Article | Received 30 September 2025; Revised 15 January 2026; Accepted 16 January 2026; Published 5 February 2026
https://doi.org/10.55092/rl20260005

Path Planning Transformers supervised by IRRT*-RRMS for multi-mobile robots

Aphilak Lonklang* and János Botzheim

Department of Artificial Intelligence, Faculty of Informatics, ELTE Eötvös Loránd University, Budapest, Hungary

* Correspondence author; E-mail: aphilak@inf.elte.hu.

Highlights:

- A novel Path Planning Transformer supervised by IRRT*-RRMS is proposed for mobile robot navigation.
- The model eliminates the need for graph-based algorithms or conventional SLAM during deployment.
- A modified right-of-way rule with 270-degree virtual obstacles enables dynamic obstacle avoidance and re-planning.
- Real-world experiments with two robots validate the system's collision-free behavior and trajectory adaptation.

Abstract: This paper proposes a learning-based path planning framework for autonomous mobile robots based on a novel Transformer architecture supervised by an Improved RRT* with Reduced Random Map Size (IRRT*-RRMS) algorithm. The system, referred to as the Path Planning Transformer (PPT), predicts sequences of intermediate waypoints from occupancy maps and eliminates the need for traditional graph-based algorithms or SLAM during online execution. The model is trained using a dataset of 10,000 samples generated from 100 randomly created maps, each containing 100 distinct start-goal trajectories. Ground truth paths are computed using IRRT*-RRMS, which enhances classical RRT* through dynamic sampling region reduction, node deletion, and bacterial mutation-based smoothing. To ensure robustness in dynamic environments, a modified right-of-way rule is introduced as a post-planning collision avoidance mechanism. This rule employs a 270-degree virtual obstacle ring projected in front of the robot upon LiDAR-based detection of unknown dynamic obstacles. It enables the system to trigger safe re-planning that biases the trajectory away from oncoming objects—especially in multi-robot scenarios. Real-world experiments using two TurtleBot3 Waffle Pi robots demonstrate the framework's capability to dynamically avoid collisions while maintaining progress toward assigned goals. In scenarios where robots initially face each other, the system successfully replans their paths without direct communication, relying solely on onboard sensing and map-based planning. The PPT model was trained for 429 epochs, achieving a best validation loss of 186. Experimental results confirm that the approach generates efficient, smooth, and collision-free trajectories in both simulation and physical deployment. The proposed framework provides a scalable solution for map-driven navigation in indoor environments and opens the door to

future integration with higher-level task-planning and semantic-reasoning systems.

Keywords: Path Planning Transformer; IRRT*-RRMS; multi-robot navigation; dynamic obstacle avoidance

1. Introduction

Autonomous mobile robots (AMRs) have become integral to operations in smart factories, warehouses, and healthcare systems, where safe and efficient navigation is vital. Central to this autonomy is path planning, the process of determining an obstacle-free trajectory from a start to a goal state within a spatial environment. Classical algorithms—such as A*, Dijkstra's algorithm, and Rapidly-exploring Random Trees (RRT)—form the backbone of traditional planning approaches. These methods are supported by theoretical guarantees regarding completeness and optimality [1,2]. Researchers often rely on heuristic tuning, suffer from reduced performance on dense or high-resolution maps, and can become computationally prohibitive in simultaneous contexts [3].

In recent years, a growing body of work has employed *deep learning* to address these limitations. Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), and Reinforcement Learning (RL) frameworks have been developed to derive navigation policies or generate full trajectory proposals from sensory data [4,5]. However, many such methods depend on *vision-based input*—including RGB, depth, or semantic maps—which can be vulnerable to lighting changes, occlusions, and texture variations commonly found in autonomous driving settings [6]. As an alternative, *LiDAR-based occupancy maps* provide a robust spatial representation that abstracts environmental geometry effectively and remains reliable under variable conditions [7].

Occupancy grid mapping, introduced by Moravec and Elfes and formalized in probabilistic robotics literature [2], discretizes space into cells representing free or occupied regions. This representation efficacy led to numerous applications in mapping and navigation [8], especially due to its compatibility with Bayesian inference and dynamic update procedures. More recently, occupancy grids have been leveraged within deep learning frameworks for tasks such as dynamic obstacle prediction and static map-based navigation, with notable approaches including RNN- and CNN-based architectures [4,9]. However, capturing *long-range spatial dependencies* required for trajectory planning remains challenging for these models.

