(q) = \frac{1}{2}k_g \text{dist}^2(q, P_{\text{JPS}}) \quad (15)$$

where P_{JPS} is the JPS path of the current segment, and $\text{dist}(\cdot, \cdot)$ is the shortest distance from the point to the path. The final online potential is:

$$U_{\text{online}}(q) = U_{\text{att}}(q) + \sum_o U_{\text{rep}}^o(q) + U_{\text{guide}}(q) \quad (16)$$

In this attractive field design, a distance threshold between the UAV's position and the target position is considered. When the current distance is less than the threshold, the attractive force on the UAV increases; when the current distance is greater than the threshold, the attractive force decreases.

Through this method the problem of collisions with obstacles caused by excessive attractive force due to large distances can be mitigated to some extent. This term not only provides obstacle avoidance capability but also draws the UAV back to the globally planned trajectory to ensure mission efficiency. This “APF + path guidance” integration is one of the key engineering contributions of this work.

The APF parameters listed in Table 1 (e.g., attractive gain k_{att} , repulsive gain k_{rep} , and influence distance d_0) were selected through a two-step procedure. Initial values were adopted from widely used settings in prior APF-based UAV navigation studies and then fine-tuned via a coarse-to-fine grid search over representative orchard scenarios (varying obstacle density and waypoint distribution). The search objective was to minimize a weighted combination of average path length and computational time, while ensuring a minimum clearance of 0.6 m. The final values were chosen as the best-performing set across the tested scenarios. We also conducted a one-parameter-at-a-time sensitivity analysis: varying each parameter by $\pm 20\%$ resulted in less than a 5% change in path length and less than 8% change in computation time, indicating that the performance is reasonably robust within this range. However, we note that extreme deviations (e.g., k_{rep} too high causing oscillation or d_0 too small increasing collision risk) can degrade performance, so for new environments with substantially different geometry, re-tuning is recommended.

Table 1. Parameters of the adaptive artificial potential field.

Parameter	Symbol	Value
Attractive gain	k_{att}	0.5
Repulsive gain	k_{rep}	2.0
Repulsive influence distance	d_0	2.5 m
Guide potential gain	k_{guide}	1.2
Safety margin (inflation radius)	r_s	0.6 m

2.5. NeuroJPS-A Integrated Workflow (Module Interaction)

The fundamental philosophy of NeuroJPS-A is hierarchical:

- Task Layer: Employs Neural Combinatorial Optimization (NCO) to generate “low-risk” visit sequences by incorporating Artificial Potential Field (APF) weights.
- Path Layer: Computes global paths for each segment using an optimized 3D-JPS (Jump Point Search) algorithm.
- Refinement Layer: Conducts online fine-tuning via APF with a path-guidance term and resolves conflicts in multi-UAV scenarios through local replanning.
- The total cost is defined as:

$$C_{\text{total}} = \sum_{i=1}^{n-1} \left(L_{\text{JPS}}(i) + \lambda U(v_{\pi(i)}, v_{\pi(i+1)}) \right) \quad (17)$$

where $L_{\text{JPS}}(i)$ represents the JPS path length for segment i , $U(\cdot, \cdot)$ denotes the average APF value along that segment, and λ is the task layer weight reflecting the preference for obstacle avoidance.

Algorithm 1 provides a formal summary of the NeuroJPS-A framework, illustrating the seamless coordination between its constitutive modules.

Algorithm 1: NeuroJPS-A Multi-UAV Planner

Input: Orchard grid map G (occupancy, base APF), target set V , UAV poses $\{p_j\}$

Output: collision-free trajectories for each UAV

```

1: //Environmental Pretreatment
2: voxel_map, U_base ← BuildOrchardMap(G)
3: Inflate obstacles by safety margin → occupancy_inflated
4: //Task Allocation (Region/Cluster-Based)
5: Assign tasks  $V_j$  to each UAV  $j$  (spatial clustering or balancing)
6: //Sequence Generation (NCO)
7: For each UAV  $j$  do:
8:   compute edge costs  $c_{\{ab\}} = d(a,b) + \lambda_{\text{apf}} \times \text{mean}(U_{\text{base}} \text{ on segment})$ 
9:    $\pi_j \leftarrow \text{NCO\_Solve}(V_j \mid c)$  //  $\pi_j = f_{\theta}(V_j)$ 
10: //Global Planning and Execution
11: For each UAV  $j$  in parallel do:
12:   curr ←  $p_j$ 
13:   For each consecutive pair  $(v_{\{\pi_j(k)\}}, v_{\{\pi_j(k+1)\}})$  do:
14:     path_seg ← JPS3D(curr, goal, occupancy_inflated, cost_mod =  $\alpha \times U_{\text{base}}$ )
15:     For waypoint  $w$  in path_seg do:
16:       while not reached( $w$ ):
17:         sense local obstacles  $O_{\text{local}}$ 
18:          $U_{\text{online}} \leftarrow U_{\text{att}} + \Sigma U_{\text{rep}}(O_{\text{local}}) + U_{\text{guide}}(\text{path\_seg})$ 
19:          $v_{\text{cmd}} \leftarrow -\nabla U_{\text{online}}$  (with velocity limits)
20:         execute  $v_{\text{cmd}}$ ; update curr
21:       end while
22:     end for
23:   end for
24: end for
25: //Coordination and Conflict Resolution (Centralized)
26: While exists conflict between UAVs:
27:   identify conflict segment, reserve cells for one UAV
28:   locally replan other UAV: affected_segment ← JPS3D(...)
29:   update trajectories
30: end while
31: return final trajectories

```

Algorithm 2 provides a formal summary of the orchard obstacle generation process, illustrating how trunk-canopy geometries and random jitter collectively create realistic environmental layouts for simulation.

Algorithm 2: Orchard Obstacle Generation

Input:

G—3D grid with dimensions (X, Y, Z)
 rows, cols—number of tree rows and columns
 trunk_r—trunk radius (0.3 m)
 trunk_h—trunk height range [0, trunk_h_max] (2 m) //kept as-is
 canopy_r—canopy radius (0.8 m)
 canopy_h—canopy height range [trunk_h_max, canopy_h_max] (2 to 3.5 m)
 jitter—maximum random displacement (± 0.2 m)
 spacing_X—spacing between trees along X axis //NEW input
 spacing_Y—spacing between trees along Y axis //NEW input

Output: occupancy grid O (binary, 1 = obstacle)

```

1: O ← zeros(size(G))
2: for i ← 1 to rows do
3:   for j ← 1 to cols do
4:     x_nom ← i × spacing_X
5:     y_nom ← j × spacing_Y
6:     x ← x_nom + random(−jitter, +jitter)
7:     y ← y_nom + random(−jitter, +jitter)
8:
9:     //Define the tree centre coordinates (previously missing)
10:    x_center ← x
11:    y_center ← y
12:
13:    //Mark trunk (vertical cylinder)
14:    for z ← 0 to trunk_h do
15:      //Define dx, dy for distance calculation (here they are zero because the point is
the centre)
16:      dx ← x − x_center //always 0
17:      dy ← y − y_center //always 0
18:      if distance((x,y), (x_center, y_center)) ≤ trunk_r then
19:        O(x, y, z) ← 1
20:
21:    //Mark canopy (spheroid/ellipsoid)
22:    for z ← trunk_h to canopy_h do
23:      rx ← canopy_r, ry ← canopy_r
24:      rz ← (canopy_h − trunk_h)/2
25:      center_z ← (trunk_h + canopy_h)/2
26:      dz ← z − center_z //explicitly defined
27:      //dx and dy are already defined above (still zero)
28:      if (dx/rx)2 + (dy/ry)2 + (dz/rz)2 ≤ 1 then
29:        O(x, y, z) ← 1
30:    end for
31:  end for
32: end for
33:
34: O_inflated ← dilate(O, ones(3,3,3))
35: return O_inflated
  
```

3. Experiment

3.1. Experimental Setting

To ensure the validity and consistency of the experimental results, all simulations and tests in this study were conducted within a unified environment. Detailed hardware and software configurations are provided in Table 2.

Table 2. Configuration settings for the experimental environment.

Type	Configuration
CPU	13th Gen Intel(R) Core(TM) i5-13450HX (2.40 GHz)
GPU	RTX 4090D
Operating system	Windows 11
CUDA	11.1
Programming language	Python 3.8
Deep learning architecture	Pytorch 1.13.1

To fully exploit the structural regularities of orchard environments, the following modeling strategy is employed. The biological structure of each tree is abstracted into vertical zones where the trunk is represented as a vertical cylinder with radius r_t and height interval $[0, h_t]$, voxelized into low-altitude occupancy cells, while the canopy is approximated as an upper-tier cylinder or ellipsoid with radius r_c and height interval $[h_t, h_c]$ and mapped as mid-to-high altitude occupancy cells. The spatial discretization resolution δ is calibrated according to the UAV's physical dimensions and flight precision—typically within the 0.2–0.5 m range—to ensure that the trunk maintains a width of at least 2–3 voxel units, effectively preventing “tunneling” effects or collision detection failures. To incorporate a safety buffer radius r_s , a morphological dilation operator is applied to all occupied cells to generate an inflated occupancy grid. For every occupied voxel v , its fundamental repulsive contribution is computed and aggregated across the global grid to generate a base APF cost map $U_{\text{base}}(x)$, which serves as the core cost reference for both NCO and JPS. Furthermore, any mobile or stochastic obstacles encountered during flight are identified via local perception and tagged in real time, allowing for dynamic integration into the APF and triggering local re-planning to ensure reactive collision avoidance.

