Semi-Supervised Voronoi-TSVM Path Planning for Automated Guided Vehicles under Multi-Modal Perception

Jing Sun
Tianjin Urban Construction Management & Vocation Technology College, Tianjin 300134, China
E-mail: Jing_Sun521@126.com

Keywords: transductive support vector machine, path planning, Voronoi diagram, automated guided vehicle

Received: June 13, 2025

This work proposes a fusion optimization method combining Voronoi diagram and Transductive Support Vector Machine (TSVM) to address the path planning problem of Automated Guided Vehicle (AGV) in complex environments. First, this method uses a Voronoi diagram to construct the spatial skeleton of the environment and generate an initial path. Second, it introduces TSVM to perform semi-supervised classification and optimized screening on path segments, to screen out safer and smoother segments. Finally, this method ensures the safety of the path while improving its smoothness and operational efficiency. In evaluation indicators, path smoothness is defined as the standard deviation of path angle changes (unit: radian), and a smaller value indicates a smoother path; collision rate refers to the proportion of path points where the minimum distance from obstacles is lower than the safety threshold. The experimental results show that the optimized model has a path length of 42.987 meters (m) under LiDAR data; it is remarkably shorter than that of Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) (46.879m) and Deep Reinforcement Learning - Model Predictive Control (DRL-MPC) (45.212m). This indicates that the optimized model can offer more efficient path choices. Concerning computation time, this model performs excellently, with 1.345 seconds (s) under LiDAR data, much lower than CNN-LSTM's 2.346s and DRL-MPC's 1.879s, demonstrating its high efficiency in path planning. Regarding path smoothness, the optimized model achieves 0.208, superior to CNN-LSTM's 0.345 and DRL-MPC's 0.298, thus reducing vibrations and path deviations. Moreover, this model has a collision rate of 0.012 under LiDAR data, remarkably lower than CNN-LSTM's 0.062 and DRL-MPC's 0.045. Consequently, this work develops a novel path planning strategy integrating classification learning with spatial mapping. It also validates the strategy's universality and robustness in multi-source perception scenarios, providing theoretically sound and engineering-feasible optimization solutions for intelligent AGV systems.

Povzetek: Članek obravnava učinkovitost klasičnih AGV-metod pri načrtovanju poti v kompleksnih okoljih. Predlaga hibrid Voronoi-TSVM: Voronoijev diagram ustvari varnostni prostorski skelet, TSVM pa polnadzorovano razvrsti in filtrira odseke poti glede na dolžino, zavoje in gladkost. Model skrajša pot, izboljša gladkost in zmanjša kolizije.

1 Introduction

In modern manufacturing and intelligent logistics systems, Automated Guided Vehicle (AGV) has been widely applied in warehousing operations, production line material handling, and hospital automated delivery scenarios [1-3]. The AGV's core functionality relies on its path planning capability. Traditional methods, such as grid-based mapping or A* algorithm, demonstrate satisfactory performance in structured environments. However, they often generate path redundancy and obstacle avoidance delays when encountering complex obstacle configurations in dynamic, unstructured settings [4]. The Voronoi diagram, with its superior spatial partitioning capability, has been extensively utilized in robotic path planning. The basic idea is to divide the space into several regions, where each region consists of all points closest to a specific reference point. This division method can naturally create obstacle-avoiding boundaries and provide a relatively safe navigation framework for AGV [5-7]. Nevertheless, paths generated by Voronoi diagrams frequently suffer from excessive turns and suboptimal length, limiting further improvements in path quality [8]. Recent advancements have introduced machine learning (ML) techniques into path optimization. Transductive Support Vector Machine (TSVM), as a semi-supervised classification model, exhibits excellent boundary discrimination capability that proves valuable for path region classification problems. The integration of TSVM with Voronoi diagrams enables safety-zone-based path mapping; this integration also optimizes waypoint selection and connection patterns, thus enhancing path smoothness and overall efficiency.

Building upon these foundations, this work proposes a novel AGV path planning method combining Voronoi diagrams with TSVM. It aims to achieve efficient, safe,

and continuous path generation in complex environments. The methodology initially constructs obstacle-avoidance regions through Voronoi diagrams. Then, it employs TSVM for classification and filtering of candidate paths to optimize waypoint distribution while minimizing unnecessary turns and path fluctuations.

This work addresses the following questions:

(i) How can the partitioning capability of Voronoi diagrams be effectively integrated with the discriminative advantages of TSVM for path generation and optimization of AGV?

(ii) Does the proposed model outperform existing comparative methods under typical perception data?

(iii) Does this approach exhibit stable advantages in key indicators such as path length, smoothness, and collision rate?

Through discussions of these questions, this work provides a modeling paradigm for AGV path planning that integrates the advantages of spatial geometry and intelligent classification. Thus, it can offer new ideas and practical support for intelligent mobile agent navigation in complex environments.

2 Related work

With the continuous development of AGV technology, AGV path planning has become a hot research topic. Researchers have proposed various methods from different perspectives to achieve efficient and safe path planning for the AGV.

Ming et al. proposed a shortest path algorithm that traversed all the nodes in a graph to find the shortest path from the starting point to the destination. The Dijkstra algorithm could effectively handle path planning problems in static environments. However, it suffered from high computational complexity and poor real-time performance in complex environments [9]. Pan applied the Voronoi diagram to AGV path planning and proposed a path generation method based on the Voronoi diagram. By creating a Voronoi diagram of the environment, this method could effectively avoid path overlap with obstacles, ensuring the safe operation of the AGV. Nevertheless, the paths generated by the Voronoi diagram often suffer from path redundancy and excessive turns, requiring further optimization [10]. Gul introduced an improved Voronoi diagram path planning method that considered the impact of dynamic obstacles during the construction of the Voronoi diagram. Studies indicated that the Voronoi diagram effectively adjusted the path in dynamic environments, enhancing the AGV's flexibility and adaptability. However, this method still faced challenges in high computational complexity, especially when dealing with complex obstacle layouts [11]. Chek presented a method based on Support Vector Machine (SVM) for AGV path optimization. This approach could learn from historical path data to optimize the AGV's path selection. While SVM effectively reduced path redundancy, its computational cost during training could be quite high, impacting its application, especially in large-scale environments [12]. The work further sorts out the comparison of typical AGV path planning methods, as exhibited in Table 1:

Table 1: Comparison of typical AGV path planning methods

Method name	Input data type	Computational efficiency	Path quality	Main limitations
Dijkstra algorithm	Grid maps, static maps	Medium	Feasible, but the path is lengthy.	Path smoothness is not considered; The amount of computation grows rapidly with the number of nodes.
A* algorithm	Grid-based maps, heuristics (e.g. Euclidean distances)	Higher	The path is usually short, with frequent turns.	It has poor adaptability to the dynamic environment, and the path is not smooth enough.
Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM)	Multimodal data, such as depth images and LiDAR	>2.3 s; Lower (inference time)	The path smoothness is 0.3; it adapts to dynamic scenarios, and the path is relatively continuous.	Model training relies on large amounts of data; it has limited generalization ability.
Deep Reinforcement Learning - Model Predictive Control (DRL-MPC)	State-space imagery or LiDAR	1.88 s; Medium to lower	The average path length is approximately 45.2 meters (m), superior to conventional, balancing energy consumption and dynamic changes.	Slow convergence; difficulty interpreting the decision-making process; Sensitive to hyperparameters

The above analysis suggests that traditional methods often separate environmental modeling from path optimization. They either rely on complex numerical optimization or train end-to-end deep learning (DL) models with a large amount of labeled data. Although existing methods show specific advantages in different scenarios, they still have obvious limitations in generalization and robustness. For example, DRL-MPC

performs excellently in dynamic environments, but its decision-making process is opaque and it is sensitive to hyperparameters, leading to a decline in generalization ability in unknown environments; CNN-LSTM relies on a large amount of labeled data for training, and its performance fluctuates markedly when data is scarce or perception quality is uneven. In contrast, the proposed Voronoi-TSVM fusion model shows certain advantages in solving the AGV path planning problem. The innovation of this work lies in the in-depth integration of the Voronoi diagram's spatial structuring ability and the TSVM's semi-supervised boundary discrimination ability, forming a "generation-screening" collaborative framework. This framework aims to overcome the inherent problems of redundant paths and excessive turns in traditional geometric methods. Meanwhile, it completes high-quality path optimization based on only a small number of labeled samples. Moreover, it reduces the dependence on data scale and quality, providing a new solution for the AGV's efficient navigation in dynamic and data-scarce environments. This integration strategy can improve the reliability, interpretability, and environmental adaptability of path planning, and make up for the gap between existing methods at the structural and learning levels.