Recent advances in deep learning have significantly impacted the field of robot path planning, particularly through the integration of Transformer-based architectures. While traditional algorithms such as A* and RRT* remain reliable in static environments, they often struggle in real-time, multi-agent, or partially known contexts. To overcome these limitations, researchers have proposed learning-based alternatives that combine spatial awareness with sequential decision-making. Transformer models have shown strong performance in vision-based navigation tasks [10], as demonstrated by NavFormer, which uses visual tokens and attention mechanisms for path prediction in indoor scenes [11]. Similarly, actor-critic architectures enhanced with LSTM have been proposed for handling continuous motion in unknown environments [12]. For multi-robot systems, learning-based methods using Graph Neural Networks (GNNs) have emerged as a compelling direction, offering distributed coordination and scalability. For instance, cooperative multi-robot path planning has been achieved through task-motion

coordination with GNNs [13], and graph attention mechanisms have been used to model socially-aware navigation behaviors [14]. A decentralized approach to navigation via distributed GNNs further addresses scalability in large teams [13]. These efforts collectively motivate our adoption of a Transformer-based path planner and its integration into a rule-based multi-robot navigation pipeline, using right-of-way virtual obstacles.

To address these challenges, this work introduces the PPT, a transformer-based model designed for direct sequence generation of collision-free waypoints from LiDAR-derived occupancy grids. Operating without visual input improves resilience in challenging lighting and texture conditions found in industrial environments. The transformer encoder processes the global grid representation, while the decoder outputs discrete (x, y) waypoint sequences.

For training, we synthesize a dataset of 10,000 unique occupancy-map scenarios, each with an optimal path obtained via a sampling-based planner. To ensure both efficiency and quality in these ground-truth trajectories, we employ an enhanced version of RRT* called *Improved RRT* with Reduced Random Map Size (IRRT*-RRMS)*. This method builds upon a series of earlier RRT* enhancements, including the introduction of right-of-way rules for multi-robot coordination, obstacle avoidance in unknown static environments, reduced sampling strategies, and bacterial mutation with node deletion techniques for trajectory refinement [15]. IRRT*-RRMS dynamically restricts the sampling region based on progress toward the goal, removes redundant nodes from the search tree, and applies biologically inspired mutation for trajectory smoothing. This approach builds upon ethological control systems [16] and produces efficient, near-optimal, and smooth paths suitable for large-scale supervised learning. The occupancy maps used in this work are constructed using a mapping technique introduced in our prior work on LLM-based task planning for materials distribution [17], which integrates a lightweight SLAM method tailored for mobile robot navigation in structured environments.

In the multi-robot setup, the PPT is used independently by each robot to perform its own motion planning. The system does not rely on joint trajectory prediction or inter-robot communication. Instead, when a robot detects another one as an unknown obstacle via its onboard LiDAR, it inserts a virtual obstacle into the map using a modified right-of-way rule. This triggers the PPT model on each robot to re-plan its path based on the updated occupancy map, ensuring conflict-free and collision-avoidant trajectories.

We evaluate PPT on metrics including path optimality and collision avoidance. The results show that PPT not only competes with classical planners in structured environments but also enables collaboration-sensitive, real-time planning across two robots—demonstrating a scalable solution for industrial multi-robot applications.

2. Methods

2.1. System overview

The proposed framework consists of a modular pipeline designed for efficient path planning in structured indoor environments. The system is composed of three main stages: map generation, path label computation, and trajectory prediction using the PPT. The input to the system is a LiDAR-derived 2D occupancy grid with annotated robot start and goal positions. The output is a sequence of waypoints

forming a collision-free path. For multi-robot scenarios, the framework predicts independent trajectories for each robot based on the shared map input. Figure 1 illustrates the overall workflow of the proposed Path Planning Transformer system. The process begins with a dataset of procedurally generated occupancy grid maps with a resolution of 224×224 pixels, populated with randomly placed obstacles. These maps generate ground-truth trajectories using the IRRT*-RRMS algorithm. Each map is paired with 100 randomly sampled start and goal point pairs, producing 100 optimal paths per map. The resulting trajectories are saved as individual .csv files. To align with the input requirements of the Vision Transformer (ViT) encoder [18], the original maps are ready for training set as size of 224×224 pixels and divided into non-overlapping patches of size 32×32 pixels, yielding 49 patches per map. In this case, maps are usable without resizing. Each patch is embedded and combined with positional encodings, while the corresponding paths are tokenized and embedded for decoder input. The transformer model is trained using these map-path pairs. At inference time, the trained PPT model receives a new map, specified start and goal positions, and predicts a sequence of waypoints. If needed, a post-processing module based on bacterial mutation and node deletion is applied to refine the trajectory further.