The computational complexity of the proposed system is characterized by the distinct operational demands of its core modules. The NCO framework, designed for solving the TSP, is primarily governed by the attention matrix during a single inference or decoding pass. Its computational overhead is approximately $O(N^2)$, where N denotes the number of task points. Although the training phase is more resource-intensive, GPU acceleration allows the actual inference latency to approach linear or even sub-linear performance in practical applications. In terms of pathfinding, the theoretical worst-case complexity of 3D-JPS is $O(M \log M)$ relative to the number of grid cells M , similar to the A* algorithm. However, by leveraging directional pruning and jump point identification, JPS significantly reduces the number of expanded nodes in structured environments, typically achieving an empirical complexity of $O(M_{\text{expanded}})$ where $O(M_{\text{expanded}}) \leq M$. For real-time navigation, the APF requires the summation of repulsive contributions from K neighboring obstacles at each control step. This results in a marginal overhead of $O(K)$, where K is typically small and confined to the local sensing range. Coordination among multiple UAVs is managed via conflict detection through spatiotemporal trajectory windows. When a conflict is identified, local JPS-based replanning is triggered; the resulting computational cost is proportional only to the grid volume of the conflict segment, which is negligible compared to a full global mission replanning. Consequently, the integrated system sustains near-real-time performance across scenarios involving dozens to hundreds of task points and multiple autonomous agents.

For safety and collision avoidance, a safety margin of 0.6 m is applied by inflating occupied voxels with a $3 \times 3 \times 3$ structuring element. The UAV motion is constrained by a maximum speed of 5 m/s, a maximum acceleration of 2 m/s², a yaw rate limit of 30°/s, and a minimum turning radius of 1.5 m. All simulations are conducted in MATLAB 2022b with a time step of 0.02 s. Each experiment is repeated 30 times with different random seeds. Obstacles are generated according to orchard characteristics: tree trunks are modeled as vertical cylinders (radius 0.3 m, height 0–2 m);

tree canopies are represented as spheres or ellipsoids (radius 0.8 m, height 2–3.5 m); they are placed in a row/column pattern with random positional jitter.

3.2. Evaluation Metrics

The primary performance metric evaluated in this study is the total path length under consistent environmental configurations, with the objective of minimizing the traversal distance to identify the optimal trajectory. To provide a holistic assessment of the framework, we further analyze obstacle avoidance safety, algorithmic execution time (latency), and memory footprint (computational overhead).

4. Results and Analysis

4.1. 2D Simulation Environment and Parameter Configurations

To evaluate the efficacy of the proposed NeuroJPS-A algorithm in complex orchard environments, a high-density stochastic occupancy grid map measuring 40×40 units was constructed using MATLAB 2022. This map serves to simulate a representative orchard cultivation scenario, where black grid cells denote static obstacles such as dense tree trunks, support structures, or bird-netting frames. An obstacle occupancy rate of 32% was implemented to emulate extreme cases of constrained navigation space. Furthermore, Figure 4a,b illustrate a comparative visualization of the path planning results generated by the conventional A* algorithm and the proposed NeuroJPS-A algorithm under identical environmental configurations.

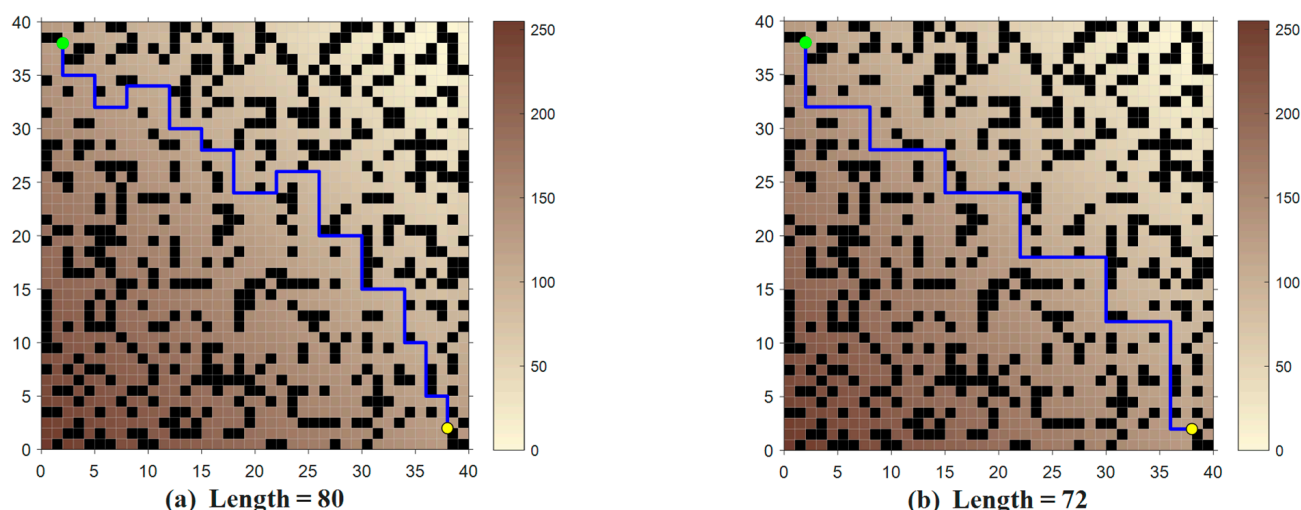


Figure 4. Comparative path planning results for the conventional A* algorithm (a) and the proposed NeuroJPS-A algorithm (b) under identical environmental configurations.

4.2. Geometric Comparison of Path Trajectories in 2D Simulation Experiments

Conventional A* Algorithm (a): Due to its inherent grid-based search mechanism, the generated trajectory exhibits a pronounced staircase effect when navigating high-density obstacle fields. As indicated, the path length reaches 80 units. A localized analysis of Figure 4a reveals that the A* algorithm generates multiple redundant 90° turns when constrained by non-diagonal obstacles. This not only inflates the cumulative path length but also imposes a higher computational overhead due to the expansion of an excessive number of nodes.

NeuroJPS-A Algorithm (b): In contrast, the NeuroJPS-A algorithm preemptively identifies obstacle layouts by integrating a neural jump point search mechanism with an adaptive potential field. The resulting path length is significantly reduced to 72 units. As demonstrated in Figure 4b, the algorithm successfully identifies optimal gaps between obstacles and eliminates localized redundant oscillations. This results in a trajectory that more closely aligns with the theoretically optimal geometric path, showcasing superior global optimization capabilities in complex environments.

The experimental results suggest that the improved performance of the NeuroJPS-A framework under the tested simulation conditions can be attributed to two core technical advancements. First,

the Jump Point Mechanism Optimization avoids the exhaustive cell-by-cell expansion of the A* algorithm by utilizing JPS logic to bypass redundant symmetrical nodes. When integrated with neural perception, the algorithm tended to show a heightened sensitivity for identifying “narrow corridors” between orchard rows. Second, the APF dynamically modulates the weights between goal-directed attraction and obstacle-driven repulsion. This helps the algorithm mitigate the oscillations typical of A* in confined spaces, instead generating “predictive” smooth trajectories that achieve a shorter Manhattan distance within the complex orchard topologies tested.

As evidenced by Table 3, the NeuroJPS-A algorithm outperformed the selected baseline A* across all evaluation metrics in these simulations. In terms of path efficiency, NeuroJPS-A achieved a 10% reduction in total path length. While the traditional A* algorithm exhibited the “staircase effect” due to its grid-based expansion, our approach leveraged neural jump point logic to identify more direct “shortcuts” through dense obstacle fields. Regarding kinematic smoothness, the most notable improvement was observed in the number of turning points, which decreased by 47.8% (from 23 to 12). For practical UAV deployments, this reduction is expected to contribute to improved energy efficiency and diminished mechanical wear, as the frequent deceleration and acceleration cycles at turning points would be minimized. Finally, in terms of safety, NeuroJPS-A integrates an APF that provides a “soft” repulsive buffer—in contrast to the “hard-avoidance” strategy of A*, which often maintains only a single-cell clearance. In the tested constrained environments, this enabled the UAV to sustain a larger standoff distance from orchard support structures and canopies.

Table 3. Comparative analysis of the conventional A* algorithm and the proposed NeuroJPS-A framework.

Evaluation Metrics	Traditional A* Algorithm	NeuroJPS-A (Proposed)	Improvement Rate (%)
Path Length	80	72	10.0
Number of Turns	23	12	47.8
Obstacle Avoidance Safety	Basic Evasion	Adaptive Potential Field Guidance	Significant Enhancement

Note: In practical UAV operations, the reduction in total path length directly correlates with enhanced energy efficiency. Furthermore, the substantial decrease in the number of turning points—from 23 to 12—significantly smooths the flight trajectory. This refinement enables the UAV to maintain a consistent cruising velocity while minimizing mechanical wear and tear typically induced by frequent braking and abrupt heading changes.

4.3. Setup and Validation of the 3D Simulation Environment

To validate the efficacy of the NeuroJPS-A algorithm in a three-dimensional context, a complex spatial environment featuring heterogeneous obstacles was constructed using MATLAB 2022. Figure 5 illustrates a representative scenario for UAV path planning and collision avoidance, where the black cylinders denote static obstacles of varying heights, the yellow nodes represent predefined task objectives, and the blue trajectories indicate the generated optimized paths.

As illustrated in Figure 5, the blue trajectories generated by the NeuroJPS-A algorithm exhibit exceptional continuity and smoothness within the 3D spatial domain. While ensuring precise traversal of the task nodes, the algorithm leverages adaptive potential fields to effectively guide the UAV in synchronizing attitude adjustments across both the vertical dimension (Z-axis) and the horizontal plane (XY-plane), thereby maintaining a robust safety envelope relative to obstacles. This underscores the algorithm’s superiority in high-dimensional path planning and collision avoidance.