Against this backdrop, this work introduces TSVM; it integrates candidate path skeletons generated by Voronoi diagrams to achieve semi-supervised optimization of candidate path regions using minimal labeled data. Thus, it simultaneously enhances path feasibility, continuity, and execution efficiency. Unlike previous strategies that use ML as a "path re-scoring" tool, the proposed method emphasizes a linkage mechanism between geometric mapping and classification discrimination. This represents an integrated path planning framework that fuses structural generation with semantic optimization.

3 Research method

3.1 The use of voronoi diagram in AGV path planning

The Voronoi diagram, a mathematical tool based on spatial geometric partitioning, is extensively used in fields such as robot navigation, target allocation, map segmentation, and path planning. Its theoretical foundation stems from the "nearest neighbor division" concept in computational geometry. In the context of AGV path planning, the core advantage of the Voronoi diagram lies in its ability to explicitly partition the environmental space geometrically. Moreover, it can effectively distinguish between obstacle areas and free passage zones, providing an interpretable, safe, and highly robust geometric foundation for path generation [13, 14].

Specifically, the Voronoi diagram divides the entire environment into a set of non-overlapping subregions through spatial distance comparisons among several generator points. For any point within each sub-region, the distance to the corresponding generator point is the smallest. These generator points typically include obstacle boundary points, start points, end points, and key landmarks. When applied to path planning, this partitioning method restricts feasible paths to the positions farthest from obstacles (i.e., Voronoi boundaries), maximizing path safety and obstacle avoidance capabilities. The mathematical rationale for Voronoi path planning is that the greater the distance between an AGV and obstacles, the lower the collision probability [15,16]. Voronoi boundaries are formed by the perpendicular bisectors of adjacent obstacle points and naturally lie on the "maximum clearance channels" between obstacles. Consequently, paths constructed based on Voronoi diagrams can effectively provide collision-free navigation solutions with high margin redundancy. This property is particularly critical for scenarios with complex structures, dense obstacles, or irregular shapes. The typical steps for constructing a Voronoi diagram include:

(1) Choosing generator points: Based on the obstacles in the environment or specific needs, appropriate generator points are selected.

(2) Calculating Voronoi boundaries: The boundaries of the Voronoi diagram are constructed by calculating the perpendicular bisectors between each pair of generator points.

(3) Generating Voronoi cells: Grounded in the calculated boundaries, the space is divided into multiple Voronoi cells.

Through these steps, the AGV can use the Voronoi diagram to generate collision-free paths. The generated path usually lies along the boundaries of the Voronoi cells, ensuring that the AGV does not enter obstacle regions during its movement [17-19].

3.2 The use of TSVM in path optimization

In the AGV path planning system, TSVM evaluates and filters the preliminary paths generated by the Voronoi diagram. Its task is not to directly generate paths or perform numerical optimization. Conversely, it identifies high-quality path segments among multiple candidate schemes, thereby assisting in forming the final optimized path [20, 21]. As a classification model, TSVM constructs two hyperplanes, close to positive and negative class samples, thus demonstrating good performance in handling imbalanced samples and heterogeneous data [22, 23].

The path optimization process is modeled as a discriminative problem in the proposed method. First, the system constructs a set of initial path candidates based on the Voronoi diagram, which typically covers multiple possible safe passage areas. Each path is divided into several path segments. Then, three types of key features

are extracted for each segment: path length, number of turns, and smoothness (approximated by angular changes). Each path segment corresponds to a three-dimensional feature vector to characterize its structural complexity and passage quality [24, 25]. In the feature vector, "path length" refers to the total Euclidean distance of the segmented trajectory (unit: m); "number of turns" is counted by the number of points where the change in direction angle exceeds the threshold; "smoothness" is calculated as the average value of the absolute changes in angles formed by all consecutive three points within the segment. The calculation method of the angle is as follows. Taking three consecutive points P_1, P_2, and P_3, the angle is calculated between vectors P_1P_2 and P_2P_3 (unit: radian); the absolute value of this angle reflects the intensity of direction change. The smoothness of the entire segment is the average value of the absolute angles of all consecutive three points, and a smaller value indicates a smoother path. Turning point detection relies on the discrete calculation of direction angles and is realized by combining a sliding window and threshold judgment.

The system builds a path quality sample set to train the TSVM model. Well-performing segments (e.g., shorter length, fewer turns, smoother movement) are positive class samples. In contrast, poorly performing segments (e.g., close to obstacle edges, sharp consecutive turns, or obvious path oscillations) are defined as negative class samples [26]. These samples can be derived from historical paths in previous tasks, expert experience annotations, or simulation sampling. Once trained, the model can discriminate the quality of newly generated path segments.

In practical operation, the AGV system generates multiple feasible paths from the Voronoi diagram, with each path segmented and feature-extracted before being input into TSVM for classification [27, 28]. The labels output by the model determine whether each path segment is retained. A structurally superior and more reliable path is ultimately formed by preserving high-quality segments, removing low-quality ones, and reconfiguring the remaining segments. Notably, TSVM does not directly perform path optimization but provides a path quality discrimination mechanism, acting as a "path segment selector" in the system's path planning process. Its role is to evaluate the rationality and executability of path segments based on structural features. Meanwhile, it indirectly enhances overall path performance by reducing redundant length, minimizing unnecessary turns, and improving smoothness [29].

Additionally, the common issue of sample imbalance in path optimization is addressed (where high-quality path segments are far fewer than low-quality ones). TSVM's dual hyperplane construction mechanism enables more stable learning of path segment classification boundaries. This can effectively avoid misjudgments caused by skewed sample distributions [30, 31]. This mechanism is particularly beneficial for the AGV to quickly adapt to new scenarios in complex or dynamic environments.

Regarding model execution logic, TSVM does not directly modify path point coordinates or participate in the internal search process of the path generation algorithm. Instead, it performs quality evaluation and screening on a set of candidate paths or segments generated externally from the Voronoi path structure. The final path reconstruction and execution are completed through a post-processing process driven by TSVM classification results.

In summary, TSVM primarily undertakes path segment-level discrimination tasks in the proposed AGV path planning system. Through learning and classifying path structural features, the model assists in filtering and reorganizing candidate path segments, thereby optimizing overall path structure and enhancing execution performance. This mechanism effectively bridges the global spatial partitioning capability of Voronoi diagrams with the practical need for local optimization at the path execution level.

3.3 Construction of AGV path planning model based on voronoi diagram and TSVM

In complex dynamic environments, AGV path planning faces multiple constraints among obstacle avoidance, path redundancy, and real-time performance. Traditional methods typically separate path construction and optimization, failing to dynamically adapt to environmental changes. This work proposes a path planning model fusing Voronoi diagrams and TSVM to address these issues. The model is designed to combine environmental space partitioning with path region discrimination, achieving fine-grained path optimization while ensuring safe and reachable routes. The method consists of two main stages. First, Voronoi diagrams are used to partition obstacle environments and construct global path graphs; Second, the A* algorithm is applied to search for the initial path on these graphs. Then, the path candidate points are semi-supervised and classified through TSVM to eliminate the inappropriate path segments and improve path quality. The final path is smoothed and converted into control commands for input to the AGV control module.

Considering that TSVM relies on minimal labeled samples for optimization training, this work designs a weak label generation mechanism. Specifically, path start/end points and their adjacent areas are designated as positive samples. The sample data required for TSVM training is mainly obtained by running traditional planning algorithms in a simulation environment to conduct a large number of path samplings. Positive samples are selected from path points that are far from obstacles and have gentle curvature changes; negative samples are taken from areas close to obstacles or with excessively sharp turns. For some samples that are

difficult to automatically identify, experts provide auxiliary labeling based on the rules of safe distance and movement fluency.

Regions close to obstacle boundaries or with path deviations are defined as negative samples. Most remaining Voronoi diagram path candidate points are input as unlabeled samples. This strategy balances training efficiency and discriminative capabilities in scenarios with scarce labels, meeting the fast adaptation requirements of AGV online operations. The entire system comprises five modules—environmental perception, spatial partitioning, path generation, path optimization, and smooth control. The specific workflow is outlined in the following pseudocode:

```
# Algorithm: AGV Path Planning with Voronoi Graph and TSVM
Input:
    OccupancyMap ← sensor fusion (LiDAR, IMU, depth camera)
    Obstacles ← {O1, O2, ..., On}
    StartPoint, GoalPoint
    TSVM_Params ← {C=1.0, gamma=0.5, tolerance=0.001, kernel='RBF'}
Output:
    OptimizedPath ← safe, smooth and executable path sequence
1. Construct Voronoi Diagram from Obstacles
    VoronoiGraph = generate_voronoi(OccupancyMap, resolution=0.1, inflation=0.5)
2. Perform A* Search on Voronoi Graph
    InitialPath = A_star_search(VoronoiGraph, StartPoint, GoalPoint, max_length=50)
3. Generate training samples for TSVM
    PositiveSamples = select_points_near(StartPoint, GoalPoint)
    NegativeSamples = select_points_near_obstacles(OccupancyMap)
    UnlabeledSamples = all_remaining_points(InitialPath)
4. Train TSVM using semi-supervised learning
    TSVM_Model = train_tsvm(PositiveSamples, NegativeSamples, UnlabeledSamples, TSVM_Params)
5. Filter path using TSVM prediction
    FilteredPath = [p for p in InitialPath if TSVM_Model.predict(p) == 'valid']
6. Smooth the path and convert to control commands
    FinalPath = smooth_path(FilteredPath)
    send_to_controller(FinalPath)
```

The computational complexity of each step in the above pseudocode varies. The time complexity of Voronoi diagram construction is usually O(n log n), where n is the number of obstacles. The complexity of performing A* search on the Voronoi diagram depends on the size of the diagram, and in the worst case, it is O(E + V log V); among them, V and E represent the number of vertices and edges, respectively. The time complexity of the TSVM classification stage is O(n), which has a linear relationship with the number of path points to be classified. The structure of the VoronoiGraph consists of generating points and edges. The path filtering logic is mainly implemented through TSVM prediction. The classifier outputs a "valid" or "invalid" label for each path point, and only retains consecutive valid points to form the final path.