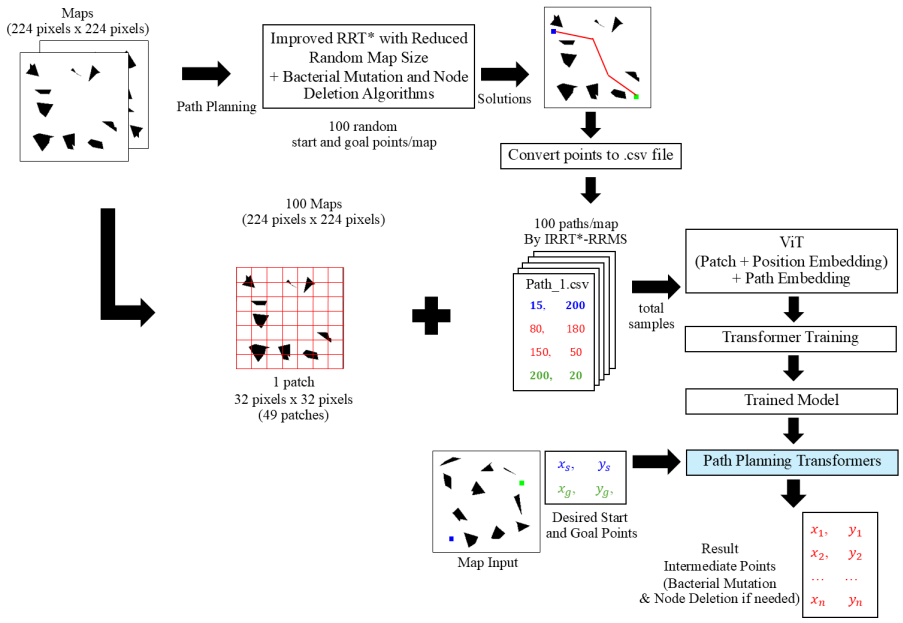


Figure 1. Overview of the Path Planning Transformer pipeline.

2.2. Occupancy map generation

Figure 2 shows the experimental environment for validating the proposed system, where two mobile robots operate within a bounded indoor space with wooden furniture and textured flooring. This real-world scenario was mapped using our previously proposed vision-based SLAM technique [17], which relies on top-view camera snapshots and color-based segmentation. The resulting occupancy grid map, shown in Figure 3, accurately captures the spatial layout and boundaries of the environment and is used in ROS as the static map for planning and localization. This approach provides a lightweight and flexible alternative to conventional LiDAR-based SLAM, enabling rapid map construction from a single calibrated image.

Figure 2. Real-world experimental scenario with two mobile robots deployed in a structured indoor environment.

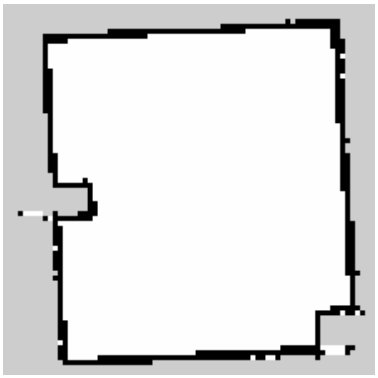
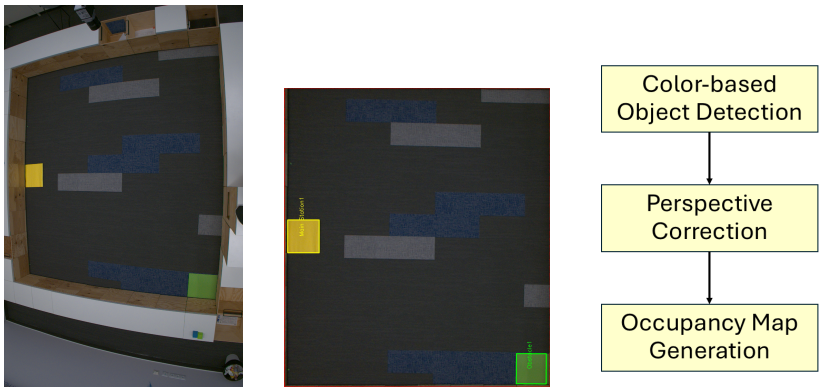


Figure 3. Occupancy grid map generated using the vision-based occupancy map generation method [17], used for navigation in ROS.