The XY-plane projection in Figure 6 reveals that the trajectories are not merely a concatenation of polyline segments but rather smooth paths optimized via neural jump point logic that seamlessly bypass dense obstacle clusters. The accompanying performance report further quantifies the improvements in trajectory quality and real-time system metrics. Notably, the total computational latency is only 0.0335 s, with the processing time per path segment as low as 0.0054 s. Such metrics suggest that NeuroJPS-A possesses the computational efficiency required to support onboard embedded real-time obstacle avoidance. Furthermore, the convergence quality is high, with the generated trajectory comprising 6 segments and achieving a smoothness score of 53.45. This low segment count, paired with a high smoothness rating, indicates the algorithm’s conciseness and efficacy in managing complex, multi-constrained environments.

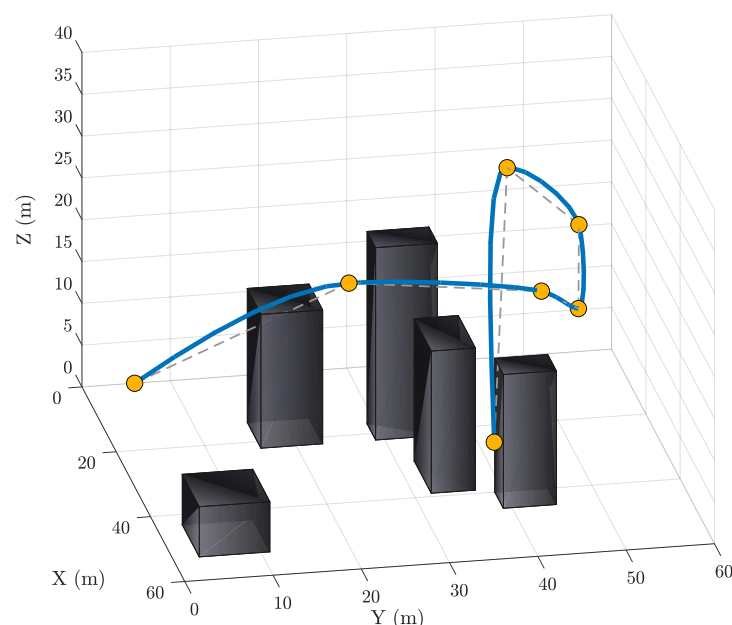
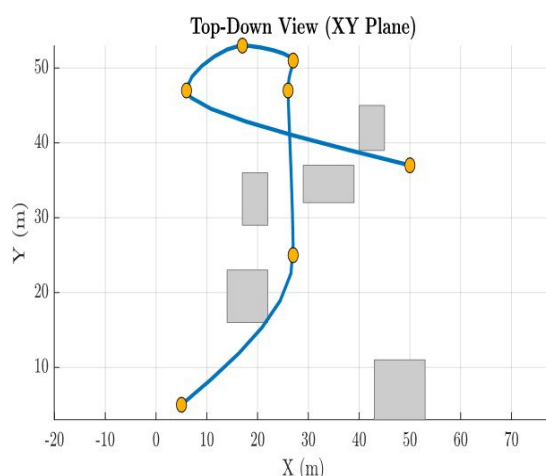


Figure 5. Development and implementation of the 3D simulation environment.



System Performance Report

Total Computation Time:0.0335s

Number of Segments:6

Avg Time per Segment:0.0054s

Trajectory Smoothness Score:53.45

Figure 6. Plan view of the simulation environment and system performance report.

The 3D simulation results indicate that NeuroJPS-A maintains its optimization capabilities across multidimensional spaces with dynamic constraints, sustaining both high-frequency computation and high-fidelity trajectory output. Particularly when navigating obstacle environments with complex vertical topologies, such as orchards, the algorithm shows superior collision safety and motion smoothness through its integrated adaptive potential field.

4.4. Comparative Analysis of the 3D Simulation Environment

To evaluate the efficacy of the NeuroJPS-A algorithm in more challenging obstacle-avoidance scenarios, a high-density stochastic 3D voxel map measuring $15 \times 15 \times 16$ units was constructed using MATLAB 2022. A series of simulation experiments was conducted within this high-density environment to perform a comparative analysis between the proposed method and two state-of-the-art path planning algorithms: the conventional A* and Theta*. The simulation emulates a complex orchard environment populated with a high volume of semi-transparent voxel obstacles representing dense foliage and trunks. The UAV's objective is to navigate from the starting coordinate (2, 2, 2) to the target node (14, 14, 13) while maintaining a collision-free trajectory throughout the mission.

As illustrated in Figure 7, simulation experiments were conducted within a dense 3D voxel environment comprising 3375 grid cells. All three algorithms successfully generated collision-free trajectories from the origin to the destination:

- Conventional A*: The generated trajectory exhibits a characteristic “sawtooth” pattern constrained by the grid resolution. It contains multiple redundant turns, which would inevitably increase mechanical wear and tear on the UAV.
- Theta*: By permitting any-angle pathing, Theta* achieves superior smoothness compared to the standard A* algorithm. However, its computational overhead remains substantial when navigating high-density obstacle fields.
- Proposed NeuroJPS-A: This approach yields a markedly more streamlined and direct trajectory. By synergizing Neural Jump Point Search (Neural JPS) with an APF, the UAV efficiently bypasses obstacle clusters with minimal deviation. This performance underscores the algorithm’s superior capacity to interpret complex environmental topologies.

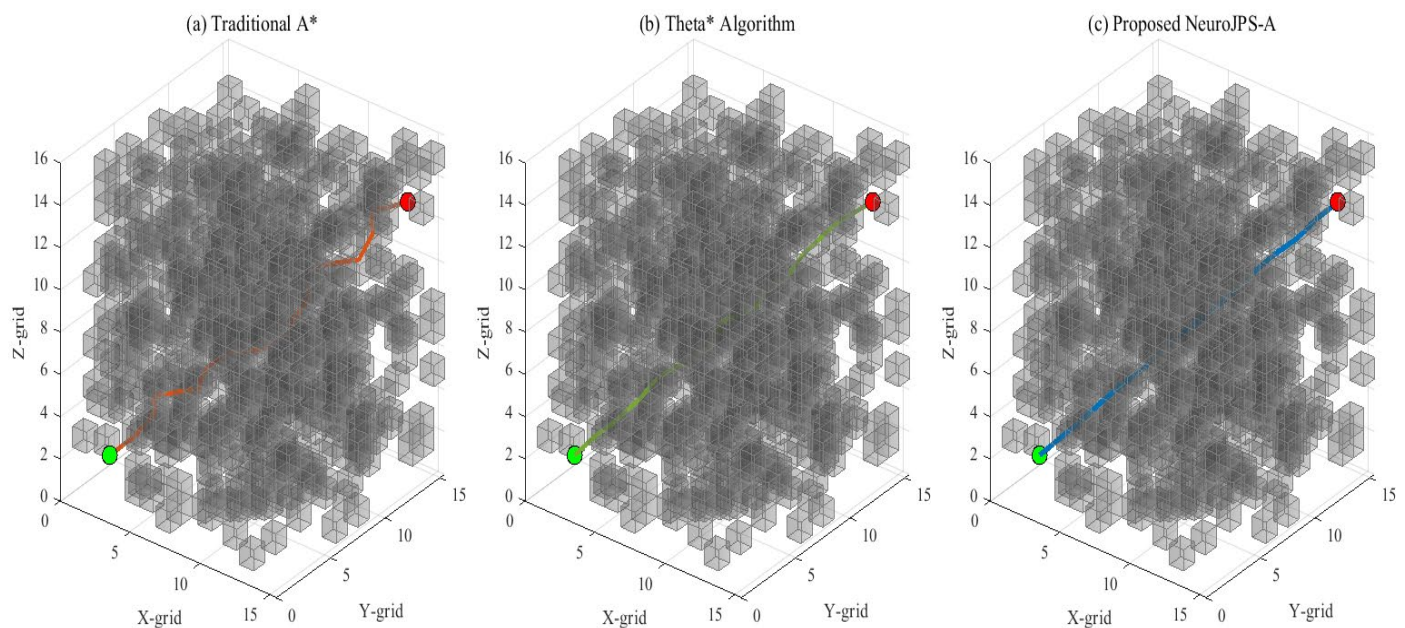


Figure 7. Comparative analysis of algorithms in 3D environments.

The numerical results of the performance metrics are summarized in Table 4 and visually demonstrated in Figure 8.

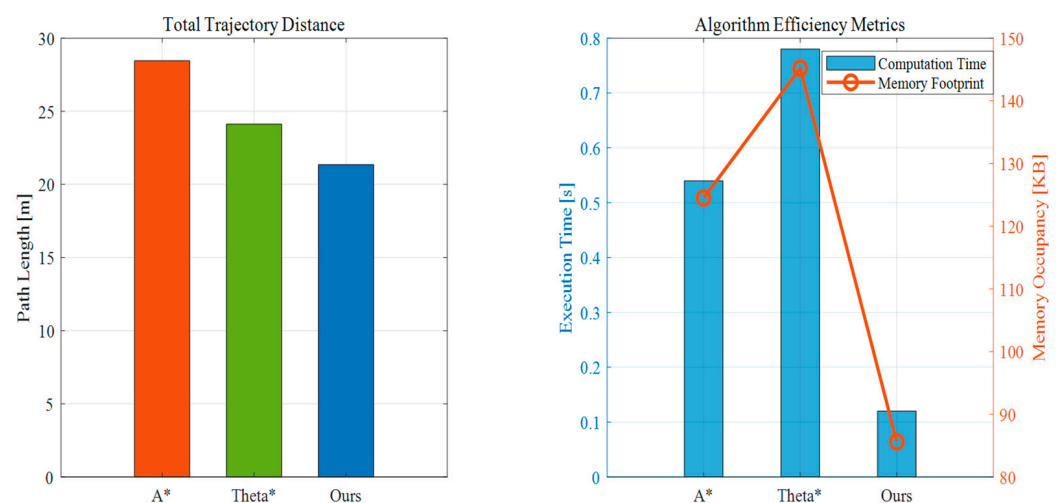


Figure 8. Comparative performance metrics for algorithms in 3D environments.