From the above analysis, the specific implementation steps of the fused path are as follows. 1) The Voronoi diagram is responsible for spatial division of the environment and generating the initial path graph; 2) On this basis, TSVM performs semi-supervised classification on path points to screen out more reasonable path segments; 3) The two are integrated through feature extraction of path points and classification labels. Among them, the information provided by the Voronoi diagram, such as the coordinates of path points and their distance from obstacles, is converted into structured feature vectors processable by TSVM. These vectors include path length, number of turns, smoothness, and other relevant parameters. The training samples of TSVM are composed of positive samples, negative samples, and a large number of unlabeled Voronoi path points. The judgment of path quality is realized through semi-supervised learning. This integration mechanism aims to keep the path safe in geometric structure and make it more continuous and feasible at the local execution level.

The method offers advantages in two aspects. On one hand, the Voronoi diagram rapidly constructs safe regions to ensure basic path feasibility. On the other hand, TSVM achieves discriminative learning at the path point level to optimize path continuity and viability. Additionally, TSVM training employs the Sequential Minimal Optimization (SMO) algorithm with minimal labeled data. The average training time is 3.7 seconds (s), and the memory usage is approximately 260 megabytes (MB). It is suitable for deployment on resource-constrained edge computing devices. It should be noted that the current experimental results show that the model exhibits characteristics potentially suitable for edge device deployment in computational efficiency and resource consumption. However, this conclusion is mainly drawn based on simulation tests on standard hardware platforms and has not yet been fully verified in actual embedded systems. Therefore, before applying it to real edge computing scenarios, further hardware-in-the-loop testing and targeted optimization still need to be conducted.

Overall, the proposed Voronoi-TSVM joint path planning method fully integrates the advantages of spatial topology construction and ML optimization. This method demonstrates good real-time performance, discriminative capability, and execution stability in complex environments. This provides a new learning-enabled path generation paradigm for intelligent AGV navigation.

4 Experimental design

This work constructs a complete simulation experiment platform to verify the effectiveness of the proposed method. This platform integrates the Gazebo environment for physical simulation under the ROS Noetic framework and uses RViz for visualization. The environment map is constructed in real time through Simultaneous Localization and Mapping (SLAM) technology and converted into a two-dimensional Occupancy Grid Map as the input for path planning. The pseudocode form of the standard execution steps of the experiment's core algorithm process—the Voronoi-TSVM-based AGV path planning model—has been presented in the previous section.

The dataset used is from the Robot Operating System (ROS), which encompasses several public datasets related to AGV and path planning, containing experimental data on AGV navigation, path planning, and obstacle avoidance in various environments. The experiment utilizes multiple indoor environment scenarios from the Technical University of Munich (TUM) dataset, which is widely adopted in ROS. Additionally, it incorporates outdoor structured road data from the Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) dataset. The test environment includes 12 different environmental layouts, among which 8 are static environments and 4 contain dynamic interference objects; in each environment, 20 paths with different start and end points are planned, resulting in a total of 240 test paths. The datasets available on the ROS platform typically include experimental data for tasks such as robot path planning, obstacle detection, and path optimization. The datasets can be downloaded from the official ROS website (https://github.com/ros-perception/). In the experiment, the dataset is divided into three dimensions based on the sensor data type: LiDAR, depth image, and Inertial Measurement Unit (IMU) data.

The experiments are based on public datasets provided by the ROS platform. However, the dataset covers multiple typical AGV application scenarios, including indoor office areas, warehouse logistics corridors, and semi-structured obstacle distribution areas, with good environmental diversity and task representativeness. In the experimental design, layouts containing static obstacles (e.g., fixed shelves and walls) and dynamic interferences (e.g., moving obstacle simulators) simulate complex scenarios in real urban navigation or industrial warehouse environments. As the ROS dataset encompasses multi-source sensor data such as LiDAR, depth images, and IMU, the data are divided and modeled by sensor type in the experiments. This ensures comparable performance of the model under different perceptual inputs.

While this work focuses on presenting path planning results based on LiDAR data, the constructed model architecture remains general to input data types. Meanwhile, training and evaluation are conducted for depth image and IMU data dimensions. Due to space limitations, this work does not separately expand on all experimental details under different visual inputs. However, the model structure and feature extraction methods exhibit adaptability to diverse sensors and strong transferable generalization potential. Regarding the use of simulation experiments, the selected ROS Gazebo environment has high physical fidelity in path modeling, sensor modeling, and dynamic obstacle simulation. This can accurately reflect control behaviors and environmental interferences in real-world path planning systems. As this work focuses on proposing the model structure and discriminative optimization strategy, testing in real physical deployments is an important direction for future research expansion. It also fully recognizes the importance of real-world environment verification for model practicality and stability.

The comparison models selected for the experiment are CNN-LSTM and DRL-MPC, representing the application of DL and reinforcement learning (RL) in path planning, respectively. CNN-LSTM strongly supports spatial feature extraction and time-series modeling, making it suitable for dynamic path planning in complex environments. Moreover, DRL-MPC combines RL and MPC, offering a novel solution for AGV path planning and real-time optimization. It is particularly suitable for dynamic environments with high uncertainty. Both models are highly innovative and provide effective comparison and analysis for the path planning method based on the Voronoi diagram and TSVM. Previous studies indicated that heuristic algorithms, such as A* and D*, exhibited strong path optimization capabilities in static environments. Moreover, some hybrid methods performed well in specific scenarios. However, this work does not include them in comparative experiments for two reasons. On one hand, this work focuses on intelligent path optimization capabilities under high-dimensional perceptual inputs and complex scenarios. CNN-LSTM and DRL-MPC are mainstream DL models applied to AGV path planning in recent years. They represent two typical paradigms based on sequential modeling and RL + predictive control, respectively, with strong representativeness and a broad application foundation. On the other hand, heuristic algorithms like A* and D* can converge quickly under the premise of known structure diagrams and static obstacles. However, in highly dynamic scenarios or those with multi-source perceptual inputs, their performance is easily limited by map construction methods; they also lack end-to-end data-driven capabilities. Traditional SLAM methods integrated with path optimization are mostly system-level encapsulations, making it difficult to compare path generation performance as single models. Therefore, selecting CNN-LSTM and DRL-MPC as benchmarks covers the characteristics of mainstream neural methods; meanwhile, it forms a contrast with the

proposed “Voronoi+TSVM” structural model at the application level. This highlights the proposed model's comprehensive advantages in path quality and real-time performance.