To clarify the mapping procedure used in this work, although the proposed system eliminates the need for traditional SLAM during deployment, a static occupancy map is indeed required and was generated using a vision-based occupancy map generation pipeline developed in our prior work. As illustrated in Figure 4, the process begins with color-based object detection on a top-down fisheye image captured by an overhead camera. Distinct colors are assigned to environmental features, such as the main station (yellow) and obstacles (green). Then, a homographic transformation is applied to perform perspective correction and rectify distortions caused by the camera's wide-angle lens. The final output is a clean, top-down binary occupancy map suitable for path planning.



(a) (b) (c)

Figure 4. Vision-based occupancy map creation from top-down camera input. **(a)** Raw fisheye view with labeled regions; **(b)** Perspective corrected image; **(c)** Occupancy map generation pipeline.

2.3. *Path generation using IRRT*-RRMS*

Ground-truth paths in this study are generated using the IRRT*-RRMS algorithm, as introduced in our previous works [15]. This enhanced variant of the classical RRT* algorithm incorporates four key modifications designed to improve planning efficiency and trajectory quality. First, the dynamic sampling region reduction technique focuses exploration on areas close to the goal, significantly reducing unnecessary branching. Second, a node deletion mechanism is applied to eliminate redundant nodes, leading to a more concise and interpretable tree structure. Third, bacterial mutation-based smoothing is employed to refine the path curvature, ensuring smoother and more realistic trajectories. Lastly, a right-of-way rule is integrated to handle multi-robot coordination and avoid conflicts during simultaneous navigation. Together, these improvements allow IRRT*-RRMS to produce efficient, smooth, and collision-free trajectories, which serve as high-quality supervision data for training the learning-based path planner. These modifications result in efficient, smooth, and collision-free trajectories used for supervised training.

Figure 5 illustrates an example of path generation using the IRRT*-RRMS algorithm. In this example, the green lines and dots represent the search tree explored by RRT*, capturing the rapid expansion of candidate trajectories in free space. Once a path from the start to the goal is found, the initial solution is shown in blue. Although feasible, this first path may include unnecessary detours and sharp turns. Post-processing steps are applied to improve efficiency and smoothness, specifically node deletion and bacterial mutation, resulting in the final trajectory shown as a bold red line. This refined path minimizes redundant nodes and sharp angles, making it more suitable for downstream learning and execution. To smoothen the final path and reduce unnecessary waypoints, two post-processing algorithms were employed: Bacterial Mutation and Node Deletion. These algorithms refine the initially planned path by optimizing curvature and pruning redundant intermediate points, resulting in a more efficient and natural trajectory for the mobile robot.

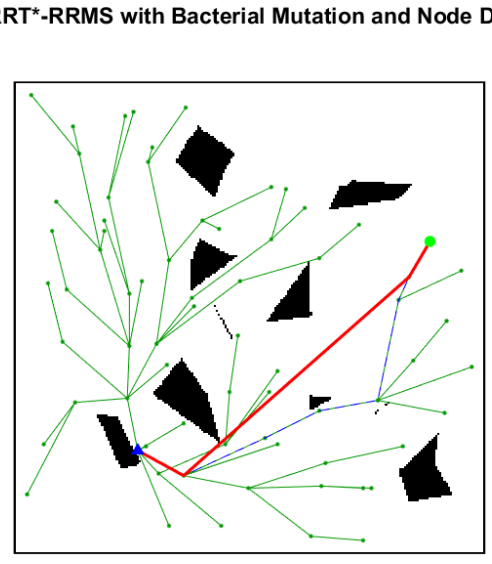


Figure 5. Visualization of a path generated by IRRT*-RRMS with post-processing algorithms. Start (blue triangle), goal (green circle), searching tree (green lines), final solution (red lines).

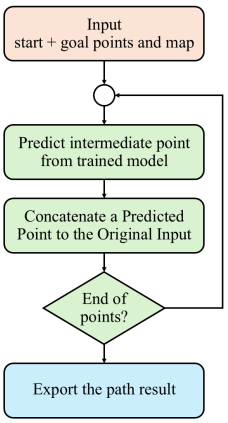
2.4. *Transformer-based path prediction*

The PPT model predicts a sequence of waypoints from an input occupancy grid. The encoder module tokenizes the grid into non-overlapping patches, embeds them, and applies multi-head self-attention to capture global spatial features. The decoder receives a special start token and autoregressively generates the path in coordinate form (x, y). In the dual-robot case, the model is conditioned to produce two separate sequences concurrently. Training is performed using the custom loss function. The loss function (Equation (1)) combines coordinate MSE and type classification cross-entropy.