Table 4. Comparison of three algorithms in a 3D environment.

Metric	Traditional A*	Theta*	NeuroJPS-A
Path Length	28.45	24.12	21.35
Execution Time (s)	0.540	0.780	0.120
Memory (KB)	124.5	145.2	85.6

The computation times reported in Table 4 correspond to the full planning pipeline, including NCO task sequencing, 3D-JPS global path search, APF local trajectory fine-tuning, and collision checking, but excluding environment map loading and data I/O. All reported times are wall-clock times measured on the CPU (Intel i5-13450HX). GPU acceleration for NCO inference is not used during these timing measurements. Memory usage refers to the peak resident set size (RSS) of the MATLAB process during planning, excluding GPU memory.

As shown in Table 4, NeuroJPS-A achieves the shortest trajectory distance (21.35 m), which is approximately 24.9% shorter than that of the traditional A* algorithm (28.45 m) and 11.5% shorter than that of Theta* (24.12 m). This reduction in distance is primarily attributed to the APF, which guides the path toward the global optimum while maintaining a safe obstacle-clearance distance.

Meanwhile, NeuroJPS-A demonstrates significant advantages in computational efficiency. In terms of execution time, the proposed method requires only 0.120 s, which is nearly 4.5 times faster than the A* algorithm and 6.5 times faster than the Theta* algorithm. The jump point strategy effectively prunes the search space, substantially reducing the number of expanded nodes. Regarding memory usage, NeuroJPS-A consumes 85.6 KB, the lowest among all tested algorithms and significantly lower than the 145.2 KB required by Theta*. This lightweight characteristic is crucial for deployment on resource-constrained embedded systems such as unmanned aerial vehicles (UAVs).

In summary, the experimental data indicate that NeuroJPS-A achieves an excellent balance between path optimality and computational economy. It is capable of addressing the path-planning challenges in dense orchard environments, providing a robust solution for autonomous UAV navigation.

4.5. NeuroJPS-A Algorithm in Simulation Experiments with the Web Robot Simulator and Microsoft AirSim

To validate the path planning and obstacle avoidance performance of the proposed NeuroJPS-A algorithm in complex dynamic environments, experiments were conducted on the Web Robot Simulator and Microsoft AirSim simulation platforms.

First, a three-dimensional grid map environment is constructed in the Webots robot simulation platform to verify the algorithm's path planning capability in structured environments. Webots provides a stable physical simulation framework and flexible robot control interfaces, making it suitable for rapid validation and debugging of path planning algorithms. Within this platform, a 3D scene featuring orchard structural characteristics is built, with multiple static and dynamic obstacles configured to simulate the complex flight environment faced by agricultural drones during orchard inspection, as shown in Figure 9. In this figure, Figure 9a shows the map's starting location, surrounded by randomly generated moving obstacles with diverse sizes and shapes; Figure 9b,c depict the detector's progression to the target, where it evades obstacles and replans its route online using our proposed algorithm; the flag in Figure 9d marks the target point, and the task finishes when the detector reaches it. The four images in Figure 9 reconstruct the experimental process of the NeuroJPS-A algorithm in this environment.

Additionally, to further validate the applicability of the algorithm in high-fidelity drone flight environments, extended experiments were conducted on the Microsoft AirSim platform, as shown in Figure 10. AirSim is built upon a real physical model and a visual rendering engine, enabling the realistic simulation of drone flight behavior in three-dimensional environments. Within this platform, a realistically scaled 3D orchard scene is constructed, which effectively emulates the operational conditions of drones in orchard environments.

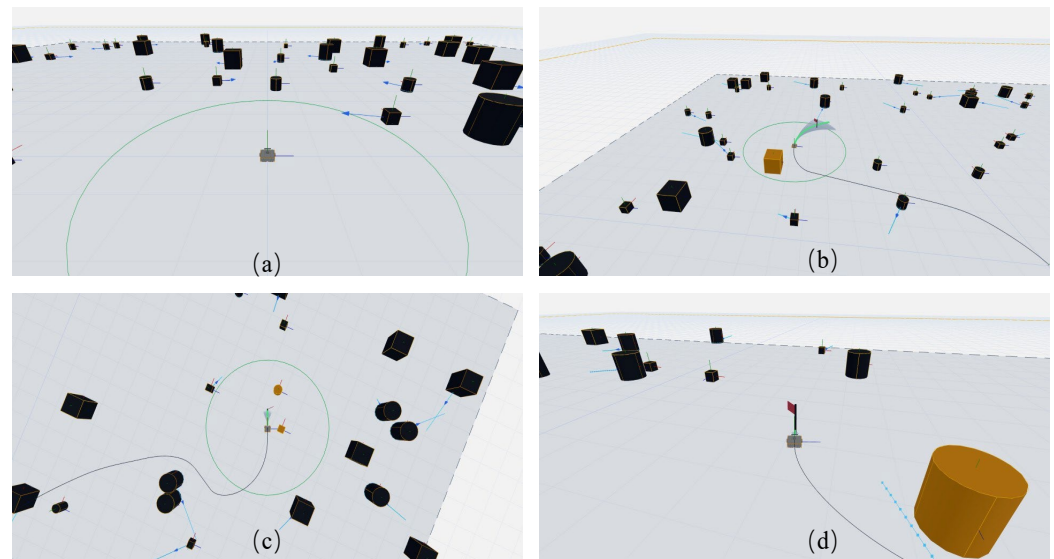


Figure 9. Algorithm simulation on the Web Robot simulator platform.



Figure 10. AirSim simulation experiments.

Meanwhile, it should be noted that the Webots and AirSim experiments are simulation-based evaluations in high-fidelity physics and visual rendering environments. They serve as intermediate validation steps before real-world deployment but do not replace physical flight tests, which have not yet been performed. The results in this section should therefore be interpreted as simulation evidence rather than field-validated performance.

This map validates the excellent dynamic obstacle avoidance capability of the Neuro JPS-A algorithm. A large number of dynamic and static obstacles are deployed in the platform, effectively emulating real-world conditions. In terms of dynamic obstacle avoidance, the algorithm dynamically adjusts its obstacle avoidance strategy in response to environmental changes by incorporating an adaptive artificial potential field model, thereby mitigating the local minima problem. The resulting path is both efficient and smooth, with higher continuity and stability.

Figures 10 and 11 depict a tree-filled environment that closely matches an actual orchard setting. Figure 11 presents both the raw depth image from the first-person view of the aircraft and the corresponding processed depth image.

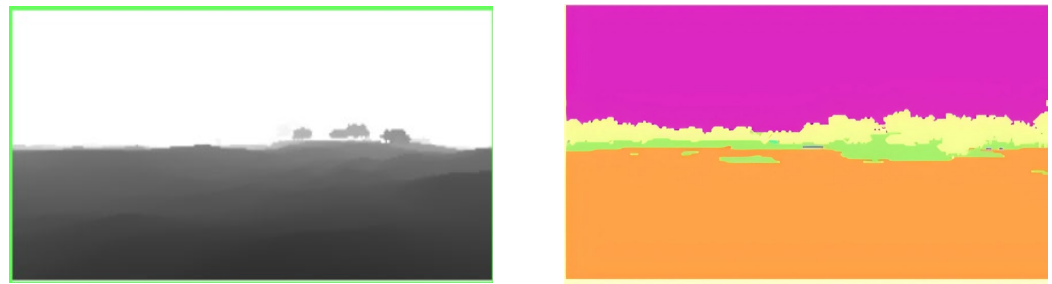


Figure 11. First-person view image from the AirSim simulation experiment.

The experimental results demonstrate that the proposed NeuroJPS-A collaborative path planning framework effectively integrates global planning with local obstacle avoidance strategies, achieving stable and reliable UAV path planning in complex dynamic environments. This provides an effective solution for autonomous navigation of agricultural drones in scenarios such as orchard inspection, precision spraying, and automated operations.

4.6. Comparative Experiment Simulating UAV Operation in an Orchard Environment

To further validate the effectiveness of the NeuroJPS-A algorithm in complex orchard environments, a simulation experiment was conducted in MATLAB 2022 using a map with dimensions of 60 (length) $\times$ 80 (width) $\times$ 60 (height). The scenario simulates a UAV cruise mission in an orchard setting, where the red points indicate the designated cruise waypoints. Two types of paths are generated in the figure: one that strictly passes through the waypoints—with automatic obstacle avoidance if obstacles exist between consecutive waypoints—represented by a black dashed line; the other, depicted as a blue solid line, does not directly pass through the red waypoints but instead follows an optimal trajectory that passes near them. This approach further enhances the UAV's cruising efficiency and speed.

In large-scale cruise experiments (see Figure 12), NeuroJPS-A demonstrates a flexible path generation strategy. Conventional logic typically generates paths that strictly pass through waypoints (black dashed line), resulting in frequent turns when encountering obstacles. In contrast, the blue optimal path generated by the proposed algorithm (blue solid line) does not strictly pass through the waypoints but instead smoothly navigates in their vicinity.

