

Efficient and Near-optimal Global Path Planning for AGVs: A DNN-Based Double Closed-Loop Approach with Guarantee Mechanism

Runda Zhang, Runqi Chai, *Senior Member, IEEE*, Kaiyuan Chen, *Member, IEEE*, Jinning Zhang, Senchun Chai, *Senior Member, IEEE*, Yuanqing Xia, *Fellow, IEEE*, and Antonios Tsourdos, *Member, IEEE*

Abstract—In this paper, a novel global path planning approach with rapid convergence properties for autonomous ground vehicles (AGVs) named neural sampling rapidly-exploring random tree (NS-RRT*) is proposed. This approach has a three-layer structure to obtain a feasible and near-optimal path. The first layer is the data collection stage. Utilizing the target area adaptive rapidly-exploring random tree (TAA-RRT*) algorithm to establish a collection of paths considering the initial noise disturbance. To enhance network generalization, an optimal path backward generation (OPBG) strategy is introduced to augment the dataset size. In the second layer, the deep neural network (DNN) is trained to learn the relationships between the states and the sampling strategies. In the third layer, the trained model is used to guide RRT* sampling, and an efficient guarantee mechanism is also designed to ensure the feasibility of the planning task. The proposed algorithm can assist the RRT* algorithm in efficiently obtaining optimal or near-optimal strategies, significantly enhancing search efficiency. Numerical results and experiments are executed to demonstrate the feasibility and efficiency of the proposed method.

Index Terms—Path planning, autonomous ground vehicles (AGVs), deep neural network (DNN).

I. INTRODUCTION

IN recent years, research on path planning methods for autonomous ground vehicles has significantly advanced, leading to diverse applications in various real-world scenarios [1]–[3]. The primary objective of motion planning for these autonomous ground vehicles is to generate an unobstructed, feasible path connecting a designated start point to a predetermined end point [4], [5]. Evaluating the efficiency of such path planning algorithms serves as a crucial metric for assessing the effectiveness of different planning methodologies.

Motion planning methodologies for autonomous systems can be primarily categorized into three distinct approaches: graph-based methods, population-intelligence-based methods, and sampling-based methods. Graph-based techniques involve partitioning the map into grids and subsequently selecting the optimal path by assessing the cost associated with each path node relative to the start and end points [6]. Renowned instances of these methods encompass the Dijkstra algorithm [7] and the A* algorithm [8], [9], with several others also being noted in the literature [10]. Population-intelligence-based methods derive their foundation from mimicking certain cooperative

behaviors observed in natural populations to navigate optimal paths. Notable examples within this category are particle swarm optimization (PSO) algorithms [11], [12], ant colony optimization (ACO) algorithms [13], and wolf pack optimization algorithms [14]. The underlying principle of these algorithms is to leverage collective behaviors found in nature to facilitate pathfinding [15]. Sampling-based methods entail discretizing the environment into a nodal space and subsequently identifying the most favorable configuration from the commencement to the termination points [16]. Prominent representatives of this category include the probabilistic roadmap (PRM) [17] and the rapidly-exploring random tree (RRT) [18]. It is worth noting that while graph-based and population-intelligence-based methods often grapple with efficiency challenges in intricate maps, sampling-based strategies exhibit a commendable performance, retaining efficacy even in high-dimensional spaces.

In the realm of sampling-based methods, computational efficiency serves as a critical metric for evaluating algorithmic performance [19]–[21]. Considerable research efforts have been directed towards refining these sampling strategies to enhance computational efficiency. For instance, the study by [22] introduced a generalized Voronoi diagram technique to establish a non-uniform sampling area. This work further incorporated an adaptive step size to bolster the computational efficiency of the RRT* algorithm. Similarly, in the research presented by Chi et al. [23], the authors employed feature nodes of the map as guiding elements to inform the sampling process in RRT*. In [24], the authors achieved efficient path planning by utilizing bidirectional RRT and assisting metrics. In [25], the RRT* algorithm was realised by designing multiple subtrees to efficiently explore the environment, which in turn achieved search efficiency. In [26], the authors combined the artificial potential field strategy and the direct connection strategy to guide RRT* sampling while efficiently connecting nodes to achieve robotic path planning. In [27], for the narrow space, the authors utilized the introduction of ellipsoid connection node strategy and slow informed guided sampling method to achieve robotic trajectory planning in the narrow space. This innovation has proven effective in achieving more efficient path planning outcomes. Thus, advancements in optimizing sampling-based methods remain an active and significant area of research. Although the above work has improved the efficiency of sampling-based methods to some extent, it is still evident that the probabilistic and blind nature of sampling-based search cannot be avoided.

Recently, neural networks have garnered substantial attention for their robust learning capabilities [28], [29]. Several path-planning methodologies have been integrated with neural networks, leading to notably enhanced outcomes [30], [31]. For instance, Chai leveraged deep neural networks (DNNs) to learn the nonlinear mapping relationships between optimal states and sampling strategies, thereby facilitating real-time trajectory planning for hypersonic vehicles [32]. Considering the collision probability, the literature [33] used neural networks to predict the safety state at each extension. In [34] the au-

R. Zhang, R. Chai, S. Chai and Y. Xia are with the school of Automation, Beijing Institute of Technology, Beijing, China, e-mail: (runda.zhang@bit.edu.cn), (r.chai@ieee.org), (chaic97@bit.edu.cn), (xia_yuanqing@bit.edu.cn).

K. Chen is with Vanke School of Public Health, Tsinghua Institute for Healthy China, Tsinghua University, Beijing, China, e-mail: (kaiyuanchen@mail.tsinghua.edu.cn).

J. Zhang is with the School of Engineering, University of Leicester, UK, e-mail: (jz388@leicester.ac.uk)

A. Tsourdos is with the School of Aerospace, Transport and Manufacturing, Cranfield University, UK, e-mail: (a.tsourdos@cranfield.ac.uk). (Corresponding authors: Runqi Chai and Kaiyuan Chen)

thors used neural networks to predict the sampling region of the RRT for non-uniform and efficient sampling planning. However, it should be noted that the prevailing approaches employing neural networks for trajectory generation predominantly follow a unidirectional open-loop paradigm. Consequently, this poses challenges in ensuring and quantifying the quality of the generated trajectories.

Building on the aforementioned methodologies, we introduce an innovative, efficient, and direct sampling-based motion planning approach. Firstly, we create a large dataset, and construct a deep neural network (DNN) to learn the nonlinear mapping relationships between sampled states and corresponding sampling strategies. This allows the network to learn an ability to accurately guide the sampling algorithm for sampling. Unlike traditional approaches such as the classic RRT* algorithm, where improvement is iteratively achieved through probabilistic sampling, our method allows the sampling planner to make more informed decisions about the next sampling point in current state.

The main contributions of this paper are as follows:

- 1) A novel hierarchical path planning method consisting of three components: data preparation, data training, and online planning, is proposed. Unlike conventional non-uniform samplers [22], a trained neural sampler is adopted to perform the sampling process of RRT*. The neural sampler can accurately sample the next optimal or near-optimal state based on the current state, avoiding extensive random sampling processes, thereby greatly enhancing search efficiency.
- 2) During the data generation phase, an optimal path reverse generation strategy is proposed to increase the scale of the dataset, thereby further enhancing the approximation performance of the network. Unlike directly generating the dataset, such a strategy renders the generation process efficient.
- 3) In the planning stage, an efficient guarantee mechanism is designed to ensure the completion of planning tasks. The entire third layer is a double closed-loop structure, ensuring the feasibility of the method.

The remaining sections are organized as follows. In Section II, a general form of the path planning problem is formulated. In Section III, the framework of the proposed hierarchical path planning method is presented. In Section IV, the effectiveness of the proposed method is verified by simulations and experiments. Finally, Section V provides concluding remarks.