$$L = \frac{1}{S}\sum_{j=1}^{S}\left(\frac{1}{n}\sum_{i=1}^{n}\left((x_{ij}-\hat{x}_{ij})^2+(y_{ij}-\hat{y}_{ij})^2-t_{ij}\log p_{ij}\right)\right) \tag{1}$$

where S is the number of samples, n is the number of intermediate points per sample, (x_{ij}, y_{ij}) and $(\hat{x}_{ij}, \hat{y}_{ij})$ are the predicted and ground truth coordinates of point i in sample j, respectively, t_{ij} is the one-hot encoded ground truth class of point i, and p_{ij} is the predicted softmax probability for that class. The MSE term encourages the network to generate geometrically accurate trajectories, while the cross-entropy term ensures that the model correctly identifies the semantic role of each point. Importantly, the *stop* token enables the model to terminate the sequence without requiring a fixed waypoint count.

Figure 6 illustrates the step-by-step prediction pipeline of the PPT. Given an input map and a pair of start and goal coordinates, the trained model iteratively predicts intermediate waypoints. Each predicted point is concatenated to the input sequence and used as part of the context for the next prediction, forming an autoregressive loop. The process terminates when a special end token (represented by coordinate $(0, 0)$) is predicted, signaling the completion of the path. The right-hand side of the diagram shows the structure of the output in tabular form, labeling each coordinate by its type (e.g., start, goal, intermediate, stop).



	x	y	Type
	51	21	0-Start
	14	26	1-Goal
Iteration 1	x_1	y_1	2-Intermediate
Iteration 2	x_2	y_2	2-Intermediate
…	…	…	2-Intermediate
Iteration n	x_n	x_n	2-Intermediate
Iteration n+1	0	0	4-Stop

Figure 6. Iterative prediction process of the Path Planning Transformer.

Figure 7 shows a representative path predicted by the PPT model. The red line indicates the complete predicted trajectory from the blue triangular start point to the green circular goal, with intermediate points shown as cyan markers. The robot successfully learns to navigate around obstacles based solely on the occupancy grid input, without explicit obstacle-aware logic inside the transformer itself. This demonstrates the model's capacity to internalize spatial reasoning and obstacle avoidance from data.

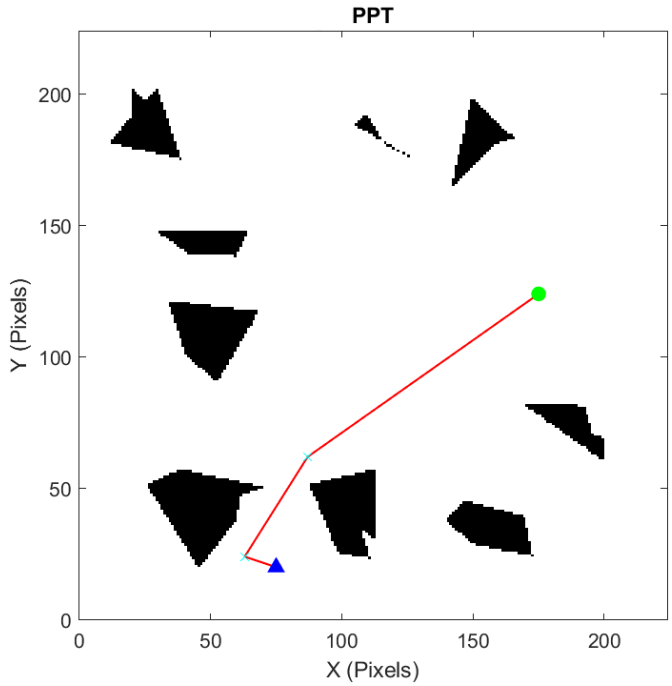


Figure 7. Example of a predicted path from the PPT model navigating around obstacles from the start (blue) to goal (green) via learned intermediate points (cyan).

2.5. Modified right-of-way rule for unknown dynamic obstacles avoidance

We implement a modified version of the right-of-way rule to handle unknown dynamic obstacles during navigation, as illustrated in Figure 8. In our prior work, a circular region was used to enforce yielding behavior around detected obstacles. This study extends that concept by introducing an asymmetrical virtual obstacle—a 270-degree arc—that selectively blocks the left side of the robot's forward field of view. This virtual arc is injected into the map when the LiDAR sensor detects a previously unknown obstacle in front of the robot. The asymmetry of the arc biases the PPT to re-plan trajectories toward the right-hand side, enforcing a rule-based local decision policy that mimics real-world traffic yielding behavior. This allows for effective collision avoidance in multi-robot systems.

2.5.1. Multi-robot conflict avoidance via right-of-way rule

In this study, the trained PPT model is applied independently to each robot. The model does not employ joint or decentralized attention mechanisms for simultaneous trajectory prediction. Instead, coordination between robots is achieved by modifying the map representation in real-time based on detected dynamic obstacles.