The experimental setup is as follows:

(1) Processor: Intel Xeon W-2295 (18 cores, 3.00 GHz)
(2) Memory: 64 GB DDR4 RAM
(3) Storage: 1 TB SSD (main storage), 4 TB HDD (data storage)
(4) Graphics Card: NVIDIA Quadro RTX 4000 8GB
(5) Development Platform: ROS Noetic
(6) ROS Version: Noetic Ninjemys
(7) Relevant Packages: move_base, gmapping, amcl, nav_core
(8) Path Planning Library: Open Motion Planning Library (OMPL)

The experiment also sets parameters for the optimized model, with the following configuration. Grid resolution is set to 0.1m, obstacle radius is 2.0 m, boundary expansion factor is 0.5m, and penalty parameter is 1.0. The kernel function is Radial Basis Function (RBF), with a tolerance of 0.001; The path planning algorithm is A*, and the maximum path length is 50m. During training, a radial basis function kernel is employed with a kernel parameter γ set to 0.5, enhancing the model's nonlinear discriminative capability in path boundary regions. The regularization constant C is 1.0 to balance the relationship between classification margin and error tolerance. The tolerance is set at 0.001 to ensure convergence accuracy during model optimization, with a maximum of 1,000 iterations. The SMO method is an optimization solver, suitable for semi-supervised classification problems of moderate scale. Approximately 30% of the path region samples in the training data are labeled, with the rest being unlabeled, leveraging TSVM's generalization ability under limited labeling conditions. In the experiment, the same TSVM model is used for path optimization in all scenarios. This model is trained on a mixed dataset containing multiple environment types and is not trained separately for individual scenarios. This setup aims to verify the model's generalization ability in unseen environments; it is more in line with the requirement in practical applications that AGV needs to adapt to diverse scenarios. Moreover, to verify the advantages of TSVM when labeled data is limited, comparative experiments are conducted under different labeling ratios (10%, 30%, 50%). The results show that when the proportion of labeled samples decreases from 50% to 10%, the path classification accuracy of TSVM decreases by 5.2%. Meanwhile, the accuracy of the CNN-LSTM method declines by 13.7% under the same conditions. This confirms that TSVM has better stability in semi-supervised scenarios and can more effectively use unlabeled data to maintain model performance. Under standard hardware configurations, each TSVM training takes approximately 3.7 s with a standard deviation (SD) of 0.6 s. The model typically converges within 200 to 350 iterations, meeting error threshold conditions. In terms of resource consumption, a single training phase occupies approximately 260 MB of memory, demonstrating feasibility for deployment on edge computing devices. To enhance model robustness, all input samples undergo standard normalization before training. Boundary sample resampling is performed on class distributions to mitigate overfitting risks caused by initial sample imbalance.

Experiments are conducted at two levels to evaluate the proposed AGV path planning model's performance. One is the performance evaluation of path planning and execution, and the other is assessing path accuracy and control performance in simulation scenarios. Four representative indicators are selected for each dimension, constructing a complete evaluation system by integrating actual path performance and control responses.

Regarding path planning performance evaluation, path length is first considered. Path length refers to the total driving distance of the planned path from the starting point to the end point for the AGV, measuring whether the path is concise and efficient. Second is the calculation time. The total time required for the algorithm to generate a complete path after receiving map information reflects the model's operational efficiency and real-time performance. Additionally, path smoothness is used to measure the intensity of turns in the path; higher smoothness indicates a more continuous and fluid path, which is conducive to the stable operation of the AGV. Finally, the collision rate refers to the proportion of path points close to or crossing obstacles in the path, employed to reflect the safety of the path; the lower the value, the safer it is. The collision rate is measured by the proportion of the minimum distance between the path point and the obstacle being lower than the safety threshold (0.2 m). In evaluating path execution performance, the arrival time is first concerned. That is, the actual time it takes for the AGV to travel from the starting point to the destination, which evaluates the execution efficiency of the path. Path deviation indicates the average offset between the actual running trajectory of the AGV and the original planned path; smaller deviations represent higher path execution accuracy. Energy consumption reflects the energy consumed by the AGV during execution, usually in watt-hours (Wh); the lower the value, the more energy-efficient the path. Energy consumption is calculated based on the integral of motor power and operation time (unit: Wh). The stability indicator measures the amplitude of attitude changes or control fluctuations of the AGV during path execution; higher stability means less interference from the path on equipment operation and a more reliable control system.

In simulation experiments, to evaluate the model's path planning accuracy, path shortness is selected as the first indicator; it represents the current path's relative

length compared to the shortest path among all candidate paths; the closer the value is to the optimal path, the more efficient the path is. Path stability is utilized to measure the consistency of path generation results under different environmental conditions, reflecting the model's robustness to environmental disturbances. This indicator is defined as the degree of change in path shape during multiple runs; it is measured by calculating the average value of the SD of each point's position (unit: m). A lower value indicates better path repeatability. Path consistency further measures the geometric overlap degree of paths generated multiple times. This indicator uses the Jaccard similarity coefficient to measure the overlap degree of paths generated in different runs. Its value is a dimensionless number between 0 and 1; higher consistency indicates more stable algorithm decisions. Path correction ability is the reciprocal of the time required for an AGV to return to the ideal path from a deviated position (unit: s^{-1}). The error rate represents the proportion of redundant segments or unreasonable path structures in the overall path and identifies potential redundancy issues in planning. In the dimension of AGV control performance evaluation, turning radius is first considered. Turning radius is characterized by the minimum value of the path curvature radius (unit: m); its calculation method is the statistical minimum value of the arc radii formed by consecutive path points; a smaller radius means sharper turns and higher requirements for the sensitivity of the AGV control system. Speed stability measures the fluctuation of the AGV's speed during path execution; smaller speed fluctuations show more precise control and more stable path execution. Path correction ability describes whether the AGV can quickly return to the correct route after deviating from the path; stronger correction ability means better recovery ability of the system to abnormal disturbances. Response time measures the time required for the AGV system to replan the path after environmental changes; the shorter the time, the more sensitive the system's response and the more adaptable it is to dynamic environments.

5 Experimental results and analysis

5.1 Model performance evaluation

Figure 1 shows the results of the path planning performance evaluation.

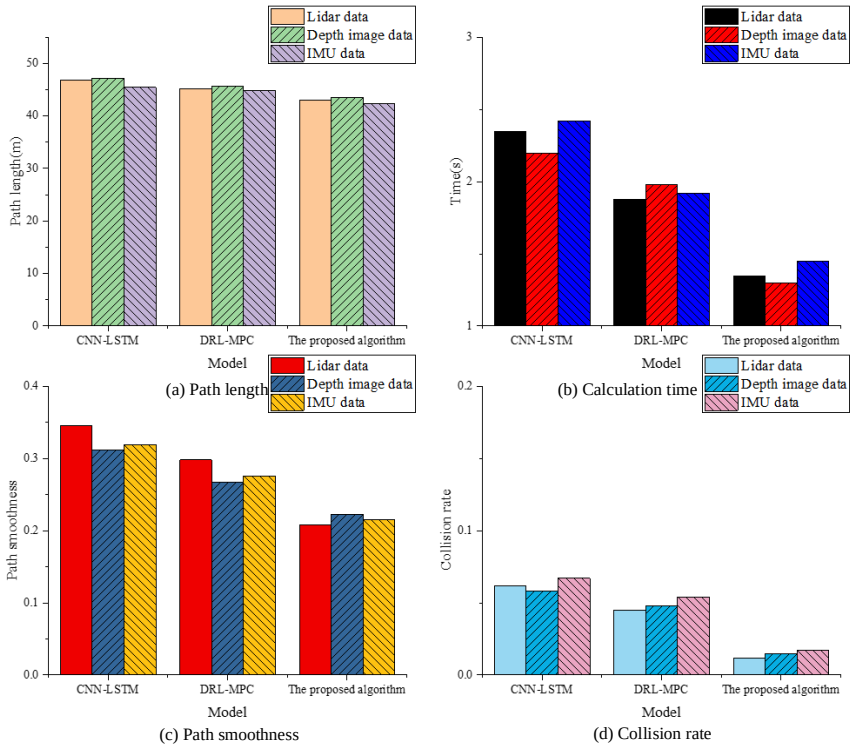


Figure 1: Performance evaluation of path planning ((a): Path length; (b): Calculation time; (c): Path smoothness; (d): Collision rate).

In Figure 1, the proposed optimized model exhibits remarkable advantages in terms of path length, especially when using LiDAR data, with a path length of 42.987 m. The optimized model is notably shorter than the CNN-LSTM (46.879 m) and DRL-MPC (45.212 m). This illustrates that the proposed model can reduce redundancy in the planned path and offer more efficient path choices. The path lengths of CNN-LSTM and DRL-MPC are longer, particularly CNN-LSTM, which demonstrates the longest path when using depth image

data. This might be due to this model choosing relatively complex paths when handling obstacles. The optimized model achieves a superior computation time of 1.345 s with LiDAR data, excelling CNN-LSTM's 2.346 s and DRL-MPC's 1.879 s. This indicates that this model has higher computational efficiency during path planning, enabling it to complete tasks more quickly. CNN-LSTM's computation time is relatively long, exceeding 2 s, especially with IMU data, likely due to the DL framework's high computational complexity. Path quality evaluation reveals the optimized model's exceptional smoothness performance (0.208 with LiDAR data), outperforming CNN-LSTM (0.345) and DRL-MPC (0.298). This suggests that the path is smoother and more continuous. CNN-LSTM and DRL-MPC have poorer path smoothness, particularly with LiDAR data, where the paths are more winding, potentially affecting the AGV's stability and precision. The optimized model's collision rate of 0.012 with LiDAR data demonstrates improvements over CNN-LSTM (0.062) and DRL-MPC (0.045). This demonstrates that the model can effectively avoid obstacles and ensure the safety of the AGV's path. CNN-LSTM and DRL-MPC have particular weaknesses with IMU data, where CNN-LSTM shows elevated collision rates (0.067). This may stem from less precise obstacle detection during the path planning process. Figure 2 presents the results of the path execution performance evaluation.