II. PROBLEM FORMULATION

A. Definitions

In this section, we describe the path planning problem, specifying it as an optimal control problem form.

In order to clearly formulate the general form of the path planning problem, we specify the entire map as a configuration space. The whole configuration space is denoted by $\mathcal{X}$, the obstacle space by $\mathcal{X}_{obs}$ and the obstacle-free space by $\mathcal{X}_{free}$ ($\mathcal{X}_{free} = \mathcal{X} \setminus \mathcal{X}_{obs}$). The aim of path planning is to find a feasible trajectory σ in the obstacle-free space, while the path needs to connect the start and end points, i.e. .

$$\begin{cases} \sigma : [t_0, t_f] \longrightarrow \mathcal{X}_{free} \\ \sigma(t_0) = x_{start} \\ \sigma(t_f) = x_{goal} \end{cases} \quad (1)$$

where x_{start} and x_{goal} indicate the starting position and the goal position, respectively. To achieve path optimality, we formulate the

path planning problem as an optimal control problem as follows:

$$\begin{aligned} \min \quad & Cost(\sigma) \\ \text{s.t.} \quad & \forall t \in [t_0, t_f], \sigma \in \Sigma \\ & \sigma(t_0) = x_{start} \\ & \sigma(t_f) = x_{goal} \\ & \sigma(t) \in \mathcal{X}_{free}. \end{aligned} \quad (2)$$

where Σ denotes the set of all feasible paths and $Cost(\sigma)$ denotes the path cost.

B. Review of RRT* Algorithm

The RRT* path planning algorithm, being an incremental method, is devised to acquire optimal paths through continuous random sampling in the environment. In each sampling iteration, a point x_{rand} is first randomly sampled within a feasible region $\mathcal{X}_{free}$ in the map environment. Then a node $x_{nearest}$ on the tree structure that is less costly to compute with its direct computation is searched for in the circular domain centred on this point with the parameter η as the radius. The steering function is then used to connect x_{rand} and $x_{nearest}$. At the same time, obstacle detection is performed on the connection between these two, and x_{new} is added to the tree structure if it passes. Unlike normal RRT, RRT* has a rewire mechanism that tries to optimize the path by changing the tree structure each time in the circular domain. Until the end of the iteration loop, the path is finally obtained.

As the complexity of the environment continues to increase, RRT* will become inefficient. Although some studies have proposed solutions to the low efficiency problem [22], [23], there are still some drawbacks in the improved algorithms.

- **Difficulty 1:** (Sampling Blindness) Although in [22], [23] the RRT* sampling is restricted to the effective area, it can still be seen that the optimal or near-optimal path is found by continuously sampling within the region. The sampling still remains blind.

To solve the aforementioned problem and enable the sampler to clearly know where to sample, a neural sampler is introduced in this paper to address this issue. Sampling algorithms may encounter a phenomenon where connections to superior sampling nodes may fail during the sampling process. Specifically, nodes sampled may reach a superior state, but the straight-line connections between two points might penetrate obstacles. Consequently, traditional sampling algorithms necessitate resampling to new feasible points and then continuing expansion, resulting in inefficient outcomes. As illustrated in Fig. 1, the purple node n_k and several points n_p are both in a superior position. However, the connection n_k traverses an obstacle, preventing node n_k from being added to the tree structure, and instead, n_p is found through extensive sampling. This leads to the difficulty 2 to be discussed next:

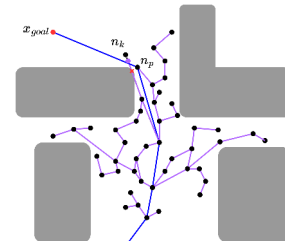


Fig. 1: Random sampling extension process of RRT* algorithm.

- **Difficulty 2:** (Infeasibility of Connections) The RRT* algorithm experiences path invalidation due to crossing obstacles after node connections through straight line.

This is due to the uncertainty of sampling by the neural sampler, which will affect the feasibility of the method. Therefore, this paper designs a double closed-loop switching structure to ensure the feasibility of the method. This paper aims to propose a more efficient improvement scheme for the RRT* planning algorithm, providing solutions for its application.

III. HIERARCHICAL PATH PLANNING FRAMEWORK

In this section, we introduce the proposed hierarchical motion planning framework. This framework is a three-layer structure, where the first layer uses the method in [22] to generate the optimal set of paths. Then calculate the sampling strategies of the optimal path set. The second layer uses a DNN network to learn the mapping relationships between the states and the optimal sampling strategies. In the third layer, the trained network model is used to guide the RRT* algorithm sampling, thus realizing the direct path planning of the RRT* algorithm.

A. Path Set Generation

To generate the optimal path set, we use the TAA-RRT* path planning algorithm from [22] to generate paths. TAA-RRT* relies on the well-established classical RRT* algorithm and has been proven to be asymptotically optimal. We utilized a number of sampling points when generating the dataset to obtain high-quality paths. Simultaneously, in order to enrich the dataset and enable the network to learn path planning sampling strategies from a certain region to a specific target point $(p_x(f), p_y(f))$, we dynamically varied the starting point for each data instance. Consequently, random perturbations (δ_x, δ_y) were applied to the starting point $(p_x(0), p_y(0))$, as indicated in line 4 of Algorithm 1 during the generation of each dataset. Step 2 is a detection mechanism that verifies whether the modified starting point lies within obstacles.

In order to allow the neural network to learn the optimal sampling strategy so that it can guide the RRT* algorithm sampling, after obtaining the optimal or near-optimal paths, we calculate the optimal sampling strategies corresponding to each node. The geometric interpretation of the sampling strategy is depicted in Fig. 2. The specific dataset generation process is summarized in Algorithm 1. After computation, pathset $\mathbb{D}$ containing only states (x, y) and sampling policies $(u(1), u(2))$ is obtained.

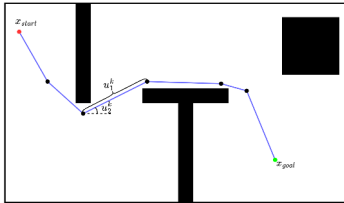


Fig. 2: sampling strategies at path nodes.

Although paths can be generated efficiently using the TAA-RRT* algorithm, a large number of sampling points are still required in order to obtain high quality paths. As a result, a large amount of computational resources are required to generate a large dataset. In this paper, an optimal path backward generation (OPBG) strategy is proposed to expand the size of the dataset. The detailed process of the OPBG strategy is shown in Algorithm 2. For the paths within the dataset, we perform positional variations on intermediate nodes to construct new paths. Simultaneously, feasibility and accuracy checks

Algorithm 1 Dataset Generation

Input: Number of path sets N_p , map.

Output: $\mathbb{D}$

- 1: Global parameter initialization.
- 2: Map initialization and generalized voronoi diagram (GVD) processing, and then building non-uniformly sampled areas.
- 3: **while** $\|\mathbb{D}\| \leq N_p$ **do**
- 4: Step 1: Set the starting position with random perturbations δ_x and δ_y .
- 5: Step 2: If the starting point is within $\mathcal{X}_{free}$, proceed; otherwise, return to Step 1.
- 6: Step 3: Execution of the TAA-RRT* algorithm.
- 7: Step 4: Calculate sampling strategies u^* of path nodes if the path σ^* is found.
- 8: **for** $k = 1 : |\sigma^*| - 1$

$$\begin{aligned} u_k^*(1) &= \|\sigma_{k+1}^* - \sigma_k^*\| \\ u_k^*(2) &= \arctan\left(\frac{\sigma_{k+1}^*(2) - \sigma_k^*(2)}{\sigma_{k+1}^*(1) - \sigma_k^*(1)}\right) \end{aligned} \quad (4)$$

end for

- 9: Step 5: Store the states and sampling strategies to the dataset.

$$\mathbb{D} = \mathbb{D} \cup (\sigma^*, u^*) \quad (5)$$

10: **end while**

11: **return** $\mathbb{D}$.

are conducted to rapidly obtain valid paths, ultimately achieving the goal of dataset expansion.