Each robot is equipped with a LiDAR sensor that identifies other robots as unknown obstacles when encountered. Upon detection, a virtual obstacle in the form of a 270-degree arc is generated in front of the robot based on the modified right-of-way rule, as illustrated in Figure 8. This obstacle is embedded into the local map, prompting the robot to re-plan a path that avoids the blocked region—typically favoring the right-hand side. The right-of-way rule ensures that one robot yields to the other to avoid potential collisions.

This decentralized mechanism allows each robot to maintain independent planning using the same trained model, while safety and coordination are handled through spatial map updates rather than communication or multi-agent learning.

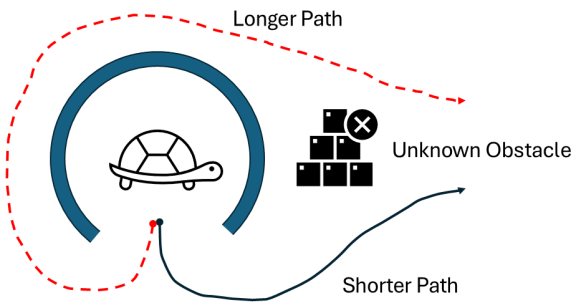


Figure 8. Modified right-of-way rule: A 270-degree virtual arc is placed in front of the robot upon detection of a dynamic obstacle, forcing the PPT to re-plan toward the right-hand side of the field.

3. Results

3.1. Training and validation performance

The PPT model was trained using a custom dataset comprising 100 randomly generated maps, each containing 100 path samples, leading to a total of 10,000 data points. The model was trained over 429 epochs. The loss log during training is depicted in Figure 9.

As shown, the training loss (red) consistently decreased throughout the epochs, while the validation loss (magenta) exhibited minor oscillations. Despite the fluctuations, the validation curve followed a general decreasing trend, indicating a successful learning process with minimal overfitting. The best validation loss achieved was 186, suggesting the strong ability of the model to generalize to unseen inputs.

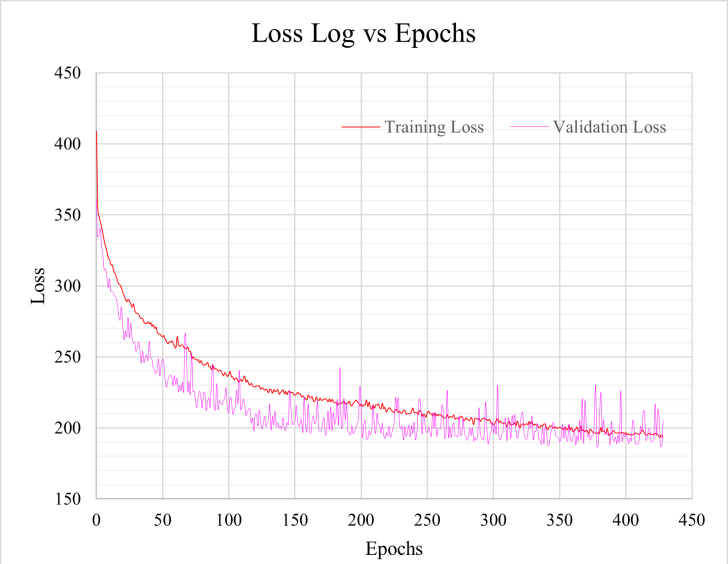


Figure 9. Training and validation loss logs over 429 epochs.

3.2. Multi-robot replanning with modified right-of-way rule

To evaluate the effectiveness of the proposed PPT, we compared it against two classical path planning algorithms: A* and RRT*. The experiments were conducted in a simulated occupancy map environment, both before and after re-planning triggered by the detection of a dynamic obstacle. The modified right-of-way rule was applied during re-planning. Figure 10 illustrates the real environment and the two robots used in

the experiments. The path results from three algorithms are shown in Figure 11 for Robot 1 which appears on the left side in Figure 10. The paths before re-planning from A*, RRT*, and PPT algorithms are shown in Figure 11a–c, respectively and after re-planning they are shown in Figure 11d–f, respectively. All experiments and simulations were implemented on a standard laptop running MATLAB, ROS, and Gazebo, equipped with an 11th Gen Intel® Core™ i7-1165G7 CPU @ 2.80 GHz and 16 GB of RAM.

Figure 10. Real environment.