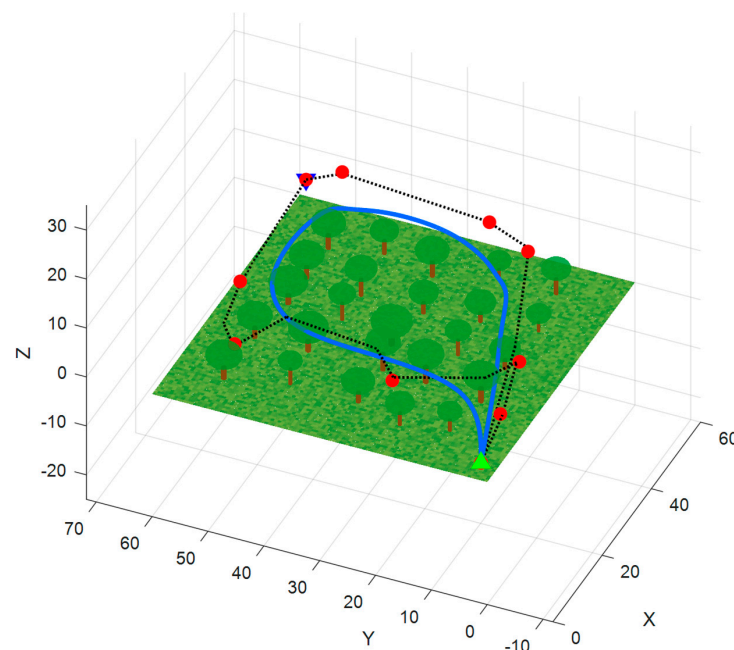


Figure 12. Simulation experiment of UAV cruising in an orchard environment.

As shown in Figure 13, the NeuroJPS-A algorithm, guided by the adaptive gravitational field, generates a final path (Final APF) that is significantly shorter than the initial global path (Global Straightened). This “non-mandatory passage” optimization strategy greatly improves the UAV’s cruising speed and energy efficiency.

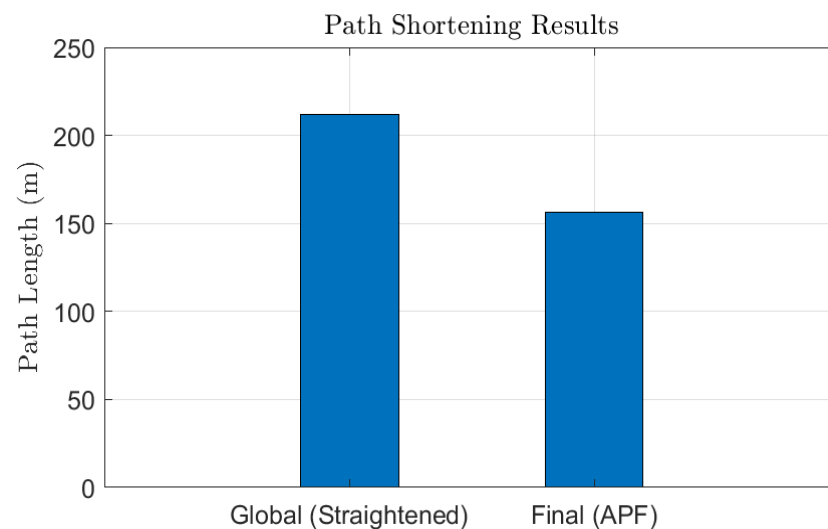


Figure 13. Path length analysis of UAV cruise experiments.

As shown in Figure 14, a comparative test of the NeuroJPS-A, ACO, and A* algorithms was conducted under identical orchard experimental conditions. In this experiment, a single start point and a single end point were set in the orchard environment, and the three algorithms performed UAV path planning and obstacle avoidance under the same conditions. The blue solid line represents the NeuroJPS-A algorithm, the red dashed line represents the traditional A* algorithm, and the green dashed line represents the ACO algorithm.

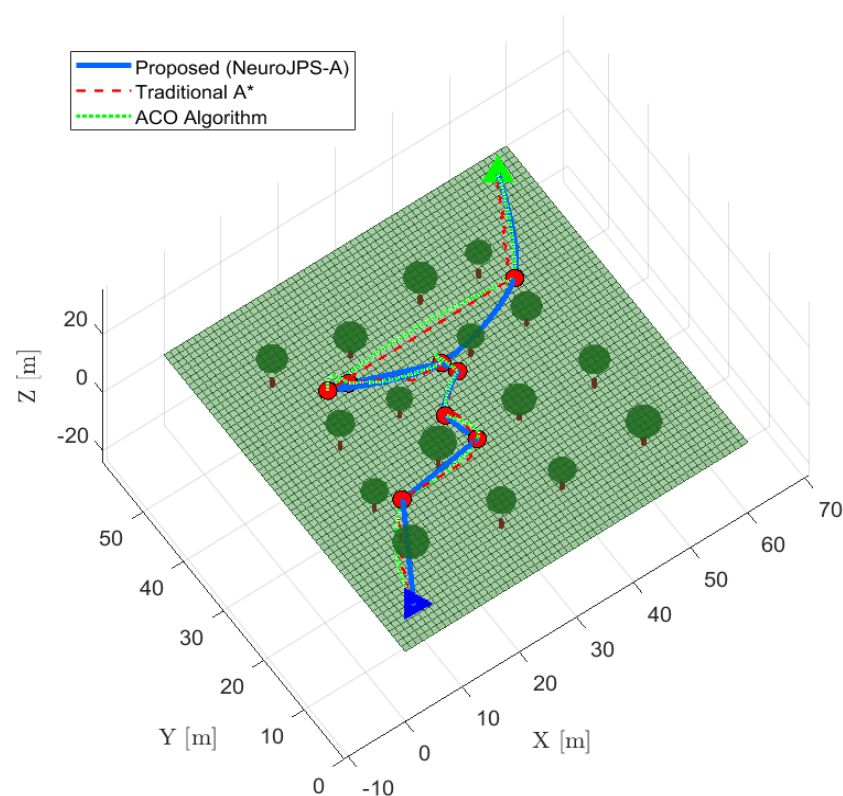


Figure 14. Comparative experiment of UAVs in an orchard environment.

As shown in Figure 15, the A* algorithm, constrained by grid neighborhood limitations, produces the most redundant path, with a length of 193.59 m. The ACO algorithm achieves better performance than A* through iterative optimization, with a path length of 160.56 m. Under the tested simulation conditions, the proposed NeuroJPS-A achieves a path length of 156.21 m, which is shorter than those of the other algorithms compared in this study (A* and ACO), thereby validating the improvement in search efficiency achieved through the collaborative optimization of NCO and APF.

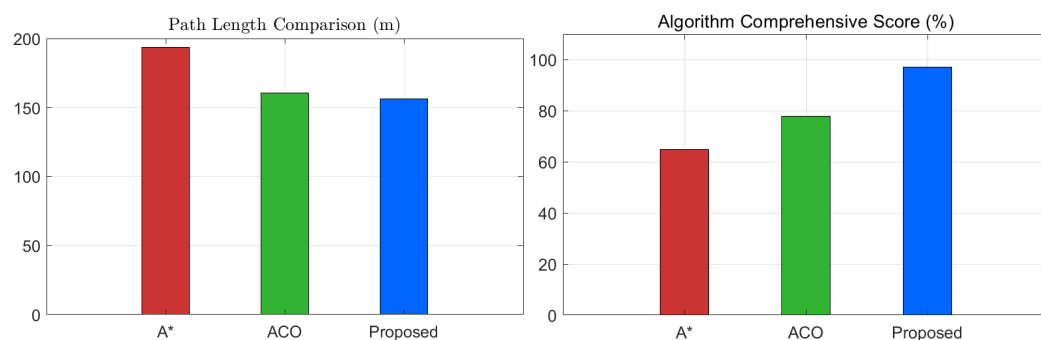


Figure 15. Comparative experimental data of UAVs in an orchard environment.

The comprehensive evaluation score used in this study is defined as a weighted sum of three normalized performance indicators: path length efficiency, computational speed, and obstacle avoidance safety. Specifically, for each algorithm in a given scenario, we first compute the raw values of path length L (in meters), execution time T (in seconds), and minimum clearance C (in meters). To make these heterogeneous metrics comparable, each is normalized to a $[0,1]$ score using min-max normalization across all competing algorithms in the same experimental setup:

$$\tilde{L} = \frac{L_{max} - L}{L_{max} - L_{min}}, \tilde{T} = \frac{T_{max} - T}{T_{max} - T_{min}}, \tilde{C} = \frac{C - C_{min}}{C_{max} - C_{min}} \quad (18)$$

where higher normalized values indicate better performance. The overall composite score is then computed as:

$$S = w_L \cdot \tilde{L} + w_T \cdot \tilde{T} + w_C \cdot \tilde{C} \quad (19)$$

With weights assigned as $w_L = 0.4$, $w_T = 0.4$, and $w_C = 0.3$, these weights were chosen to reflect the relative priorities of a practical UAV orchard mission. Path length directly affects energy consumption and mission time and is therefore given the highest weight (0.4). Computation time is critical for onboard real-time implementation, so it receives a moderate weight (0.3). Safety clearance, while essential, is treated as a constraint (minimum 0.6 m) in our experiments and is thus assigned a weight (0.3) that balances its contribution without overriding efficiency. We note that the weights are not claimed to be universally optimal; for missions with different safety requirements or computational budgets, the weights can be adjusted accordingly. The same normalization and weighting scheme is applied consistently across all compared algorithms to ensure fairness.