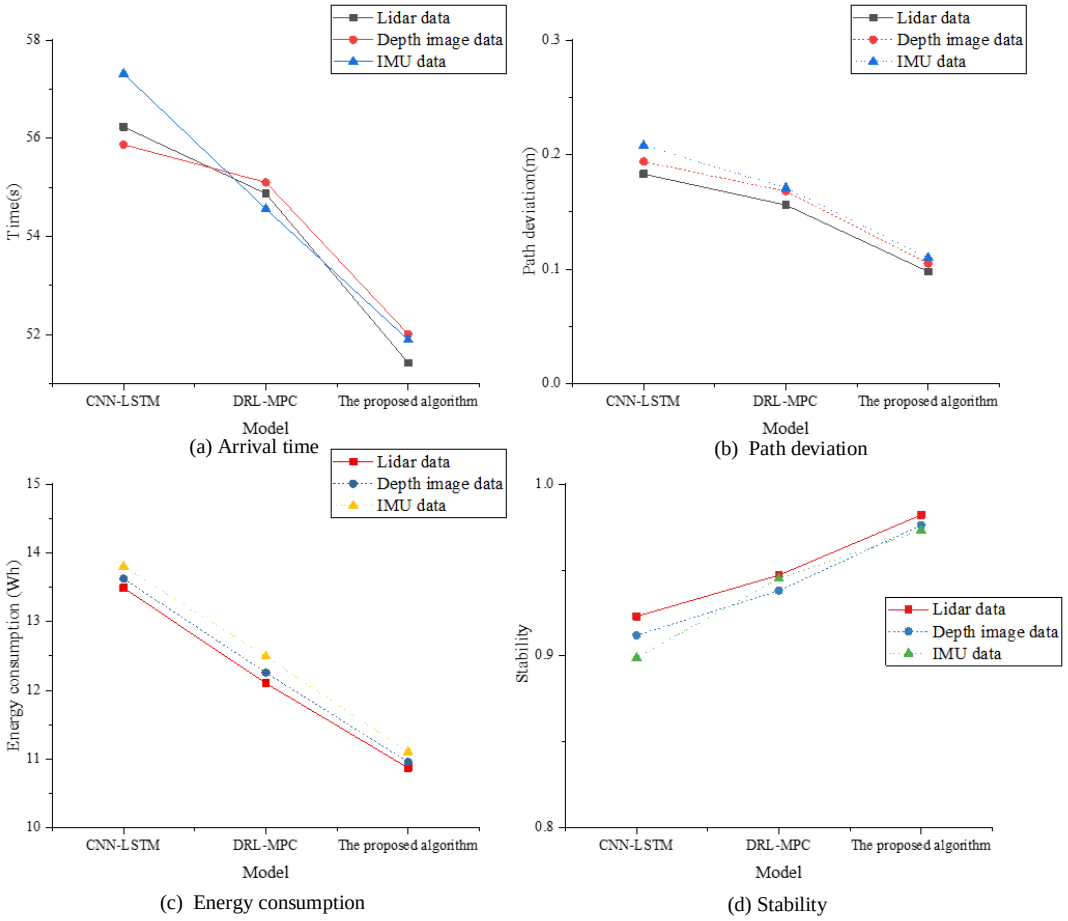


Figure 2: Performance evaluation of path execution ((a): Arrival time; (b): Path deviation; (c): Energy consumption; (d): Stability).

Figure 2 shows that the optimized model consistently achieves the shortest arrival time (51.895 s), especially with IMU data, markedly lower than CNN-LSTM (57.314 s) and DRL-MPC (54.561 s). This indicates that the optimized model exhibits superior path planning efficiency, enabling faster task completion. Notably, CNN-LSTM has the longest arrival time, particularly when processing IMU data, where it exceeds 57 s, showing its lower path planning efficiency. For path deviation, the optimized model shows the smallest value, with LiDAR data yielding a deviation of only 0.098 m, substantially lower than CNN-LSTM (0.183 m) and DRL-MPC (0.156 m). This shows that the optimized model ensures higher precision during path execution and can minimize errors. By contrast, CNN-LSTM and DRL-MPC exhibit larger path deviations, with CNN-LSTM showing particularly significant fluctuations when using depth image data, which may affect the accuracy of the AGV. Regarding energy consumption, the optimized model consistently shows the lowest energy consumption

(10.872Wh), particularly with LiDAR data, remarkably lower than CNN-LSTM's 13.489 Wh and DRL-MPC's 12.102 Wh. This dual efficiency-energy saving characteristic underscores the optimized model's operational advantages. In contrast, CNN-LSTM and DRL-MPC exhibit higher energy consumption, with CNN-LSTM's LiDAR-based results approaching 13.5 Wh; this reflects its lower path planning efficiency and resulting energy inefficiencies. In stability assessments, the optimized model is the most stable, particularly with LiDAR data, where its stability score reaches 0.982, markedly higher than CNN-LSTM (0.923) and DRL-MPC (0.947). This suggests that this model can better maintain path stability and continuity, reducing vibrations or path deviations. Conversely, CNN-LSTM has the poorest stability, especially in LiDAR scenarios, thus hindering the precise control of the AGV.

This work verifies whether the improvement of the optimized model in path planning and execution performance compared with the traditional model is statistically significant. It conducts a statistical analysis of the core indicators based on three sets of repeated experimental data using the paired t-test. The results are outlined in Table 2:

Table 2: Average ± SD of main performance indicators and paired t-test results (Taking LiDAR data as an example)

Indicator	Model name	Average ± SD	p-value of the optimized model
Path length (m)	CNN-LSTM	46.91 ± 0.24	< 0.001
	DRL-MPC	45.19 ± 0.31	0.003
	The optimized model	42.99 ± 0.18	—
Computation time (s)	CNN-LSTM	2.35 ± 0.07	< 0.001
	DRL-MPC	1.88 ± 0.04	0.005
	The optimized model	1.35 ± 0.03	—
Path smoothness	CNN-LSTM	0.347 ± 0.011	< 0.001
	DRL-MPC	0.296 ± 0.009	0.002
	The optimized model	0.208 ± 0.006	—
Collision rate	CNN-LSTM	0.064 ± 0.008	< 0.001
	DRL-MPC	0.046 ± 0.007	0.004
	The optimized model	0.012 ± 0.003	—

The results show that the proposed optimized model's improvements in all dimensions are statistically significant ($p < 0.01$). Especially in path length and collision rate, its improvements are particularly pronounced. For example, under LiDAR data, this model's path length is 42.99 ± 0.18 m, substantially outperforming CNN-LSTM (46.91 ± 0.24 m, $p < 0.001$) and DRL-MPC (45.19 ± 0.31 m, $p= 0.003$).

To further verify the contributions of each module, the supplementary ablation experiment results are listed in Table 3.

Table 3: Comparison of ablation experiment results (based on LiDAR data)

Configuration method	Path length (m)	Computation time (s)	Smoothness (rad)	Collision rate
Voronoi + A*	47.2	1.8	0.35	0.058
Voronoi + SVM	44.1	1.6	0.28	0.038
Voronoi + TSVM	42.99	1.35	0.208	0.012