B. Network Construction and Training

In this study, the selection of network architecture is determined by the scale of the set of pathways. The data volume of the pathway set involved in this study is small-scale. In comparison to Transformers, DNNs demonstrate better learning performance on small-scale datasets. Therefore, DNNs are adopted as neural samplers in this study.

1) *Data Normalization:* After the dataset is collected, the data needs to be normalized to make the network learning and training process converge quickly. This is to avoid negative effects on the convergence of the network due to some variables in the dataset being too large or too small. The states and sampling strategies in the dataset are normalized in the following way:

$$\begin{aligned} \bar{p}_x &= p_x / p_x^{max}, \bar{p}_y = p_y / p_y^{max} \\ \bar{u}_1 &= u_1 / u_1^{max}, \bar{u}_2 = u_2 / u_2^{max} \end{aligned} \quad (10)$$

where p^{max} represents the maximum value of the state and u^{max} represents the maximum value of the sampling strategies.

2) *DNN Network Construction:* In order to know exactly the sampling strategies of the RRT* algorithm, rather than finding such sampling points by random sampling, in this section we train a DNN network to guide the RRT* algorithm sampling. We expect a DNN network to map the non-linear relationships between states and sampling strategies during RRT* sampling. The DNN planner can output the following optimal or near-optimal sampling strategies in real-time based on the current sampling state.

$$u^k = N(p^k) \quad (11)$$

DNN structurally have an input layer, a hidden layer and an output layer. Where the hidden layer is a layer consisting of multiple

Algorithm 2 Process of the OPBG**Input:** The optimal path sets $\mathbb{D}$, expected scale N_D .**Output:** $\mathbb{D}$, $\mathbb{D}_{tra}$, $\mathbb{D}_{tes}$, $\mathbb{D}_{val}$

- 1: **while** $|\mathbb{D}| \leq N_D$
- 2: Step 1: Randomly select a path σ from the $\mathbb{D}$.
- 3: Step 2: Randomly select r nodes $(n_1, n_2, \dots, n_r)$ on σ other than the end point, r is a random number that can be computed by $r = \text{randi}(|\sigma| - 1)$.
- 4: Step 3: Horizontal and vertical variations were performed for each of the selected nodes. Form a new path σ_{new} with the remaining nodes.
- 5: **for** $i = 1 : r$

$$n'_{ix} = n_{ix} + \zeta_x, n'_{iy} = n_{iy} + \zeta_y \quad (6)$$

- 6: **end for**
- 7: where ζ_x and ζ_y denote randomly altered values in the horizontal and vertical directions, respectively.
- 8: Step 4: Perform obstacle detection and accuracy testing on this new path.

$$\sigma_{new} \in \mathcal{X}_{free} \ \& \ \|Length(\sigma_{new}) - Length(\sigma)\| \leq \varepsilon \quad (7)$$

where the first equation functions as obstacle detection and the second is accuracy detection. $Length(\cdot)$ denotes the function that calculates the length of the path. ε indicates the precision value.

- 9: Step 5: If (7) is satisfied, compute the corresponding sampling strategies u and add both to the dataset. Otherwise, return to Step 3.

$$\mathbb{D} = \mathbb{D} \cup (\sigma_{new}, u_{new}) \quad (8)$$

- 10: **end while**
- 11: Dividing the dataset $\mathbb{D}$ into training set $\mathbb{D}_{tra}$ (70% of $\mathbb{D}$), test set $\mathbb{D}_{tes}$ (20% of $\mathbb{D}$) and validation set $\mathbb{D}_{val}$ (10% of $\mathbb{D}$).
- 12: **return** $\mathbb{D}$, $\mathbb{D}_{tra}$, $\mathbb{D}_{tes}$, $\mathbb{D}_{val}$.

neurons. Assume that there are N_l network layers and each layer has N_n neuron nodes. Then the output of the j th node in layer i is

$$o_{ij} = f(\sum \omega_j x_{ij} + b_{ij}) \quad (12)$$

where ω_j denotes the weight coefficient, x_{i-1} denotes the input vector at layer $i - 1$, and b_{ij} denotes the bias vector. f denotes the activation function that performs a linear transformation of the weighted input. For the constructed network we apply the Tanh activation function.

In the network training phase, We use the stochastic gradient descent (SGD) method to train the weight coefficients and bias of the network so as to minimise the loss function as follows:

$$Loss = \frac{1}{N_D} \sum_{i=1}^{N_D} (x_i - \hat{x}_j)^2 \quad (13)$$

where N_D indicates the size of the data set, x_i and $\hat{x}_j$ indicate the actual output and reference output values, respectively.

C. NS-RRT* Sampling

Once the network is trained, the trained model is used to guide the RRT* sampling. Different from the classic RRT* algorithm, the random sampling process is replaced by neural sampling. The exact flow of the algorithm is described in Algorithm 3.

Firstly, the neural network output a near-optimal sampling strategies according to the current states, and we can calculate the next states based on the sampling strategies. Secondly, obstacle detection is performed for the newly generated nodes as well as for the

Algorithm 3 NS-RRT***Input:** x_{start}, x_{goal}, Map **Output:** path σ

- 1: Initialization parameters, initialize the tree nodes and add the starting point to the tree. Set the current node x_{cur} as the starting point.
 - 2: $k = 1$
 - 3: /*main loop*/
 - 4: **while** not find a path
 - 5: The neural network outputs an optimally sampling strategies u_k based on the current node x_{cur} .
 - 6: Calculate the state of the new sampling point x_{new} based on the sampling strategies u_k .
- $$x_{new} = x_{cur} + u_k \quad (9)$$
- 7: Obstacle detection for the x_{cur} and x_{new} .
 - 8: **if** accessible **then**
 - 9: Connects the current node x_{cur} to the x_{new} .
 - 10: Adds the new node x_{new} to the tree, updating the current node x_{cur} .
 - 11: **else**
 - 12: /* guarantee mechanism */
 - 13: **while** x_{new} in $\mathcal{X}_{obs}$
 - 14: Generate a new sampling strategy u' based on x_{new} .
 - 15: Compute and update the new node x_{new} by $x_{new} = x_{new} + u'$.
 - 16: Update x_{new} to x_{goal} if x_{new} is in the region near the goal point x_{goal} .
 - 17: **end while**
 - 18: Calculates the sampling circle in which the current point x_{cur} is located.
 - 19: Delineate the new sampling area Ω from x_{cur} to x_{new} .
 - 20: Randomly sample among Ω and find the optimal connection σ' between x_{cur} and x_{new} .
 - 21: Add the nodes in σ' to the tree.
 - 22: Update the current sampling point x_{cur} .
 - 23: **end if**
 - 24: **end while**
 - 25: **return** σ

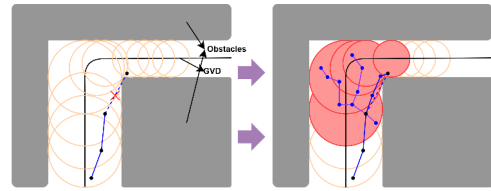


Fig. 3: Failure of the neural sampler and reconnection of the guarantee mechanism.

connections. Finally, to avoid the problem of infeasible paths due to the neural network approach, a guarantee mechanism is designed. Lines 12-22 of Algorithm 3 are the proposed guarantee mechanism. After entering the guarantee mechanism, as illustrated in Fig. 3, the mechanism selects a local region Ω (the pink region in Fig. 3) connecting x_{cur} and x_{new} . Subsequently, it continuously expands the new tree structure to establish effective connections between nodes. The black lines in Fig. 3 represent GVD nodes, and the light yellow region corresponds to obstacle-free areas [22]. After the connection detection failure, minimum sampling areas Ω are first determined and then suitable path connections are found by random sampling. When the neural sampler samples within obstacles, the handling of

the guarantee mechanism corresponds to lines 13-17 of Algorithm 3. At this point, the next sampling point will be continuously computed based on the current sampling point until it appears in the feasible region. Then local sampling will be conducted for connection. Thus, the feasibility and stability of the method is guaranteed.