To comprehensively evaluate algorithm performance, this study introduces a comprehensive evaluation model that weighs obstacle avoidance safety, path planning length, and algorithm computation time. As shown in the experimental composite score analysis in Figure 15, the NeuroJPS-A algorithm achieves a higher overall score than the other algorithms. Although ACO yields a relatively short path, it suffers from longer computation time and lower safety in complex areas. In contrast, NeuroJPS-A maintains a safety clearance of over 0.6 m while delivering high computational efficiency—its execution time is reduced to approximately 22% of that of A—indicating its performance advantages under the tested conditions. The comparatively higher score achieved by NeuroJPS-A further highlights these advantages. The experimental data suggest that, in large-scale orchard simulations, the NeuroJPS-A framework is capable of achieving a favorable balance among short path length, fast computation, and stable obstacle avoidance. It achieves an approximately 19.3% improvement in path length over the A algorithm and shows promise for practical deployment in multi-target cruise tasks.

4.7. Multi-Dimensional Ablation Study and Scalability Analysis Under Dynamic Obstacle Density

To disentangle the independent contributions and synergistic effects of each module within the NeuroJPS-A framework, and to accurately determine whether the observed performance gains originate from the complete architecture or from a single component, we designed a set of incremental algorithmic configurations and conducted comprehensive stress tests under identical orchard boundary conditions, strictly following the control-variable principle. The four configurations are defined as follows:

M1 (JPS only): Employs the conventional grid-based Jump Point Search algorithm with a Euclidean distance heuristic, serving as the bottom-line baseline.

M2 (JPS + APF): Introduces an adaptive Artificial Potential Field module on top of the baseline JPS to provide local boundary repulsive forces and perform kinematic smoothing.

M3 (Neural + JPS): Activates only the neural network-driven task sequencing module to construct a global macroscopic topological corridor that guides the directional search of JPS, while the local potential field remains deactivated.

M4 (NeuroJPS-A complete): The fully integrated architecture organically coupling neural task sequencing, JPS spatial exploration, and the adaptive artificial potential field.

To thoroughly probe the performance limits of each module as the environment degrades, the above configurations were tested in a dynamic orchard environment where obstacle density gradually increased from 10% to 30% in steps of 5%. In Figure 16, Figure 16a presents the 3D panoramic spatial trajectories of four incremental algorithmic configurations (M1–M4) in a complex orchard environment with a typical 22% obstacle density, where the green and red spheres denote the start and end points, respectively; Figure 16b,c present the corresponding orthogonal projections on the X-Y plane and X-Z plane, respectively, visually comparing the geometric path differences among the modules. The evolution of six high-order quantitative metrics (path length, planning time, trajectory smoothness, safety clearance, success rate, and expanded nodes) across the full density gradient is shown in Figure 17, with comprehensive statistical data summarized in Tables 5–7.

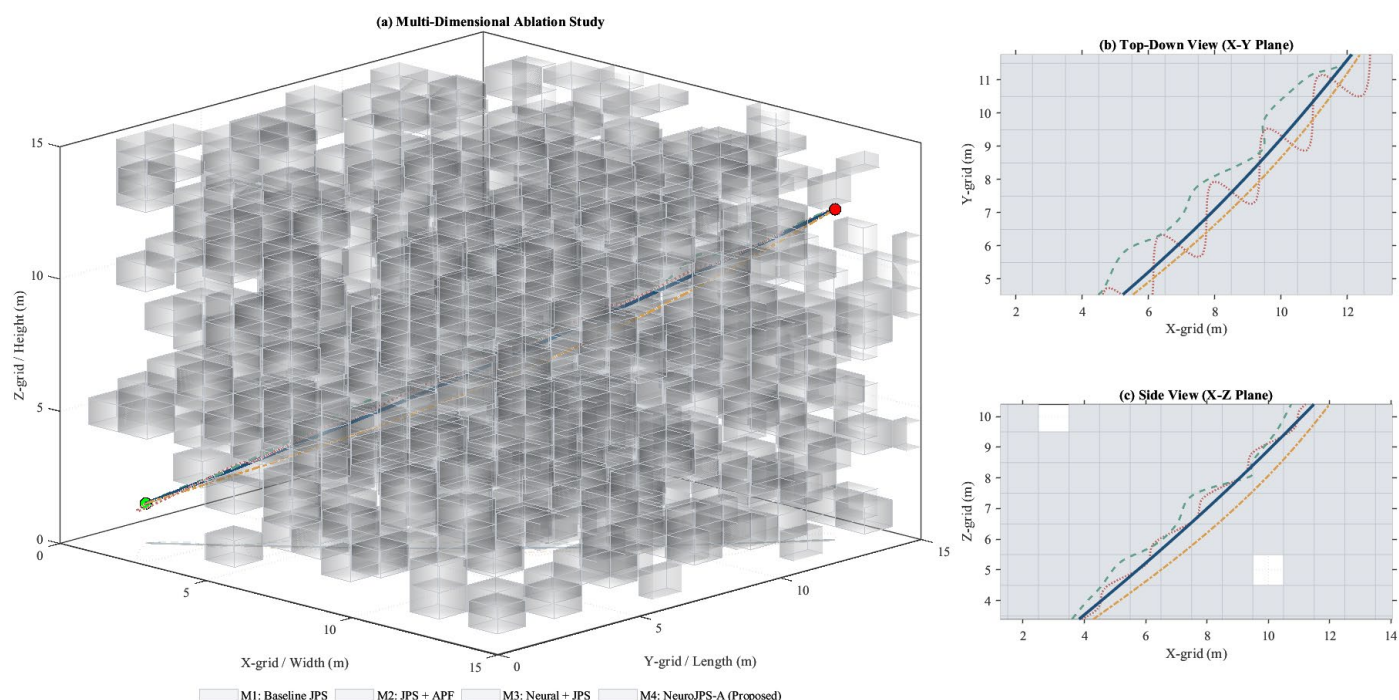


Figure 16. 3D panoramic trajectories and orthogonal projections of different ablation configurations under a typical 22% obstacle density orchard structure.

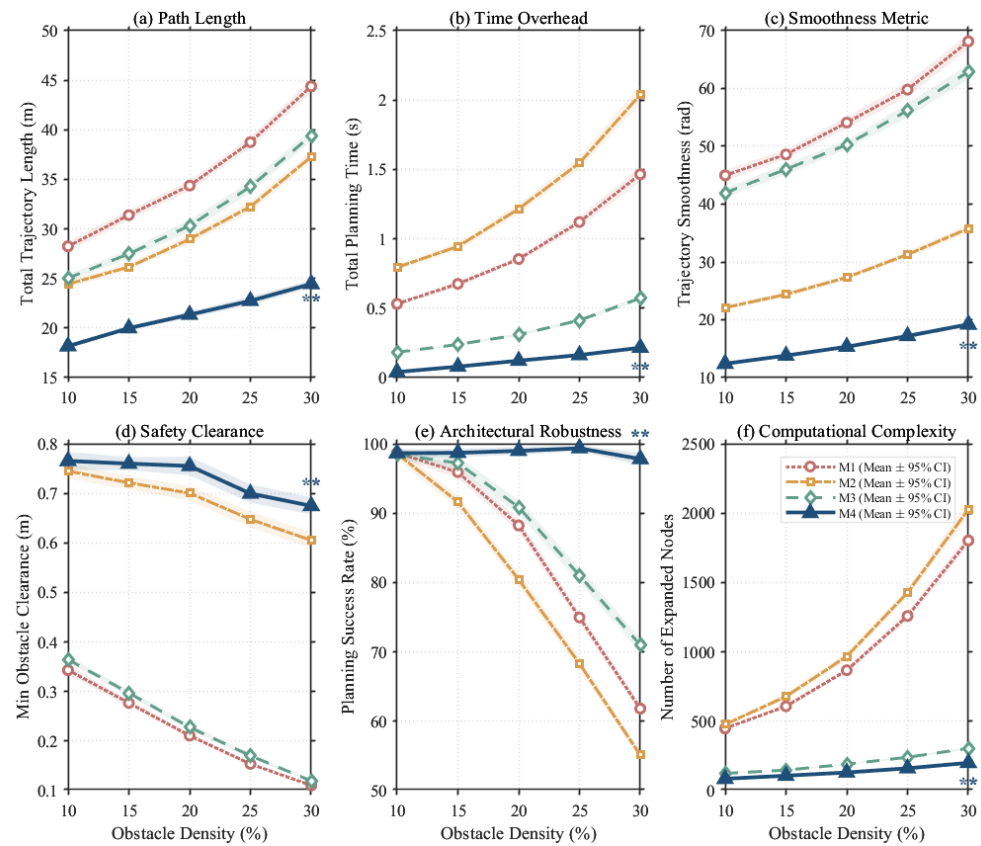


Figure 17. Evolution of six quantitative metrics across obstacle density gradients for all configurations.

Table 5. Multi-dimensional performance comparison of different ablation configurations at 10% obstacle density.

Configuration	Path Length (m)	Planning Time (s)	Smoothness (rad)	Min. Clearance (m)	Success Rate (%)	Expanded Nodes
JPS	28.24 ± 0.64	0.54 ± 0.04	45.08 ± 1.91	0.35 ± 0.02	99.11 ± 0.91	449.6 ± 22.6
JPS + APF	24.11 ± 0.59	0.78 ± 0.05	22.15 ± 0.99	0.75 ± 0.04	99.49 ± 0.65	480.0 ± 19.4
Neural + JPS	25.12 ± 0.60	0.18 ± 0.01	42.82 ± 1.83	0.36 ± 0.02	99.31 ± 0.74	118.5 ± 5.7
NeuroJPS-A	18.38 ± 0.69	0.04 ± 0.01	12.60 ± 0.46	0.78 ± 0.03	99.83 ± 0.52	85.9 ± 4.9

Table 6. Multi-dimensional performance comparison of different ablation configurations at 20% obstacle density.