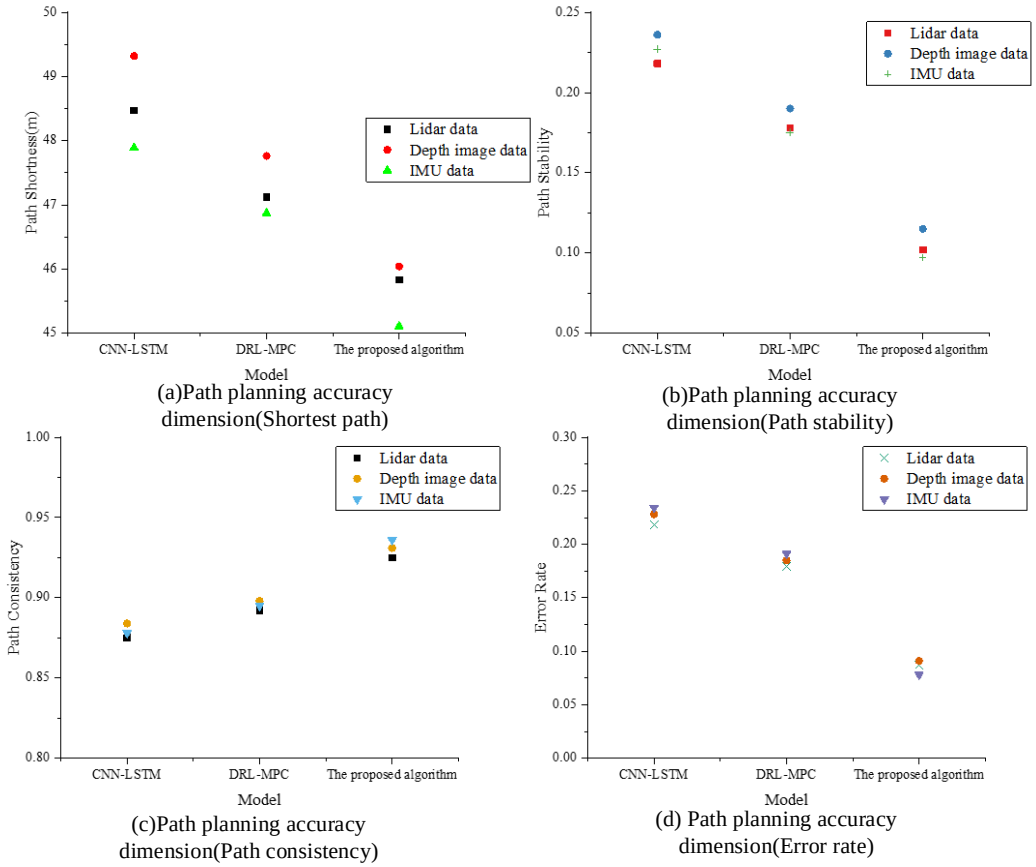
The results of the ablation experiment show that after combining the Voronoi path planning algorithm with TSVM semi-supervised optimization, the path length is shortened to 42.99 m. Meanwhile, the calculation time is reduced to 1.35 s, the smoothness is optimized to 0.208 rad, and the collision rate is reduced to 0.012. These results verify the key role of TSVM in improving navigation efficiency and safety.

5.2 Simulation experiment

Figure 3 presents the experimental results of the path planning accuracy dimension.

Figure 3 illustrates that the optimized model exhibits the best path optimality (45.11 m), especially with IMU data, shorter than CNN-LSTM (47.89 m) and DRL-MPC (46.87 m). It indicates that the optimized model reduces redundant path segments, offering a shorter and more efficient path. CNN-LSTM exhibits longer path lengths,

particularly with depth image data, reaching 49.32 m; this is because the model prioritizes more safety margins in path planning at the expense of efficiency. The optimized model also has the strongest path stability, with a score of 0.102 for LiDAR data, significantly lower than CNN-LSTM (0.218) and DRL-MPC (0.178); this indicates better smoothness during execution and reduced vibrations or fluctuations. The good performance of IMU data mainly stems from the good matching between its data characteristics and the TSVM algorithm. The attitude change information provided by IMU itself has low dimensionality and relatively regular noise. It reduces the difficulty of feature learning for TSVM when constructing classification boundaries, enabling it to focus more on judging the geometric rationality of the path. In contrast, although lidar and depth image data are rich in information, they have problems of high dimensionality and complex noise. These problems increase the burden of feature extraction and model generalization to a certain extent. Thus, when processing low-dimensional, structured data such as IMU, TSVM can more efficiently learn the judgment rules of path quality. Conversely, CNN-LSTM shows poorer path stability, especially with depth image data, where higher fluctuations may lead to instability in AGV movement. The optimized model demonstrates the highest path consistency, reaching 0.936 with IMU data, remarkably higher than the other models. This highlights that this model can provide consistent results in different environments or multiple planning scenarios, enhancing path reliability. CNN-LSTM. However, the model exhibits lower path consistency, especially with depth image data, where notable variations across multiple trials suggest instability in decision-making. Regarding error rate, the optimized model minimizes discrepancies during path planning, achieving 0.078 with IMU data, ensuring precision in path execution. CNN-LSTM has a higher error rate (0.218 for LiDAR data), indicating potential biases or inaccuracies in its path planning logic. Figure 4 illustrates the results of the AGV control performance dimension.



(a)Path planning accuracy dimension(Shortest path)

(b)Path planning accuracy dimension(Path stability)

(c)Path planning accuracy dimension(Path consistency)

(d) Path planning accuracy dimension(Error rate)

Figure 3: Path planning accuracy dimension ((a): Shortest path; (b): Path stability; (c): Path consistency; (d): Error rate).

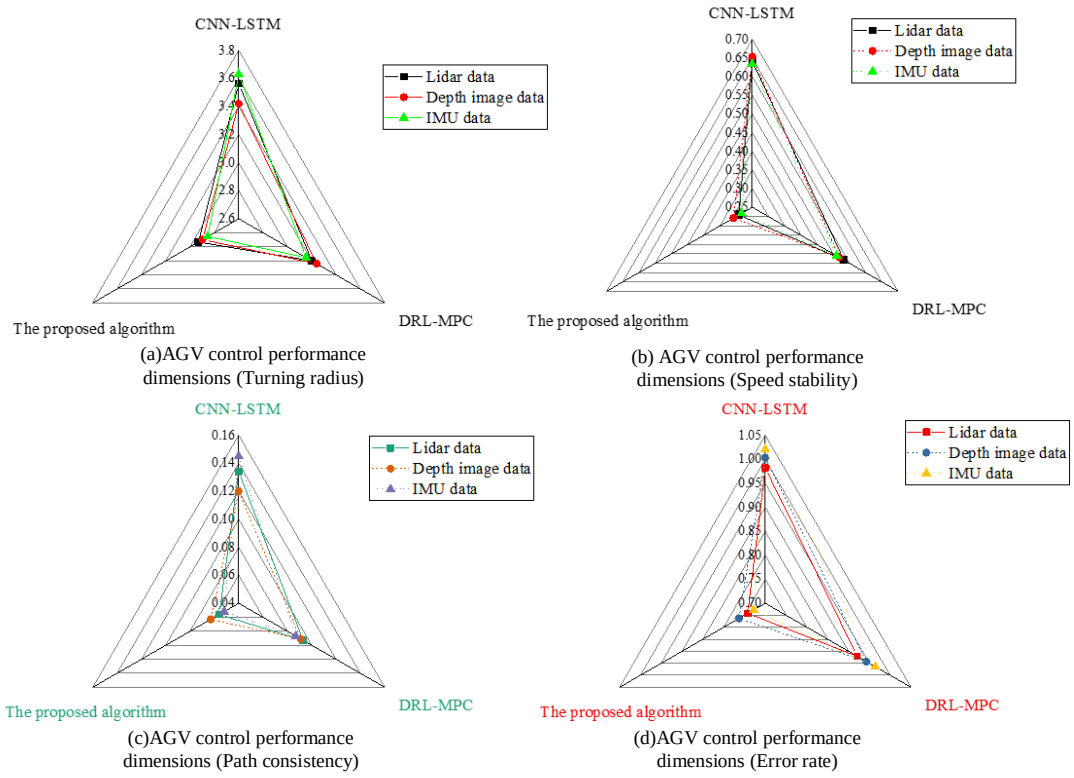


Figure 4: AGV control performance dimensions ((a): Turning radius; (b): Speed stability; (c): Path correction capability; (d): Response time).

Figure 4 shows that the optimized model achieves the smallest turning radius of 2.854m with IMU data, demonstrating better maneuverability than CNN-LSTM's 3.564 m radius with LiDAR data. This indicates enhanced flexibility in constrained environments. The optimized model exhibits the strongest speed stability (0.292) with LiDAR data, markedly lower than other models. This illustrates that the optimized model can maintain a more stable speed during path planning, avoiding unnecessary speed fluctuations. In contrast, CNN-LSTM shows relatively poorer speed stability, particularly with depth image data, where there is greater speed fluctuation, potentially affecting the AGV's operational smoothness. Regarding path correction ability, the optimized model performs best, with an IMU data-based correction score of 0.052, reflecting its capacity to rapidly and accurately realign the path when deviations or obstacles are encountered. Conversely, CNN-LSTM demonstrates weaker correction capability, particularly with IMU data (0.145), suggesting potential delays or precision limitations in its path adjustment processes. The optimized model also has the shortest response time, measuring 0.728s with IMU data, highlighting its rapid adaptability to environmental changes and efficient path reconfiguration. By comparison, CNN-LSTM has the longest response time, 1.003s with depth image data, potentially hindering real-time path adjustments and overall path efficiency.

6 Discussion

Combined with the experimental results presented earlier, this section addresses the three research questions proposed in the introduction. First, regarding the fusion mechanism of Voronoi and TSVM, the path length is shortened and smoothness is improved in the experiment; this verifies the effectiveness of the collaborative framework of topological segmentation and semi-supervised screening. Among them, the shortened path length (42.987 m) mainly benefits from the spatial topological structure provided by the Voronoi diagram, explicitly restricting the path search range and avoiding redundant exploration. The improvement in computational efficiency (1.345 s) stems from the dimensionality reduction of high-dimensional features by TSVM and the small-sample learning capability brought by its semi-supervised mechanism; this reduces dependence on labeled data and complex calculations. Second, in terms of model performance, TSVM exhibits shorter computation time and lower collision rate under all types of perceptual data. This indicates that TSVM is superior to the comparison models under different input conditions. The improvements in smoothness (0.208) and collision rate (0.012) also reflect TSVM's effective

perception of geometric continuity and safety boundaries during path segment screening. Finally, the comprehensive advantages in path consistency, energy consumption, and stability confirm the proposed method's good robustness and practicality in complex environments from multiple dimensions. However, in extremely narrow or highly dynamic environments, this method may still experience performance fluctuations due to increased time consumption for Voronoi diagram generation and blurred feature discrimination boundaries. This is a key issue to be optimized in future research.