D. Overall Planning Framework

The overall framework of the method is shown in Fig. 4. The Offline Dataset Generation is the dataset collection process, the Offline Network Training is the network learning training process. The Online Planning is the actual planning process. The trained model will be used to guide the sampling process.

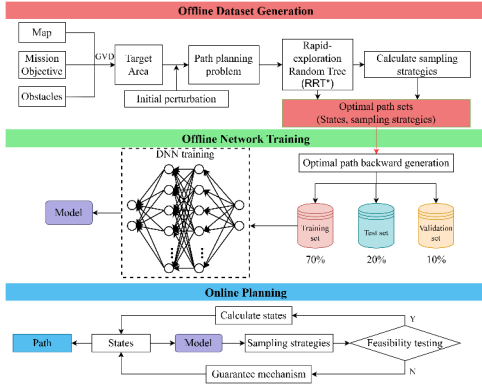


Fig. 4: Overall framework.

Of particular note is that, as illustrated in Fig. 4, the methodology proposed is structured into offline and online components. Unlike the offline planning methods discussed in reference [2], the offline component in this study pertains to the offline preparation phase. In fact, planning still heavily relies on online procedures. The offline phase of work will affect the effectiveness and quality of online planning, significantly enhancing the online planning speed of the RRT* algorithm. The online planning layer is a double loop structure, determining entry into either loop based on the feasibility testing results during each run. The input of the online planning layer is the current state of the AGV (i.e., its position), and then the planning framework continuously plans based on the inputs. The upper loop facilitates the planning process rapidly, while the lower loop ensures effective node connectivity by employing minimal sampling nodes in the local region. Such a double loop structure not only pursues efficiency but also guarantees the feasibility of the method.

E. Theoretical Propertie

1) Completeness:

Theorem 1: For a global path planning problem, if a feasible solution exists, the probability that the neural sampling planner with the guarantee mechanism mentioned in this paper finds a feasible path, $\mathcal{P}(x_{start}, x_{goal}, \mathcal{X}_{free}) = 1$.

Proof 1: Theorem 1 indicates that if a feasible solution exists, the neural sampling planner is guaranteed to find a feasible solution. For the neural sampling planner in this paper, the neural sampler determines the next near-optimal state based on the current state. Considering the worst case, assuming that the neural network fails to sample in each attempt (sampling nodes are in obstacles or connection failures occur), the safety mechanism will continuously sample in local regions to find feasible solutions connecting the start and goal points. Since the establishment of local regions is based on generalized voronoi paths, which are effective paths connecting the

start and goal points, feasible paths are guaranteed to exist. Therefore, even in the case of complete failure of neural sampling, the method can find a feasible solution. In the case of partial failure of the neural sampler, the safety mechanism can still ensure the feasibility of the path by guaranteeing effective connections between local nodes. In summary, the proposed neural sampling planner with a guarantee mechanism is complete. ■

2) Optimality Discussion: This work employs DNN to learn near-optimal or optimal sampling strategies from a set of paths to guide RRT* sampling exploration. For the path set, TAA-RRT* algorithm is utilized to rapidly generate the path set. The TAA-RRT* algorithm samples within the homotopy region of the optimal path, which differs from the traditional RRT* algorithm's strategy of sampling the entire graph, but the extension mechanism remains the same. As the RRT* algorithm has been proven to possess asymptotic optimality [35], the TAA-RRT* algorithm inherits the asymptotic optimality of the RRT* algorithm, thereby ensuring the quality of the path set. Asymptotic optimality is an important property based on the sampling algorithm, indicating that the path is optimal as the number of sampled points tends to infinity. This work designs a neural sampler to guide the sampling algorithm for acquiring near-optimal or optimal paths with minimal nodes, thus the aim of this work is to rapidly obtain a near-optimal path based on the RRT* sampling algorithm. Therefore, in cases where the number of sampled points is not abundant, the property of asymptotic optimality does not apply to the method proposed in this paper. The method proposed is more akin to quickly generating a high-quality initial path compared to the RRT* algorithm. If one desires to imbue it with asymptotic optimality, further sampling can be conducted after obtaining the path to continuously improve it. However, in general, the solutions obtained by the NS-RRT* algorithm are near-optimal and require fewer computational costs, which is crucial in practice, eliminating the necessity to sample a large number of nodes to improve the path.

Remark 1: Similar to the algorithms presented in references [22] and [34], the proposed method in this paper belongs to a global planning approach and does not involve dynamic obstacle avoidance. When encountering static obstacles or dynamic obstacles in the environment, a replanning strategy can be employed, or existing local path planning methods can be utilized for obstacle avoidance.

Remark 2: This paper employs DNN for RRT* samplings. The focus of this study lies not in generalization but in utilizing neural networks to accelerate the sampling process of RRT*, aiding in its rapid convergence. Considering future practical applications, targeted data collection and training can be conducted for different path planning tasks. Moreover, in real-world applications, an incremental learning strategy can be employed for extensive expansion and application.

Remark 3: The focus of this work is to rapidly search for a feasible near-optimal path, with no guarantee of achieving the optimal solution. Even in practical scenarios, sampling-based algorithms do not expend significant computational time in pursuit of the optimal path. The solutions obtained in this study are near-optimal and can be efficiently searched, which is crucial in practical applications.

IV. SIMULATION AND EXPERIMENT RESULTS

A. Simulation Setup

In this section, we verify the feasibility and effectiveness of the proposed hierarchical planning framework for path planning through numerical simulations. The numerical simulation results were executed in Matlab R2022b on Windows 11, and our computer is configured as Intel-i9 with 16G RAM memory.

As shown in Table I, x and y represent the length and width ranges of the spatial map, indicating that the simulated map size

is uniformly 500×500 . d_{min} and d_{max} denote the minimum and maximum step sizes, respectively, for the RRT* algorithm during dataset generation. r_{rewire} signifies the re-connection radius for the RRT* algorithm. δ and ζ represent the initial perturbation values during dataset generation and the node perturbation values during backward path generation, respectively. ε denotes the threshold for path accuracy detection. η represents the precision at which the network training converges. N_D denotes the number of paths in the dataset.

TABLE I: Related Parameter Settings.