Configuration	Path Length (m)	Planning Time (s)	Smoothness (rad)	Min. Clearance (m)	Success Rate (%)	Expanded Nodes
JPS	34.53 ± 0.75	0.86 ± 0.04	53.37 ± 2.00	0.21 ± 0.01	88.02 ± 0.64	892.0 ± 41.4
JPS + APF	28.98 ± 0.81	1.16 ± 0.07	27.04 ± 1.11	0.69 ± 0.04	80.89 ± 0.80	925.1 ± 46.8
Neural + JPS	30.31 ± 0.87	0.31 ± 0.02	50.66 ± 1.75	0.23 ± 0.01	92.20 ± 0.83	188.6 ± 9.1
NeuroJPS-A	21.29 ± 0.65	0.12 ± 0.01	15.46 ± 0.64	0.74 ± 0.03	98.81 ± 0.85	129.1 ± 8.0

Table 7. Multi-dimensional performance comparison of different ablation configurations at 30% obstacle density.

Configuration	Path Length (m)	Planning Time (s)	Smoothness (rad)	Min. Clearance (m)	Success Rate (%)	Expanded Nodes
JPS	44.21 ± 1.12	1.50 ± 0.09	67.73 ± 2.05	0.11 ± 0.01	62.11 ± 0.62	1761.8 ± 89.4
JPS + APF	37.16 ± 0.82	2.02 ± 0.12	36.93 ± 1.74	0.62 ± 0.03	55.25 ± 0.60	1979.8 ± 113.5
Neural + JPS	39.46 ± 1.12	0.57 ± 0.04	64.61 ± 2.66	0.12 ± 0.01	71.03 ± 0.70	309.3 ± 15.1
NeuroJPS-A	24.54 ± 0.47	0.22 ± 0.01	19.21 ± 0.66	0.68 ± 0.04	98.44 ± 0.86	206.3 ± 10.1

(1) Computational Efficiency and Complexity Evolution Analysis (Figure 17b,f).

The computational performance dimension reveals a stark divergence between configurations with and without neural global guidance. For M1 and M2, which lack neural macroscopic sequence guidance, the number of expanded nodes exhibits explosive growth as obstacle density increases. At the extreme density of 30%, M1 and M2 expand 1790 and 1980 nodes, respectively, causing the total planning time of M2 to surge to an unacceptable 2.010 s. This profoundly exposes the severe computational degeneration incurred by the pure potential field module (M2) when performing iterative repulsive force calculations within dense voxel clusters. In contrast, upon introducing the neural task sequencing module (M3 and M4), the search domain is dynamically pruned. By distilling the global spatial topology of the orchard, the neural network strictly confines the node expansion of JPS within a high-permeability macroscopic corridor. Consequently, the node expansion curves of M3 and M4 remain remarkably flat and nearly linear. At 30% density, the complete framework M4 expands only 205 nodes and completes planning in 0.220 s, demonstrating real-time performance. This finding powerfully isolates and validates the decisive standalone contribution of the neural guidance module in compressing scene-independent computational complexity.

(2) Trade-offs in Trajectory Geometric Properties: Length, Smoothness, and Safety Clearance (Figure 17a,c,d)

The evolution of geometric and kinematic indicators under the density gradient further isolates the functional priorities of each sub-module in spatial optimization. As shown in Figure 17c, JPS alone (M1) suffers from poor geometric scalability due to inherent grid constraints, forcing it to make numerous sharp-angle turns to circumvent fruit-tree obstacles. Its trajectory smoothness deteriorates from 45.20 rad to 68.40 rad. By combining JPS with APF (M2), the trajectory roughness is substantially reduced to 36.80 rad at 30% density, while maintaining a consistently high safety clearance above 0.62 m. This demonstrates the unique and specific efficacy of the potential field module in enhancing safety and smoothness. However, lacking a global sense of macroscopic direction, M2 is easily pushed by repulsive forces into long-winded, wide-detour dead ends within dense tree clusters, causing its path length to stretch to 37.10 m under strong adversarial conditions. The complete framework M4 perfectly resolves this inherent conflict between “safety” and “length” through cross-module synthesis. With the “scaffolding” provided by the neural global sequence, redundant detours are eliminated, while fluid-like smoothness is achieved through the local potential field. At 30% density, M4 simultaneously attains the shortest path length (24.60 m) and the best smoothness (19.20 rad), while steadfastly maintaining a generous safety clearance of 0.68 m, reaching a Pareto-optimal state.

(3) Architectural Robustness Boundary and Synergistic Effect Validation (Figure 17e)

The planning success rates are the most direct piece of evidence to determine whether the performance enhancement stems from the “entire architecture” or a “single module”. At a sparse density of 10%, all four configurations yield success rates above 99.0%. As the density deteriorates to 30%, the limitations of the individual modules are clearly revealed: the success rates of M1 and M3 drop to $62.11 \pm 0.62\%$ and $71.03 \pm 0.70\%$, respectively, due to search timeouts in complex non-convex terrain. More critically, the M2 (JPS+APF) configuration suffers marked performance degradation, with its success rate decreasing to the lowest among all, at $55.25 \pm 0.60\%$. This empirically confirms, on a data level, the critical limitation of traditional artificial potential field methods—extreme susceptibility to deadlocks caused by local minima and oscillatory stagnation within dense, non-convex orchards. In contrast, the complete NeuroJPS-A framework (M4) demonstrates strong robustness under the same extreme 30% disorder density, maintaining a success rate of $98.44 \pm 0.86\%$. This performance indicates that the improved performance of NeuroJPS-A is not simply a linear superposition of the performances of its individual modules, but rather a substantial improvement brought about by the integration of the entire architecture. The global macroscopic awareness supplied by neural task sequencing effectively steers the underlying JPS search engine into the correct topological corridor in advance, thereby rendering the local potential field robust to the “local minimum trap”. In conclusion, only by activating the complete cross-domain synergy of the full NeuroJPS-A architecture can safe, smooth, and highly robust autonomous navigation be sustained in complex agricultural environments.

(4) Statistical Confidence Intervals and Significance Analysis (Figure 17, shaded bands).

To rigorously verify the repeatability and anti-interference capability of the proposed method against random environmental variations, the shaded error bands covering the trajectory curves in all subplots of Figure 17 explicitly represent the 95% confidence intervals (95% CIs), which are derived from the standard deviations summarized in Tables 5–7. As the environmental complexity gradually increases to the 30% density threshold, the confidence intervals of M1, M2, and M3 widen markedly, highlighting their growing sensitivity to random obstacle distributions. In contrast, the 95% CI of the complete M4 framework remains exceptionally narrow and tightly bounded across the entire density gradient. These narrow intervals mathematically indicate superior algorithmic predictability and parameter insensitivity, ensuring stable operation regardless of random seed fluctuations. Moreover, a two-sample *t*-test was conducted at the 30% density benchmark to evaluate the performance difference between the proposed M4 framework and its closest ablation competitor. The results showed *p*-values strictly below 0.01 ($p < 0.01$), confirming that the multidimensional superiority of the complete NeuroJPS-A architecture is statistically significant and is clearly driven by structural synergy rather than random variance.

The minimum clearance values reported in Tables 5–7 are measured in continuous physical space (meters). For each planned trajectory, we sample points at 0.1 m intervals and compute the Euclidean distance from each sample point to the nearest obstacle surface in the original (non-inflated) occupancy grid. The reported clearance is the minimum over all samples. Although a 0.6 m inflated safety margin is used during path search, the reported clearance is measured against the original obstacle boundaries, not the inflated layer, to reflect the actual physical separation and enable fair comparison with baselines that do not use inflation.

The planning times reported in Tables 5–7 refer to the same full-pipeline wall-clock time measured on the CPU, covering JPS search, APF computations, and collision checks for the ablation variants. NCO inference is included only for configurations M3 and M4. Memory usage in these tables is measured as the peak MATLAB process RSS, consistent with the measurement in Table 4. Timing measurements are averaged over 30 runs with different random seeds.

5. Practical Deployment Considerations

For real-world deployment, we recommend a lightweight sensor suite consisting of an Intel RealSense D455 stereo camera for depth perception, a Hokuyo UTM-30LX LiDAR for horizontal obstacle detection, and an IMU/GPS module for localization. As the onboard computing platform, the NVIDIA Jetson Orin NX (16 GB) or Jetson AGX Xavier is recommended. We also estimate the inference time on the Jetson platform to be less than 15 ms; however, this is a theoretical estimate, and actual testing will be conducted in future work. Furthermore, (1) APF local obstacle avoidance must run onboard the UAV (millisecond-level response, directly receiving sensor data); (2) if the onboard edge computing platform (e.g., NVIDIA Jetson Orin) has sufficient resources, the 3D-JPS global planner can also run onboard; otherwise, it can be offloaded to a ground station; and (3) NCO task sequencing and multi-UAV coordination, due to their requirement for global information, are more suitable for execution on a ground station or edge server. This hybrid deployment strategy ensures real-time safety and autonomous navigation even when communication is interrupted. Meanwhile, it should be noted that the Jetson inference time estimate (<15 ms) is a theoretical projection from component-level benchmarks rather than a measured onboard result. Actual flight tests have not yet been conducted, and real-world performance—including sensor noise, wind disturbances, and communication latency—requires further evaluation.