Moreover, before experimenting, this work evaluates the computational complexity of the proposed algorithm. Under normal circumstances, the time complexity of Voronoi diagram construction is O(n log n). Among them, n refers to the number of obstacles in the environment, allowing it to efficiently handle medium-scale environments. As a semi-supervised learning algorithm, TSVM's training process is based on sequential minimal optimization. In practical applications, its computational burden mainly depends on the number of samples and feature dimensions, with a time complexity roughly between $O(n^2)$ and $O(n^3)$. In the experimental setup, the average time for generating a Voronoi diagram in a single environment is 0.8s. At the same time, TSVM's average training time is 3.7s, demonstrating good computational efficiency.

From the perspective of generalizability, TSVM functions as a classification model based on structural features. It relies on discriminative learning of path geometric characteristics, such as length, number of turns, and smoothness, rather than large-scale end-to-end training data. This enables the model to exhibit stronger adaptability and stability when switching between perception data types (e.g., LiDAR, depth images, or IMU). Particularly, its discriminative mechanisms are robust under incomplete data or uneven perception quality. In contrast, DL methods like CNN-LSTM often greatly depend on the distribution of training data and the structure of perceptual inputs. This leads to significant performance fluctuations when generalized to unseen scenarios or input changes. In terms of scalability, the proposed model has certain advantages in environmental complexity. The Voronoi diagram provides a sparse safety graph structure constructed based on obstacle distributions, naturally adapting to changes in obstacle density. This ensures reasonable basic path generation even in areas with dense obstacles or complex local structures. Meanwhile, as a lightweight classifier, TSVM has low training overhead and fast classification computation, making it suitable for deployment on resource-constrained platforms. Notably, with the increase in path dimensions (such as significantly longer paths or finer path segmentation), the model's burden in sample quantity and classification boundary complexity also grows. This requires dynamic adjustment of feature selection granularity or the addition of lightweight path screening strategies to maintain efficiency.

Regarding real-time performance, the model's operational efficiency is primarily constrained by two stages: Voronoi diagram construction and TSVM discrimination. Both stages involve well-defined and computationally stable processes. Thus, their overall time cost is lower than the policy updates and multi-round simulation training in DRL models, making the model suitable for industrial scenarios with high real-time requirements. Especially when paths need frequent updates or face dynamic obstacles, TSVM can quickly classify path segments without requiring retraining or global optimization. In terms of data source adaptability, experiments show that the model performs particularly well in processing LiDAR data. This is mainly due to the Voronoi diagram's strong reliance on geometric structures. LiDAR provides precise and dense spatial distance information, facilitating graph-structure partitioning and boundary mapping. Meanwhile, TSVM more easily constructs clear classification boundaries when handling data with obvious geometric edges. In contrast, when the input consists of depth images or IMU data, characterized by high dimensionality, noise, or temporal instability, methods like CNN-LSTM and DRL-MPC may encounter perceptual misguidance or feature extraction errors. These issues can compromise the quality and stability of path selection. The proposed model mitigates these impacts to some extent through intermediate structural feature modeling and interpretable path segment discrimination.

From the perspective of large-scale environmental scalability, this work achieves good results in medium-scale simulation environments. However, the model faces challenges in more complex maps (such as multi-story buildings, urban blocks, or cross-regional logistics systems). These difficulties include increased path search depth, higher feature diversity, and more complex path segment combinations. In such cases, Voronoi diagram construction efficiency may be limited, and TSVM discrimination rules may need regional specialization or dynamic adjustment to adapt to diverse local path patterns. Therefore, future research could consider introducing regional hierarchical strategies, adaptive feature extraction mechanisms; besides, fuzzy rules with TSVM can be combined for joint path screening to enhance the model's scalability in large-scale scenarios.

In short, the proposed model demonstrates good overall performance under current experimental conditions, with substantial advantages in structural simplicity, perceptual adaptability, and operational efficiency. However, it requires further expansion to address large-scale, extremely dense, or non-Euclidean path space scenarios. Follow-up research focuses on strengthening the model structure, simplifying training mechanisms, and enhancing real-world deployment capabilities.

7 Conclusion

This work proposes a TSVM and Voronoi diagram-based optimized AGV path planning model. The proposed model has been evaluated and compared across multiple performance dimensions through experiments and simulations. The experimental results demonstrate the optimized model's remarkable advantages in several aspects, encompassing path optimality, path stability, path consistency, error rate, turning radius, speed stability, and path correction capability. Regarding path planning accuracy, this model outperforms the comparison models in path optimality and path consistency, maintaining stable performance across diverse data dimensions. Particularly with IMU data, the optimized model effectively reduces path redundancy and improves planning efficiency. Considering AGV control performance, the optimized model performs better in turning radius and speed stability, exhibiting greater flexibility and stability. The model maintains high adaptability and control capabilities when faced with different environments and data types. Simulation results confirm the optimized model's superiority in path correction and response time, demonstrating its rapid adjustment path ability and exhibiting strong adaptability in dynamic environments.

Testing the real-time classification capability of the current TSVM model in a single AGV scenario reveals that classification time increases from a baseline of 3.7 s to 8.2 s. This occurs when the number of path segments exceeds 500. This is because the scale of the sample similarity matrix in semi-supervised learning expands. Specifically, in an environment containing 50 dynamic obstacles, the number of candidate paths generated by the Voronoi diagram increases in $O(n^2)$ (n represents the number of obstacles); this phenomenon expands the feature dimension from the standard 3 dimensions to 5-7. At this time, the kernel function calculation cost of TSVM increases markedly with the improvement of dimension. Experimental data show that when the number of path segments exceeds 800, the memory usage exceeds 1GB, which brings challenges to the deployment of edge devices.

Although the proposed AGV path planning model based on TSVM and Voronoi diagrams has shown excellent performance in multiple performance dimensions, there are still certain limitations. Subsequent research still needs to be further expanded and optimized in several aspects.

First, the discriminative capability in high-dimensional perceptual input spaces still requires further evaluation. As a classification model based on geometric boundary learning, TSVM's advantages lie in its structural simplicity and high training efficiency. However, when faced with highly complex sensor-fused data (e.g., multi-modal perception integrating LiDAR, RGB-D images, thermal imaging, sonar), the model may encounter issues like feature dimensionality explosion and sparse sample distribution during boundary construction. These issues affect the discriminative accuracy and stability of classification boundaries. Especially in adversarial scenarios with high noise or strong nonlinear data perturbations, TSVM's robustness may decline, thereby influencing the reliability of path segment discrimination and screening. Second, in large-scale scenarios, the Voronoi diagram can effectively partition environmental regions. However, the complexity of its graph structure and the number of path candidate segments increase nonlinearly, imposing higher discriminative loads on TSVM for path segments. This may cause delays in model inference time and affect the real-time performance of path planning. Consequently, future research could consider introducing regional path screening mechanisms or graph simplification algorithms to reduce redundant boundaries and enhance model scalability.

Additionally, current experiments primarily rely on static simulation environments. While covering different perceptual data types, the model's adaptive capabilities have not been fully validated in real-world scenarios with dynamic obstacle interference, multi-source perceptual conflicts, or frequent environmental semantic changes. Particularly in complex environments, a single data type cannot fully cover the environmental information required for AGV decision-making. Follow-up research could consider introducing multi-sensor fusion mechanisms and exploring the integration of a state credibility evaluation module at the decision-making layer to enhance the model's discriminative ability for environmental changes. Finally, this work does not address multi-AGV path coordination and collaborative planning. In practical applications, multiple AGVs coordinate obstacle avoidance in shared spaces, ensure non-interfering paths, and complete collaborative task allocation. To address this need, future work could explore collaborative Voronoi area partitioning strategies, introducing multi-agent avoidance mechanisms at the path generation stage. Meanwhile, a federated TSVM learning framework is introduced to enable collaborative training of path evaluation model parameters among multiple AGVs without sharing raw data. Thus, path selection efficiency, communication load control, and global coordination capabilities can be improved in multi-intelligent agent systems.