Parameter	Value	Parameter	Value	Parameter	Value
x, m	[0,500]	y, m	[0,500]	ε, m	1.5
d_{min}, m	20	d_{max}, m	35	η	10^{-5}
δ_x, m	(0,0.8)	δ_y, m	(0,0.8)	r_{rewire}, m	40
ζ_x, m	(0,10)	ζ_y, m	(0,10)	N_D	30000

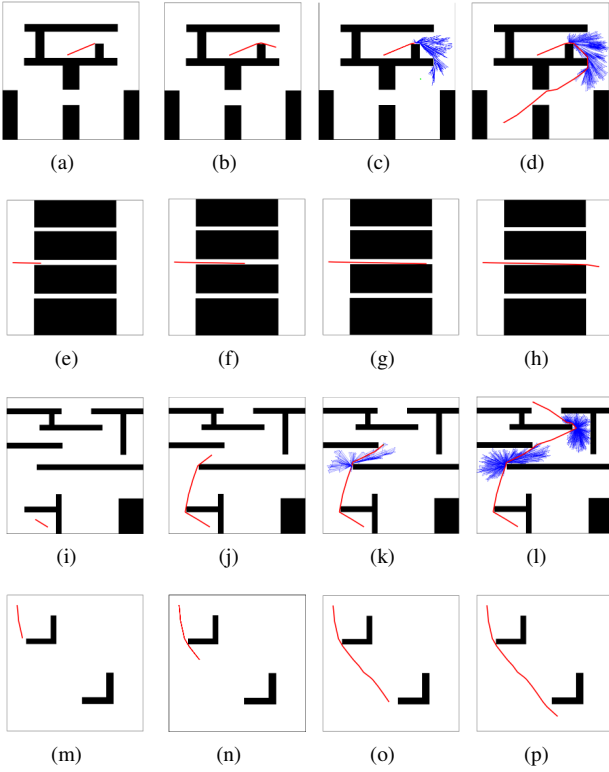


Fig. 5: The planning process of the proposed method in four scenarios. The four scenarios are trap spaces, narrow passages, mazes and open spaces. Each row from left to right indicates the path forward expansion process of the proposed method.

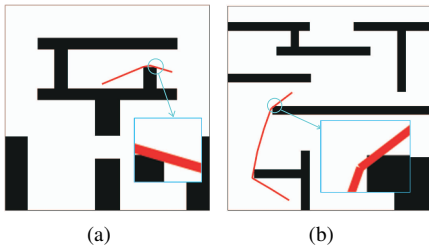


Fig. 6: The case of infeasible paths due to neural sampling.

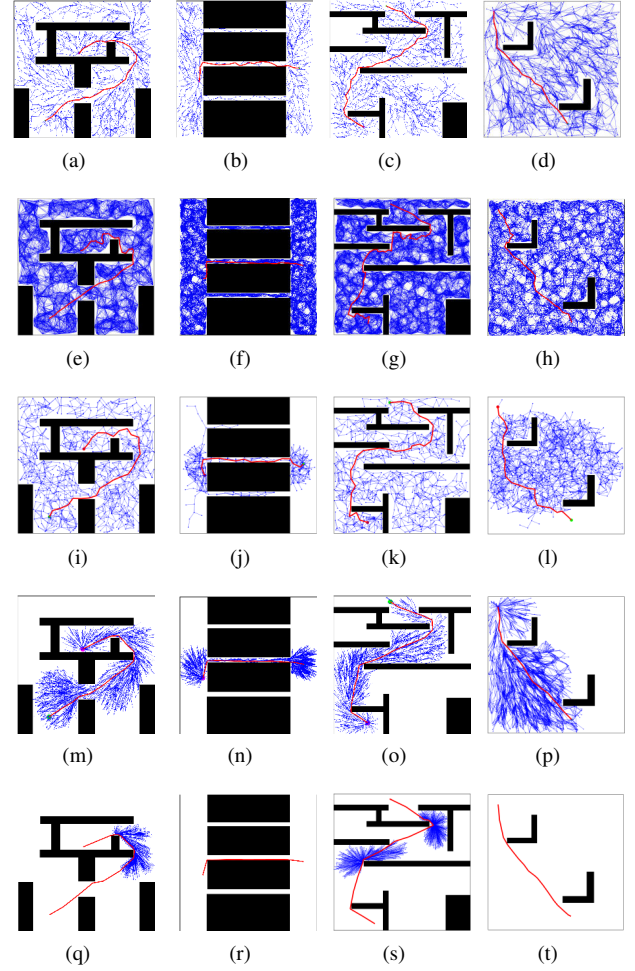


Fig. 7: Compare the performance of other algorithms in four scenarios. From up to down in the columns, the sampling algorithms are RRT*, PRM, Informed-RRT*, TAA-RRT* and NS-RRT*. The red route is the path found by the algorithm.

B. Results in Four Scenarios

In the simulation experiments, we verified the feasibility of the proposed method in four scenarios. These four scenarios represent trap spaces, narrow passages, mazes, and open environments. In each scenario, collect the corresponding set of paths for the path planning task and train the respective network model. Subsequently, utilize the model to guide RRT* sampling for the corresponding task.

First, the path set is generated by combining efficient path planning algorithms with OPBG. To achieve the desired scale of the dataset, the TAA-RRT* algorithm is employed to generate the path set, followed by the utilization of the OPBG strategy to expand the path set. The generation rate has been greatly enhanced. Taking Map 1 as an example, with a total of 1000 data generated, compared to direct generation, the speed using the strategy proposed in this paper increased by 64.3%. To achieve better guidance efficacy, specific paths set need to be generated for each map for individualized training purposes. After generation, the model is then trained.

The convergence of the sets of paths for different tasks in various maps is illustrated in Fig. 8. The network converges around 1000 iterations. At intervals of a certain number of iterations, testing is conducted on the validation set utilizing the trained model. As depicted in the Fig. 8, the validation set also gradually converges,

thus affirming the efficacy of the model. Then, upon completion of training, they can be directly utilized in the online planning process. Moreover, in the future, incorporating strategies such as incremental learning and learning-on-the-fly can significantly enhance the efficiency of the overall framework.

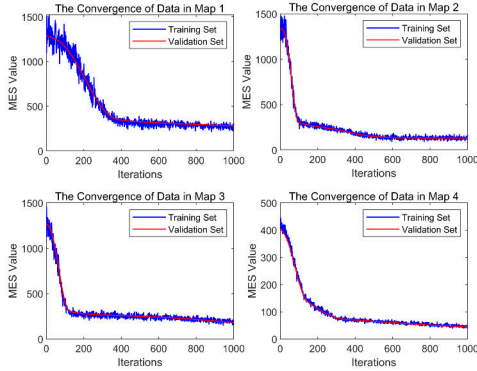


Fig. 8: Convergence status of networks under different tasks.

Fig. 5(a)-(d) provide a comprehensive depiction of the sampling mechanism applied in Scenario 1. The process initiates from a predefined starting location, with the subsequent path expansion governed by the neural sampler, as demonstrated in Fig. 5(a). For every iteration of the neural sampler's sampling, followed by its integration with antecedent nodes, an obstacle recognition subroutine is triggered. Upon identification of non-viable connections, exemplified in Fig. 5(b), such links are systematically discarded. Thereafter, local sampling planning is instituted, leveraging our newly proposed efficiency-guarantee mechanism, as represented in Fig. 5(c). This sequence culminates in implementing an optimized path planning strategy tailored to the stipulated scenario.

Fig. 6 shows the path infeasibility that may arise when using the neural sampler. In the process of using the network model to guide the sampling, since there is a certain distance between the sampling points, there will be situations where the path passes through obstacles. Fig. 6 corresponds to Fig. 5(b) and 5(j). At this time, the guarantee mechanism will correct the path.

It can be seen that the path planning task is completed in all four scenarios. When an infeasible connection is encountered in Scenario 1 and Scenario 3, the guarantee mechanism is used to make the planning task run successfully. In scenario 2 and scenario 4, the overall planning task can be completed directly by using the neural sampler. And it can be seen through the path that it is feasible to achieve planning using the proposed method.

We conducted stability testing on the method through numerical simulations, employing the Monte Carlo stability testing approach. The uncertainty perturbation information for the initial state and the sampling strategy at each step is presented in the table II. The results, as shown in Table III, showed that under 2000 trials, there were 1954 successes in finding a feasible path. The reason for the failure of finding the path is that the initial perturbation of the starting point causes the starting point to fall inside the obstacle. In other words, our method can find the path feasible solution stably with the guarantee of initial feasibility.