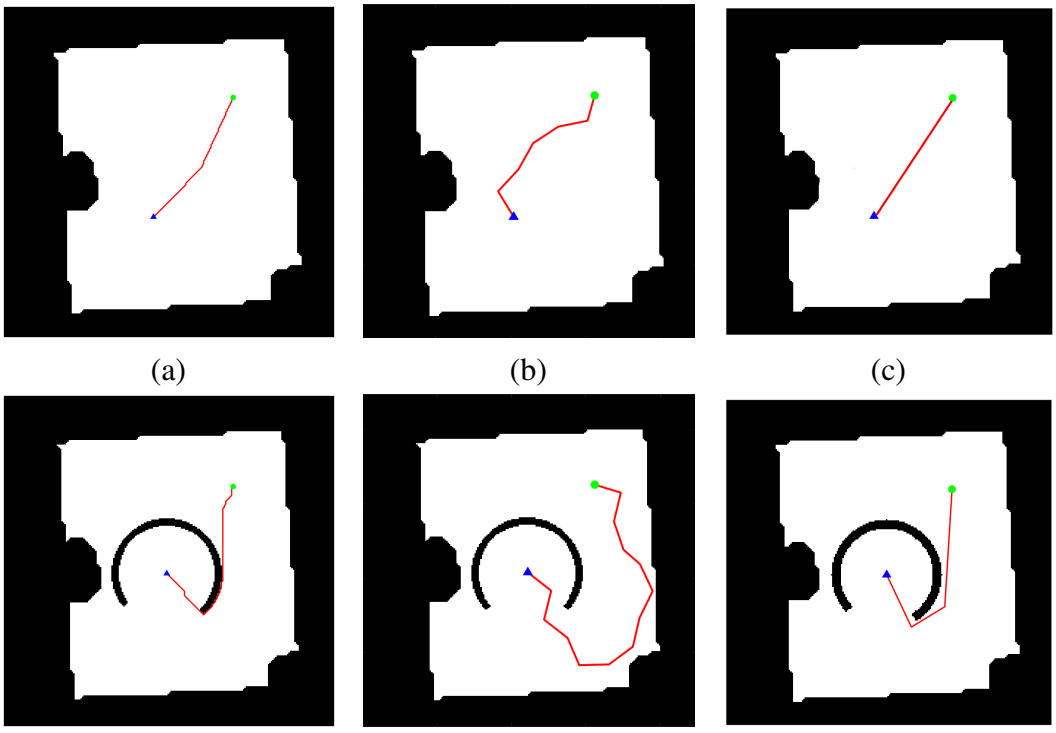


Figure 11. Re-planning behavior of Robot 1 under dynamic obstacle detection using the modified Right-of-Way rule by A*, RRT* and PPT algorithms from the start (blue) to the goal (green) via solution path (red). **(a)** A* – Initial path; **(b)** RRT* – Initial path; **(c)** PPT – Initial path; **(d)** A* – Replanned path; **(e)** RRT* – Replanned path; **(f)** PPT – Replanned path.

3.2.1. Robot 1: before re-planning

The initial evaluation of the three planners, which are PPT, A*, and RRT*, was conducted in a known environment, assuming no unexpected obstacles. The numerical results are shown in Table 1. The results indicated that the PPT achieved the shortest path length of 97.35 pixels, outperforming A* with 103.37 pixels and RRT* with 117.48 pixels. Moreover, PPT produced the smoothest trajectory with zero turns, while A* and RRT* generated paths with 5 and 2 turns, respectively, indicating better global path consistency by PPT. Although PPT had a slightly longer computation time of 0.09 seconds compared to RRT* (0.023 seconds) and A* (0.052 seconds), it remained well within practical real-time limits and acceptable margins, supporting its use in responsive robotic navigation.

Table 1. Numerical results of path planning before and after dynamic obstacle detection and re-planning of Robot 1.

Before re-planning			
Path Planner	**A***	**RRT***	**PPT**
Path Length (px)	103.37	117.48	**97.35**
Computational Time (s)	0.052	**0.023**	0.090
Number of Turns	5	2	**0**
After re-planning			
Path Planner	**A***	**RRT***	**PPT**
Path Length (px)	**133.64**	253.04	138.61
Computational Time (s)	**0.024**	0.033	0.880
Number of Turns	20	12	**2**

3.2.2. Robot 1: after re-planning

When another robot was detected in the path, the system triggered re-planning using a modified right-of-way rule. Under these conditions, the PPT again demonstrated strong performance by generating a safe and efficient path with only 2 turns, increasing the path length to 138.61 pixels. In contrast, both A* and RRT* experienced notable performance degradation. A* produced a less smooth path of 133.64 pixels with 20 turns, while RRT* generated the longest and most erratic path at 253.04 pixels with 12 turns, indicating difficulty in managing obstacle-dense environments. Despite the increased computational load, PPT maintained a reasonable runtime of 0.88 seconds, which is justifiable given its superior post-processing and smooth trajectory output.