In addition, although the model is designed to support multi-sensor input, the analysis and discussion of the experimental results are indeed mainly focused on lidar data. Results of depth image and IMU data are collected in the experiment. For example, IMU data performs best in path stability, and depth images have unique advantages in complex texture environments. However, due to differences in data characteristics and noise patterns, their performances in different indicators have various focuses, so they cannot be fully expanded. This arrangement may affect the complete presentation of the model's cross-sensor generalization ability to a certain extent. Future

research could further improve this aspect of work through a more balanced experimental design and analysis.

In conclusion, the current model has demonstrated excellent path planning performance in the simulation environment. However, its robustness under high-dimensional perceptual input, real-time response capability in complex scenes, and portability in multi-AGV systems remain important research directions for the future. Based on algorithm structure optimization and system-level integrated design, the model can be further extended to more challenging industrial and urban automatic navigation applications.

References

[1] Chen, X., Liu, S., Zhao, J., Wu, H., Xian, J., & Montewka, J. (2024). Autonomous port management based AGV path planning and optimization via an ensemble reinforcement learning framework. Ocean & Coastal Management, 251, 107087. https://doi.org/10.1016/j.ocecoaman.2024.107087

[2] Hu, H., Yang, X., Xiao, S., & Wang, F. (2023). Anti-conflict AGV path planning in automated container terminals based on multi-agent reinforcement learning. International Journal of Production Research, 61(1), 65-80. https://doi.org/10.1080/00207543.2021.1998695

[3] Jiang, Z., Zhang, X., & Wang, P. (2023). Grid-map-based path planning and task assignment for multi-type AGVs in a distribution warehouse. Mathematics, 11(13), 2802. https://doi.org/10.3390/math11132802

[4] Fransen, K., & Van Eekelen, J. (2023). Efficient path planning for automated guided vehicles using A*(Astar) algorithm incorporating turning costs in search heuristic. International journal of production research, 61(3), 707-725. https://doi.org/10.1080/00207543.2021.2015806

[5] Ye, S., & Liu, N. (2024). Automated logistics control model based on improved ant colony algorithm. Informatica, 48(16). https://doi.org/10.31449/inf.v48i16.6371

[6] Ho, G. T. S., Tang, Y. M., Leung, E. K., & Tong, P. H. (2025). Integrated reinforcement learning of automated guided vehicles dynamic path planning for smart logistics and operations. Transportation Research Part E: Logistics and Transportation Review, 196, 104008. https://doi.org/10.1016/j.tre.2025.104008

[7] Machavaram, R. (2025). Intelligent path planning for autonomous ground vehicles in dynamic environments utilizing adaptive Neuro-Fuzzy control. Engineering Applications of Artificial Intelligence, 144, 110119. https://doi.org/10.1016/j.engappai.2025.110119

[8] Bao, Q., Zheng, P., & Dai, S. (2024). A digital twin-driven dynamic path planning approach for multiple automatic guided vehicles based on deep reinforcement learning. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 238(4), 488-499. https://doi.org/10.1177/09544054231180513

[9] Ming, S. I., Bofan, W. U., Can, H. U., & Weiqiang, X. I. N. G. (2024). Collaborative control method of intelligent warehouse traffic signal and multi-AGV path planning. Journal of Computer Engineering & Applications, 60(11). https://doi.org/10.3778/j.issn.1002-8331.2302-0113

[10] Pan, Y. (2023). Autonomous path planning for industrial omnidirectional AGV based on mechatronic engineering intelligent optical sensors. International Journal of Advanced Computer Science and Applications, 14(5). https://doi.org/10.14569/IJACSA.2023.0140582

[11] Gul, O. M. (2024). Energy-aware 3d path planning by autonomous ground vehicle in wireless sensor networks. World Electric Vehicle Journal, 15(9), 383. https://doi.org/10.3390/wevj15090383

[12] Chek, L. W. (2023). Low latency extended dijkstra algorithm with multiple linear regression for optimal path planning of multiple AGVs network. Engineering Innovations, 6, 31-36. https://doi.org/10.4028/p-t122xr

[13] Rousselet, D. (2024). Reinforcement Learning for AGV Path Planning and Task Selection: An Industrial Use Case of an Austrian Manufacturing Company.

[14] Jia, B. (2024). Reservoir irrigation operation design based on dijkstra algorithm combined with aco algorithm. Informatica, 48(12). https://doi.org/10.31449/inf.v48i12.6005

[15] Barbierato, E., & Gatti, A. (2024). Decoding urban intelligence: clustering and feature importance in smart cities. Future Internet, 16(10), 362. https://doi.org/10.3390/fi16100362

[16] Dong, D., Feng, M., Kurths, J., & Zhang, L. (2024). Fuzzy large margin distribution machine for classification. International Journal of Machine Learning and Cybernetics, 15(5), 1891-1905. https://doi.org/10.1007/s13042-023-02004-3

[17] Aláez, D., Prieto, M., Villadangos, J., & Astrain, J. J. (2024). On constructing efficient uav aerodynamic surrogate models for digital twins. IEEE Transactions on Industrial Informatics, 20(11), 13181-13189. https://doi.org/10.1109/TII.2024.3431106

[18] Sang, W., Yue, Y., Zhai, K., & Lin, M. (2024). Research on AGV path planning integrating an improved A* algorithm and DWA algorithm. Applied Sciences, 14(17), 7551. https://doi.org/10.3390/app14177551

[19] Phua, A., Smith, J., Davies, C. H., Cook, P. S., & Delaney, G. W. (2023). Understanding the structure and dynamics of local powder packing density variations in metal additive manufacturing using set

Voronoi analysis. Powder Technology, 418, 118272. https://doi.org/10.1016/j.powtec.2023.118272

[20] Srivastava, V., Basu, B., & Prabhu, N. (2024). Application of Machine Learning (ML)-based multi-classifications to identify corrosion fatigue cracking phenomena in Naval steel weldments. Materials Today Communications, 39, 108591. https://doi.org/10.1016/j.mtcomm.2024.108591

[21] Bolyachkin, A., Dengina, E., Kulesh, N., Tang, X., Sepehri-Amin, H., Ohkubo, T., & Hono, K. (2024). Tomography-based digital twin of Nd-Fe-B permanent magnets. npj Computational Materials, 10(1), 34. https://doi.org/10.1038/s41524-024-01218-5

[22] Noh, H., Son, G., Kim, D., & Park, Y. S. (2024). H-ADCP-based real-time sediment load monitoring system using support vector regression calibrated by global optimization technique and its applications. Advances in Water Resources, 185, 104636. https://doi.org/10.1016/j.advwatres.2024.104636

[23] Taghavi, M., & Perera, L. P. (2024). Advanced data cluster analyses in digital twin development for marine engines towards ship performance quantification. Ocean Engineering, 298, 117098. https://doi.org/10.1016/j.oceaneng.2024.117098

[24] Zhang, J., Zhang, X., Han, M., Ren, N., & Hou, Y. (2024). AGV path planning method and intelligent obstacle avoidance strategy for intelligent manufacturing workshops. Journal of Computers, 35(6), 137-151. 10.53106/199115992024123506011

[25] Mei, Y., Liu, Y., Sun, C., Wang, X., Wang, D., Yuan, L., & Tan, J. (2023). Multi-stage rotors assembly of turbine-based combined cycle engine based on augmented reality. Advanced Engineering Informatics, 58, 102160. https://doi.org/10.1016/j.aei.2023.102160

[26] Pratissoli, F., Brugioni, R., Battilani, N., & Sabattini, L. (2023). Hierarchical traffic management of multi-AGV systems with deadlock prevention applied to industrial environments. IEEE Transactions on Automation Science and Engineering, 21(3), 3155-3169. 10.1109/TASE.2023.3276233

[27] Chen, Y., Shi, S., Chen, Z., Wang, T., Miao, L., & Song, H. (2023). Optimizing port Multi-AGV trajectory planning through priority coordination: enhancing efficiency and safety. Axioms, 12(9), 900. https://doi.org/10.3390/axioms12090900

[28] Lei, Y., Hou, J., Ma, P., & Ma, M. (2025). Voronoi-GRU-based multi-robot collaborative exploration in unknown environments. Applied Sciences, 15(6), 3313. https://doi.org/10.3390/app15063313

[29] Lin, S., Liu, A., & Wang, J. (2023). A dual-layer weight-leader-vicsek model for multi-agv path planning in warehouse. Biomimetics, 8(7), 549. https://doi.org/10.3390/biomimetics8070549

[30] Zhang, R., Chai, R., Chai, S., Xia, Y., & Tsourdos, A. (2023). Design and practical implementation of a high efficiency two-layer trajectory planning method for AGV. IEEE transactions on industrial electronics, 71(2), 1811-1822. 10.1109/TIE.2023.3250847

[31] Tan, C., & Li, P. (2025). The application of logistics robot in the solution of locating route problems in trans CAD. Informatica, 49(11). https://doi.org/10.31449/inf.v49i11.6602