TABLE II: Monte-Carlo Test Perturbations

Object	Distribution	3- σ range	Object	Distribution	3- σ range
x_{start}	Gaussian	0.5m	y_{start}	Gaussian	0.5m
u_1	Gaussian	1m	u_2	Gaussian	2°

TABLE III: Monte-Carlo Test Results

Test Count	Success Count	Failure Count	Success Rate
2000	1954	46	97.7%

C. Method Comparison and Discussion

In this section, we compare the proposed method with sampling-based path planning algorithms. The compared path planning algorithms are shown in Table IV. To ensure the fairness of the comparison, the same sub-algorithms in all algorithms are kept consistent. As shown in Fig. 7, we conducted comparison tests in four scenarios to reflect the generality of the proposed method. Each algorithm was run 50 times in each scenario and the average was used to make comparisons between the data.

TABLE IV: Related Comparison Algorithms.

Algorithm	Algorithm characteristics
RRT* [18]	The classic rapidly-exploring random tree algorithm.
PRM [17]	A centralized sampling-based algorithm.
Informed-RRT* [36]	The RRT* algorithm for non-uniform sampling in elliptical regions.
TAA-RRT* [22]	GVD-based non-uniform sampling and adaptive step-size strategies.
NS-RRT*	The algorithm proposed in this paper.

As shown in Fig. 7, the quality of the paths found by the NS-RRT* and TAA-RRT* algorithms is significantly higher than that of the RRT*, PRM and Informed-RRT* algorithms. From the distribution of sampling points, RRT*, PRM and Informed-RRT* have to sample almost the whole area, which is more inefficient. The proposed NS-RRT* algorithm, on the other hand, maximises the efficiency and basically rarely needs to require sampling to improve the path. Especially in narrow spaces and open spaces, the NS-RRT* algorithm can plan the path directly.

At the same time, we compare the proposed methods at the data level so as to demonstrate the superiority of the proposed methods. We mainly focus on the following three performance indicators when the planner converges, which reflect the different characteristics of the proposed method.

- Indicator 1: the number of sampling points (to verify the real-time performance of the proposed method).
- Indicator 2: the length of the planned path (to demonstrate the stability and optimality or near-optimality of the proposed method).
- Indicator 3: the Mean Absolute Percentage Error (MAPE) value of the planned path (to verify the accuracy of the proposed method).

The indicator 1 is the average values when the path solution converges. It can be seen from Fig. 9 that the NS-RRT* algorithm converges with a minimum number of sampling points compared to other algorithms. Taking the scenario 3 as an example, the number of sampling points of the proposed method is reduced by about 87.5% compared to the TAA-RRT* algorithm, about 95% compared to the Informed-RRT* algorithm, about 95.5% compared to the PRM algorithm and about 96.3% compared to the classic RRT* algorithm. As shown in Fig. 9, a bar chart comparing the number of sampling points reflects the efficiency of the proposed algorithm. When path solutions converge, we also record the planning time of the paths, as shown in Table V. The random sampling planner requires a larger number of sampling points to generate and improve paths. The neural sampling planner requires only fewer points to complete the planning. Thus only a shorter time is required to complete the planning. This reflects the real time performance of the proposed method.

TABLE V: Planning time for five algorithms in multiple scenarios.

Simulation Scenario	Method	Path Length	Planning Time
Scenario 1	RRT*	712.19m	65.38s
	PRM	783.87m	78.46s
	Informed-RRT*	717.59m	35.14s
	TAA-RRT*	652.18m	5.49s
	NS-RRT*	653.21m	0.878s
Scenario 2	RRT*	428.75m	18.63s
	PRM	450.76m	35.11s
	Informed-RRT*	425.27m	12.53s
	TAA-RRT*	413.81m	6.95s
	NS-RRT*	413.92m	0.058s
Scenario 3	RRT*	785.94m	66.67s
	PRM	1054.00m	68.87s
	Informed-RRT*	792.88m	22.51s
	TAA-RRT*	771.69m	4.12s
	NS-RRT*	770.62m	0.88s
Scenario 4	RRT*	528.07m	76.75s
	PRM	618.39m	78.29s
	Informed-RRT*	521.31m	32.51s
	TAA-RRT*	508.69m	13.75s
	NS-RRT*	508.35m	0.076s

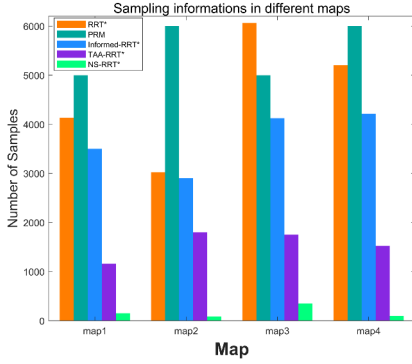


Fig. 9: Number of samples in different maps.

Under a certain number of sampling points, we counted the indicator 2 of the four algorithms. We plotted the box-line diagrams of the indicator 2 to verify the stability of the proposed method, as shown in Fig. 10. The box-line diagram shows the stability of the 50 sets of paths generated by each method. It can be seen from Fig. 10 that the box corresponding to the proposed method is very narrow, which shows the strong stability of the proposed method.

To verify the accuracy the proposed method, we calculated the indicator 3 of five algorithms in four maps, as shown in Table VI. The MAPE value is calculated by the following:

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}}{x_i} \right| \quad (14)$$

where x_i represents the length of the i -th path among the n paths, and $\hat{x}$ represents the average value of the n paths. The smaller the value of MAPE, the higher the accuracy of the path generated by the planner. Therefore, the proposed method can accurately plan paths under all four maps.

TABLE VI: MAPE value of four algorithms in different maps.

Algorithm(Map)	map1	map2	map3	map4
RRT*	5.20%	13.41%	3.44%	9.69%
PRM	3.11%	1.28%	3.24%	4.31%
Informed-RRT*	2.35%	11.39%	3.05%	3.46%
TAA-RRT*	4.65%	0.14%	4.51%	4.62%
NS-RRT*	0.26%	0.06%	0.19%	0.05%

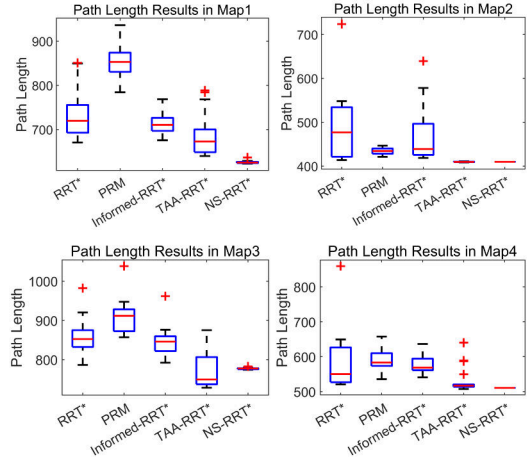


Fig. 10: Path length results in different maps.

In summary, the superiority of the proposed method can be seen through comparative tests.

D. Real-world Experiments

In this section, we conduct experiments in a real-world. The experimental platform is shown in the Fig. 11. The AGV is equipped with LS01D lidar, Astra camera, inertial measurement unit (IMU), odometer, etc.

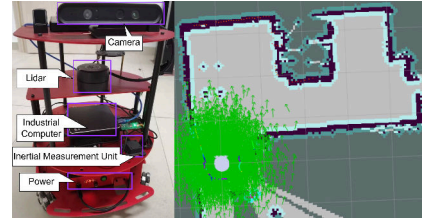


Fig. 11: Experimental platform.