6. Conclusions

This paper proposes NeuroJPS-A, a hybrid framework that integrates neural combinatorial optimization, 3D Jump Point Search, and an adaptive artificial potential field for UAV path planning and obstacle avoidance in semi-structured orchard environments. The main contributions can be summarized as follows: (1) an NCO module enhanced with APF-informed edge costs, enabling obstacle-aware task sequencing; (2) a 3D-JPS global planner augmented with a potential-field correction term that tends to favor low-risk corridors; (3) a path-guided APF local controller that appears to help mitigate the local minima issue while maintaining consistency with the global path;

and (4) a closed-loop information flow that couples these modules, which differs from conventional loosely coupled hybrid planners.

Experimental results from 2D and 3D simulations suggest that NeuroJPS-A performs competitively compared to the selected baselines (A*, Theta, and ACO) under the tested conditions. In 2D high-density obstacle environments, we observed a reduction in path length and the number of inflection points relative to A*; in 3D voxel spaces, the computation time was lower than that of the comparison algorithms, and the memory footprint remained modest. Ablation studies further indicate that the neural guidance module appears to contribute to reducing search complexity, while the APF component seems to enhance trajectory smoothness and safety clearance. It should be noted, however, that these findings are derived entirely from simulation environments (MATLAB, Webots, and AirSim) and have not yet been validated through real UAV flight tests.

Several limitations should be acknowledged. First, the NCO training relies on offline simulation data, and its generalization to unseen orchard layouts or significantly different obstacle densities remains to be established. Second, the current multi-UAV coordination strategy uses a centralized conflict-resolution scheme, which may not scale well to large fleets; distributed approaches warrant further investigation. Third, the performance of the APF module may be sensitive to parameter selection, and although we have provided a sensitivity analysis under the tested scenarios, re-tuning may be required for environments with substantially different geometries. Fourth, the reported inference time on the Jetson platform is a theoretical projection rather than an onboard measurement, and actual performance under real-world sensor noise, wind disturbances, and communication latency has yet to be evaluated. We therefore regard the current findings as simulation-validated rather than field-proven.

Future work will focus on conducting real UAV flight experiments in outdoor orchards, extending the framework to fully distributed multi-UAV coordination, and investigating transfer learning to reduce the generalization gap of the NCO module across diverse environments.

Author Contributions: Conceptualization, B.C.; methodology, B.C. and M.W.; software, K.Z.; validation, B.C.; Formal analysis, P.D.; investigation, M.W.; resources, M.W.; data curation, B.C.; writing—original draft, M.W. and P.D.; writing—review and editing, B.C. and P.D.; supervision, B.C. and L.Z. (Lei Zhang); project administration, B.C. and L.Z. (Liang Zhao); funding acquisition, L.Z. (Lei Zhang), K.Z. and L.Z. (Liang Zhao). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China Youth Science Fund Project (62506119); the Science and Technology Tackling Project of Henan Province (242102211040, 232102240056), and the Natural Science Youth Fund of Henan Province (242300421439).

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Zhang, C.; Valente, J.; Kooistra, L.; Guo, L.; Wang, W. Opportunities of UAVs in orchard management. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 673–680. [\[CrossRef\]](#)
2. Zuo, K.; Cheng, X.; Zhang, H. Overview of obstacle avoidance algorithms for UAV environment awareness. *J. Phys. Conf. Ser.* **2021**, *1865*, 042002. [\[CrossRef\]](#)
3. Debnath, D.; Vanegas, F.; Sandino, J.; Hawary, A.F.; Gonzalez, F. A review of UAV path-planning algorithms and obstacle avoidance methods for remote sensing applications. *Remote Sens.* **2024**, *16*, 4019. [\[CrossRef\]](#)
4. Dechter, R.; Pearl, J. Generalized best-first search strategies and the optimality of A. *J. ACM (JACM)* **1985**, *32*, 505–536. [\[CrossRef\]](#)
5. Sonmez, A.; Kocyigit, E.; Kugu, E. Optimal path planning for UAVs using genetic algorithm. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*; IEEE: New York, NY, USA, 2015; pp. 50–55.
6. Yu, Z.; Si, Z.; Li, X.; Wang, D.; Song, H. A novel hybrid particle swarm optimization algorithm for path planning of UAVs. *IEEE Internet Things J.* **2022**, *9*, 22547–22558. [\[CrossRef\]](#)

7. Konatowski, S.; Pawłowski, P. Ant colony optimization algorithm for UAV path planning. In *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*; IEEE: New York, NY, USA, 2018; pp. 177–182.
8. Huang, T.; Fan, K.; Sun, W. Density gradient-RRT: An improved rapidly exploring random tree algorithm for UAV path planning. *Expert Syst. Appl.* **2024**, *252*, 124121. [\[CrossRef\]](#)
9. Diao, Q.; Zhang, J.; Liu, M.; Yang, J. A disaster relief UAV path planning based on APF-IRRT* fusion algorithm. *Drones* **2023**, *7*, 323. [\[CrossRef\]](#)
10. Song, Q.; Zhao, Q.; Wang, S.; Liu, Q.; Chen, X. Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization. *IEEE Access* **2020**, *8*, 62107–62115. [\[CrossRef\]](#)
11. Gao, J.; Xu, X.; Pu, Q.; Petrovic, P.B.; Rodić, A.; Wang, Z. A hybrid path planning method based on improved A* and CSA-APF algorithms. *IEEE Access* **2024**, *12*, 39139–39151. [\[CrossRef\]](#)
12. Zhang, Z.; Liu, X.; Feng, B. Research on obstacle avoidance path planning of UAV in complex environments based on improved Bézier curve. *Sci. Rep.* **2023**, *13*, 16453. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Zhao, L. 3D Obstacle Avoidance for Unmanned Autonomous System (UAS). Master's Thesis, University of Nevada, Las Vegas, Las Vegas, NV, USA, 2015.
14. Zheng, L.; Zhang, P.; Tan, J.; Li, F. The obstacle detection method of uav based on 2d lidar. *IEEE Access* **2019**, *7*, 163437–163448. [\[CrossRef\]](#)
15. Ross, S.; Melik-Barkhudarov, N.; Shankar, K.S.; Wendel, A.; Dey, D.; Bagnell, J.A.; Hebert, M. Learning monocular reactive uav control in cluttered natural environments. In *2013 IEEE International Conference on Robotics and Automation*; IEEE: New York, NY, USA, 2013; pp. 1765–1772.
16. Stanoev, A.; Audinet, N.; Tancock, S.; Dahnoun, N. Real-time stereo vision for collision detection on autonomous UAVs. In *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*; IEEE: New York, NY, USA, 2017; pp. 1–6.
17. Tu, G.-T.; Juang, J.-G. UAV path planning and obstacle avoidance based on reinforcement learning in 3d environments. *Actuators* **2023**, *12*, 57. [\[CrossRef\]](#)
18. Yang, F.; Fang, X.; Gao, F.; Zhou, X.; Li, H.; Jin, H.; Song, Y. Obstacle avoidance path planning for UAV based on improved RRT algorithm. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 4544499. [\[CrossRef\]](#)
19. Fedele, G.; D'Alfonso, L.; Chiaravallotti, F.; D'Aquila, G. Obstacles avoidance based on switching potential functions. *J. Intell. Robot. Syst.* **2018**, *90*, 387–405.
20. Hao, G.; Lv, Q.; Huang, Z.; Zhao, H.; Chen, W. UAV path planning based on improved artificial potential field method. *Aerospace* **2023**, *10*, 562. [\[CrossRef\]](#)
21. Wu, X.; Wang, D.; Wen, L.; Xiao, Y.; Wu, C.; Wu, Y.; Yu, C.; Maskell, D.L.; Zhou, Y. Neural combinatorial optimization algorithms for solving vehicle routing problems: A comprehensive survey with perspectives. *arXiv* **2024**, arXiv:2406.00415.
22. Debnath, D.; Vanegas, F.; Boiteau, S.; Gonzalez, F. An integrated geometric obstacle avoidance and genetic algorithm tsp model for uav path planning. *Drones* **2024**, *8*, 302. [\[CrossRef\]](#)
23. Bederina, H. Optimizing UAV Trajectories via a Simplified Close Enough TSP Approach. *arXiv* **2025**, arXiv:2507.03775.
24. Silva Arantes, J.D.; Da Silva Arantes, M.; Toledo, C.F.M.; Júnior, O.T.; Williams, B.C. Heuristic and genetic algorithm approaches for UAV path planning under critical situation. *Int. J. Artif. Intell. Tools* **2017**, *26*, 1760008. [\[CrossRef\]](#)
25. Garmendia, A.I.; Ceberio, J.; Mendiburu, A. Neural combinatorial optimization: A new player in the field. *arXiv* **2022**, arXiv:2205.01356.
26. Bai, X.; Jiang, H.; Cui, J.; Lu, K.; Chen, P.; Zhang, M. UAV path planning based on improved A* and DWA algorithms. *Int. J. Aerosp. Eng.* **2021**, *2021*, 4511252.
27. Luo, Y.; Lu, J.; Zhang, Y.; Qin, Q.; Liu, Y. 3D JPS path optimization algorithm and dynamic-obstacle avoidance design based on near-ground search drone. *Appl. Sci.* **2022**, *12*, 7333.
28. Joshi, C.K.; Cappart, Q.; Rousseau, L.-M.; Laurent, T. Learning the travelling salesperson problem requires rethinking generalization. *Constraints* **2022**, *27*, 70–98. [\[CrossRef\]](#)

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.