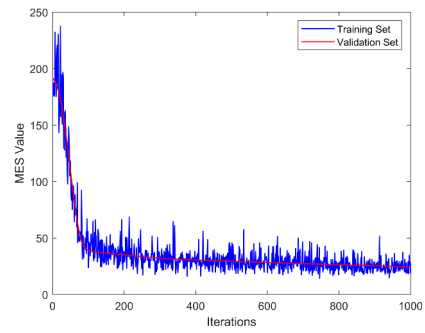


Fig. 12: The training results of experimental data.

We carry out the open-loop experimental validation in the scenario shown in Fig. 11. First, we need to utilize radar sensors to conduct mapping operations in Robot Operating System to gather environmental information. Then we collect the dataset by continuously performing a planning task. After each planning, the coordinate values of the AGVs are recorded by broadcasting their positions in the ROS. After collecting the data, we calculate the optimal sampling strategies and then use the network to learn the optimal sampling strategy. After collecting the data, OPBG is employed to rapidly expand the scale of the dataset. Subsequently, a sampling model is trained using a network, with the convergence of training depicted

in Fig. 12. Finally, the model is utilized for guided sampling. The planned path is feasible during the actual planning of travelling. The AGV moving along the reference path process in Rviz is shown in Fig. 13. In Fig. 14, we have documented the experimental results under various tasks. It can be observed that the planner is capable of producing corresponding sampling strategies at different positions. In summary, the feasibility of the proposed method is verified by real-world experiments.

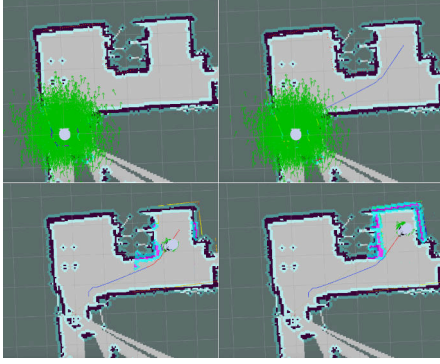


Fig. 13: Demonstration of AGV planning and tracking experiment in Rviz.

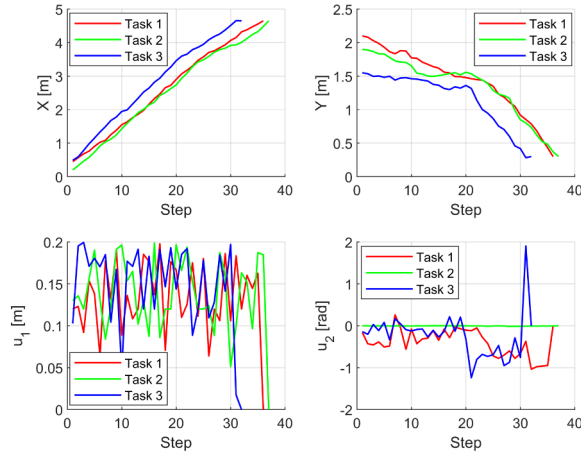


Fig. 14: Experimental data of sampling points and sampling strategies at each step in the path results.

V. CONCLUSION

In this study, a novel hierarchical sampling-based path planning algorithm with a three-layer structure is proposed. In this approach, the core is the use of DNN to guide the RRT* algorithm to sample intelligently. The first layer is the dataset preparation phase, where efficient RRT* algorithm is used to generate the path dataset. Meanwhile, an OPG strategy is proposed for the diversity of the datasets and the efficiency of the generation process. The second layer is the learning phase of the DNN network. The trained model will be used for the sampling process of RRT* in the third layer. The third layer is the sampling planning phase where the trained model intelligently guides the algorithm to sample. In the third layer, we implement a designed efficient guarantee mechanism to ensure fast path connectivity in the event of network sampling planning failure. Finally, the feasibility, efficiency, accuracy and stability of the proposed method are verified by simulations and experiments.

In future work, firstly, we intend to incorporate the kinematic information of robots to generate a smooth path that adheres to kinematic constraints. Secondly, we aim to extend the proposed methodology to more intricate three-dimensional scenarios, such as space robot [37], to enhance overall operational efficiency. Lastly, generalization will also be a focal point of our attention going forward, and we plan to explore the applicability of the proposed framework through techniques like transfer learning.

REFERENCES

- [1] B. Li, Y. Ouyang, X. Li, D. Cao, T. Zhang, and Y. Wang, "Mixed-integer and conditional trajectory planning for an autonomous mining truck in loading/dumping scenarios: A global optimization approach," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1512–1522, 2022.
- [2] T. Liu, R. Chai, S. Chai, and Y. Xia, "Chance-constrained trajectory optimization for automatic parking based on conservative approximation," *IEEE Transactions on Vehicular Technology*, DOI 10.1109/TVT.2023.3342421, pp. 1–11, 2023.
- [3] Y. Ji, Y. Tanaka, Y. Tamura, M. Kimura, A. Umemura, Y. Kaneshima, H. Murakami, A. Yamashita, and H. Asama, "Adaptive motion planning based on vehicle characteristics and regulations for off-road uavs," *IEEE Transactions on Industrial Informatics*, vol. 15, DOI 10.1109/TII.2018.2870662, no. 1, pp. 599–611, 2019.
- [4] J. Wang, M. Q.-H. Meng, and O. Khatib, "EB-RRT: Optimal motion planning for mobile robots," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 2063–2073, 2020.
- [5] G. Zhang, S. Chai, R. Chai, M. Garcia, and Y. Xia, "Fuzzy goal programming algorithm for multi-objective trajectory optimal parking of autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, 2023.
- [6] J. Votion and Y. Cao, "Diversity-based cooperative multivehicle path planning for risk management in costmap environments," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 8, pp. 6117–6127, 2018.
- [7] E. Gough, A. T. Conn, and J. Rossiter, "Planning for a tight squeeze: Navigation of morphing soft robots in congested environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4752–4757, 2021.
- [8] G. Du, Y. Zou, X. Zhang, Z. Li, and Q. Liu, "Hierarchical motion planning and tracking for autonomous vehicles using global heuristic based potential field and reinforcement learning based predictive control," *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [9] Z. Li, Y. Fei, and R. Shi, "UAV 3D Trajectory Planning and Optimization Using A* Algorithm Based on Free Space Method," *Unmanned Systems*, pp. 1–20, 2023.
- [10] N. Ganganath, C.-T. Cheng, and C. K. Tse, "A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains," *IEEE Transactions on Industrial Informatics*, vol. 11, DOI 10.1109/TII.2015.2413355, no. 3, pp. 601–611, Jun. 2015.
- [11] J.-J. Kim and J.-J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE transactions on industrial informatics*, vol. 11, no. 3, pp. 620–631, 2015.
- [12] L. Zhang, Y. Zhang, and Y. Li, "Mobile robot path planning based on improved localized particle swarm optimization," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6962–6972, 2020.
- [13] Y.-N. Ma, Y.-J. Gong, C.-F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: An ant colony algorithm incorporating alarm pheromone," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 141–154, 2018.
- [14] Y. Chen, D. Yang, and J. Yu, "Multi-uav task assignment with parameter and time-sensitive uncertainties using modified two-part wolf pack search algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 6, pp. 2853–2872, 2018.
- [15] H. Yang, J. Qi, Y. Miao, H. Sun, and J. Li, "A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, pp. 8557–8566, 2018.
- [16] C.-b. Moon and W. Chung, "Kinodynamic planner dual-tree rrt (dt-rrt) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Transactions on industrial electronics*, vol. 62, no. 2, pp. 1080–1090, 2014.
- [17] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

- [18] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [19] J. Xu, K. Song, D. Zhang, H. Dong, Y. Yan, and Q. Meng, "Informed anytime fast marching tree for asymptotically optimal motion planning," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5068–5077, 2020.
- [20] Y. Chen, Z. Zhang, Z. Wu, Z. Miao, H. Zhang, and Y. Wang, "Set: Sampling-enhanced exploration tree for mobile robot in restricted environments," *IEEE Transactions on Industrial Informatics*, vol. 19, DOI 10.1109/TII.2023.3240935, no. 10, pp. 10 467–10 477, 2023.
- [21] E. Sumer, M. R. Imamoglu, and H. Temeltas, "Indoor 2d autonomous exploration with an omnidirectional robot: A strategy based on rapidly-exploring random trees," *Unmanned Systems*, 2024.
- [22] R. Zhang, R. Chai, S. Chai, Y. Xia, and A. Tsourdos, "Design and practical implementation of a high efficiency two-layer trajectory planning method for agv," *IEEE Transactions on Industrial Electronics*, vol. 71, DOI 10.1109/TIE.2023.3250847, no. 2, pp. 1811–1822, 2024.
- [23] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized voronoi diagram-based efficient heuristic path planning method for rrt in mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 4926–4937, 2021.
- [24] Y. Zhang, H. Wang, M. Yin, J. Wang, and C. Hua, "Bi-AM-RRT*: A Fast and Efficient Sampling-Based Motion Planning Algorithm in Dynamic Environments," *IEEE Transactions on Intelligent Vehicles*, DOI 10.1109/TIV.2023.3307283, pp. 1–12, 2023.
- [25] Z. Sun, B. Lei, P. Xie, F. Liu, J. Gao, Y. Zhang, and J. Wang, "Multi-Risk-RRT: An Efficient Motion Planning Algorithm for Robotic Autonomous Luggage Trolley Collection at Airports," *IEEE Transactions on Intelligent Vehicles*, DOI 10.1109/TIV.2023.3349171, pp. 1–13, 2024.
- [26] J. Dai, Y. Zhang, and H. Deng, "Novel potential guided bidirectional rrt* with direct connection strategy for path planning of redundant robot manipulators in joint space," *IEEE Transactions on Industrial Electronics*, vol. 71, DOI 10.1109/TIE.2023.3269462, no. 3, pp. 2737–2747, 2024.
- [27] H. Ji, H. Xie, C. Wang, and H. Yang, "E-RRT*: Path Planning for Hyper-Redundant Manipulators," *IEEE Robotics and Automation Letters*, vol. 8, DOI 10.1109/LRA.2023.3325716, no. 12, pp. 8128–8135, 2023.
- [28] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, and S. Chai, "Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver," *IEEE Transactions on Automation Science and Engineering*, 2022.
- [29] B. Ma, Z. Xie, X. Yang, Y. Liu, Z. Wang, and Z. Jiang, "Neurodynamics-based configuration transformation with engineering application to robot manipulators using two intelligent approaches," *Engineering Applications of Artificial Intelligence*, vol. 120, p. 105861, 2023.
- [30] H. Ma, C. Li, J. Liu, J. Wang, and M. Q.-H. Meng, "Enhance connectivity of promising regions for sampling-based path planning," *IEEE Transactions on Automation Science and Engineering*, 2022.
- [31] B. Ma, Z. Xie, B. Zhan, Z. Jiang, Y. Liu, and H. Liu, "Actual shape-based obstacle avoidance synthesized by velocity–acceleration minimization for redundant manipulators: An optimization perspective," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [32] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. P. Chen, "Six-dof spacecraft optimal trajectory planning and real-time attitude control: a deep neural network-based approach," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 11, pp. 5005–5013, 2019.
- [33] F. Meng, L. Chen, H. Ma, J. Wang, and M. Q.-H. Meng, "NR-RRT: Neural Risk-Aware Near-Optimal Path Planning in Uncertain Nonconvex Environments," *IEEE Transactions on Automation Science and Engineering*, vol. 21, DOI 10.1109/TASE.2022.3215562, no. 1, pp. 135–146, 2024.
- [34] J. Wang, W. Chi, C. Li, C. Wang, and M. Q.-H. Meng, "Neural RRT*: Learning-Based Optimal Path Planning," *IEEE Transactions on Automation Science and Engineering*, vol. 17, DOI 10.1109/TASE.2020.2976560, no. 4, pp. 1748–1758, 2020.
- [35] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [36] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Informed sampling for asymptotically optimal path planning," *IEEE Transactions on Robotics*, vol. 34, DOI 10.1109/TRO.2018.2830331, no. 4, pp. 966–984, 2018.
- [37] B. Ma, Z. Jiang, Y. Liu, and Z. Xie, "Advances in space robots for on-orbit servicing: A comprehensive review," *Advanced Intelligent Systems*, p. 2200397, 2023.



energy systems, and transportation-energy nexus.



active disturbance rejection control and flight control.



Runda Zhang received the B.S. degree in traffic equipment and control engineering from Taiyuan University of Technology, Taiyuan, China in 2021. He is currently studying for a Ph.D. in artificial intelligence at the School of Automation, Beijing Institute of Technology, Beijing, China. His research interests are path planning and trajectory optimization for robots.

Runqi Chai (Senior Member, IEEE) received the B.S. degree in information and computing science from the North China University of Technology, Beijing, China, in 2015 and the Ph.D. degree in Aerospace Engineering from Cranfield University, Cranfield, U.K, in August 2018. He is currently a professor at Beijing Institute of Technology. His research interests include unmanned vehicle trajectory optimization, guidance and control.

Kaiyuan Chen (Member, IEEE) received the PhD degree from Swansea University in 2022. She currently serves as a research fellow at Vanke School of Public Health, Tsinghua University. Her research interests include public health, global health governance, smart healthcare, the application of unmanned system in public health and other relevant field.

Jinning Zhang received the B.S. in vehicle engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2018 and received the MSc and PhD degree in Aerospace Propulsion with Cranfield University, U.K in 2019 and 2022. She is currently a Lecturer (Assistant Professor) in Aerospace and Computational Engineering with University of Leicester, UK. Her research interests include energy management strategies and integrated control system design for aerospace propulsion, AI for transport and

Senchun Chai (Senior Member, IEEE) received the Ph.D. degree in Networked Control System from University of South Wales, Pontypridd, U.K., in 2007. He is currently a professor of School of Automation with Beijing Institute of Technology. His current research interests focus on flight control system, networked control systems, embedded systems and multi-agent control systems.

Yuanqing Xia (Fellow, IEEE) received the B.S. degree in fundamental mathematics from the Department of Mathematics, Chuzhou University, Chuzhou, China, in 1991, the M.S. degree in fundamental mathematics from Anhui University, Wuhu, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. His current research interests are in the fields of networked control systems, robust control and signal processing,

Antonios Tsourdos (Member, IEEE) obtained a PhD on Nonlinear Robust Autopilot Design and Analysis from Cranfield University in 1999. He joined the Cranfield University in 1999 and currently serve as the Director of Research - Aerospace, Transport and Manufacturing in 2015. He conducts basic and applied research in the fields of guidance, control and navigation for single and multiple unmanned autonomous vehicles as well as research on cyber-physical systems.



DSPACE

<https://dspace.org/>

Efficient and near-optimal global path planning for AGVs: a DNN-based double closed-loop approach with guarantee mechanism

Zhang, Runda; Chai, Runqi; Chen, Kaiyuan; Zhang, Jinning; Chai, Senchun ...

2024

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/tie.2024.3409883>

Zhang R, Chai R, Chen K, et al., (2024) Efficient and near-optimal global path planning for AGVs: a DNN-based double closed-loop approach with guarantee mechanism. IEEE Transactions on Industrial Electronics, Volume 72, Issue 1, January 2025, pp. 681-692