3.2.3. Robot 2: before and after re-planning

As shown in Figure 12 and Table 2 for Robot 2, A* achieved the shortest path after re-planning, but A* produced the highest number of turns (31), indicating a less smooth trajectory. In contrast, the proposed PPT method maintained a good balance between path length and trajectory smoothness, achieving a low number of turns (only 2) with competitive planning time. This highlights the practicality of the PPT for real-world navigation where smoother paths are preferred. A video demonstration of the multi-robot path planning experiment is available at: (Video Link).

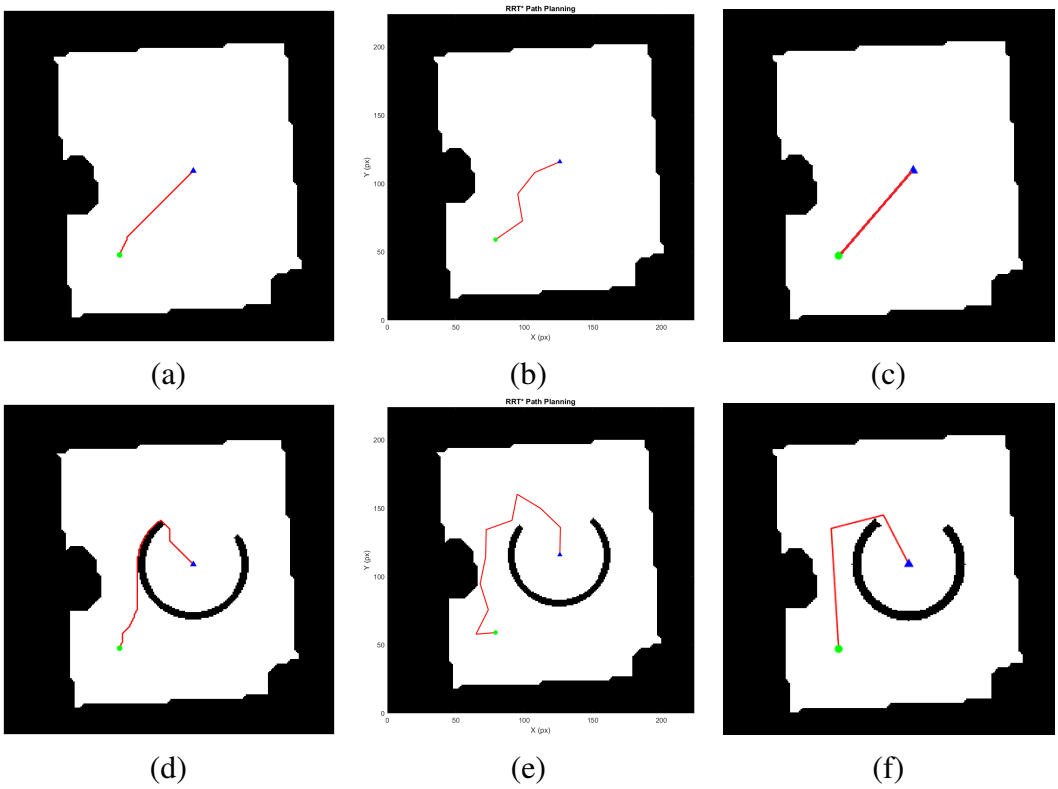


Figure 12. Re-planning behavior of Robot 2 under dynamic obstacle detection using the modified Right-of-Way rule by A*, RRT* and PPT algorithms from the start (blue) to the goal (green) via solution path (red). **(a)** A* – Initial path; **(b)** RRT* – Initial path; **(c)** PPT – Initial path; **(d)** A* – Replanned path; **(e)** RRT* – Replanned path; **(f)** PPT – Replanned path.

Table 2. Numerical results of path planning before and after dynamic obstacle detection and re-planning of Robot 2.

Before re-planning			
Path Planner	**A***	**RRT***	**PPT**
Path Length (px)	77.71	83.97	**73.88**
Computational Time (s)	0.568	**0.014**	0.092
Number of Turns	10	3	**0**
After re-planning			
Path Planner	**A***	**RRT***	**PPT**
Path Length (px)	**138.30**	193.79	154.41
Computational Time (s)	**0.568**	0.599	0.890
Number of Turns	31	9	**2**

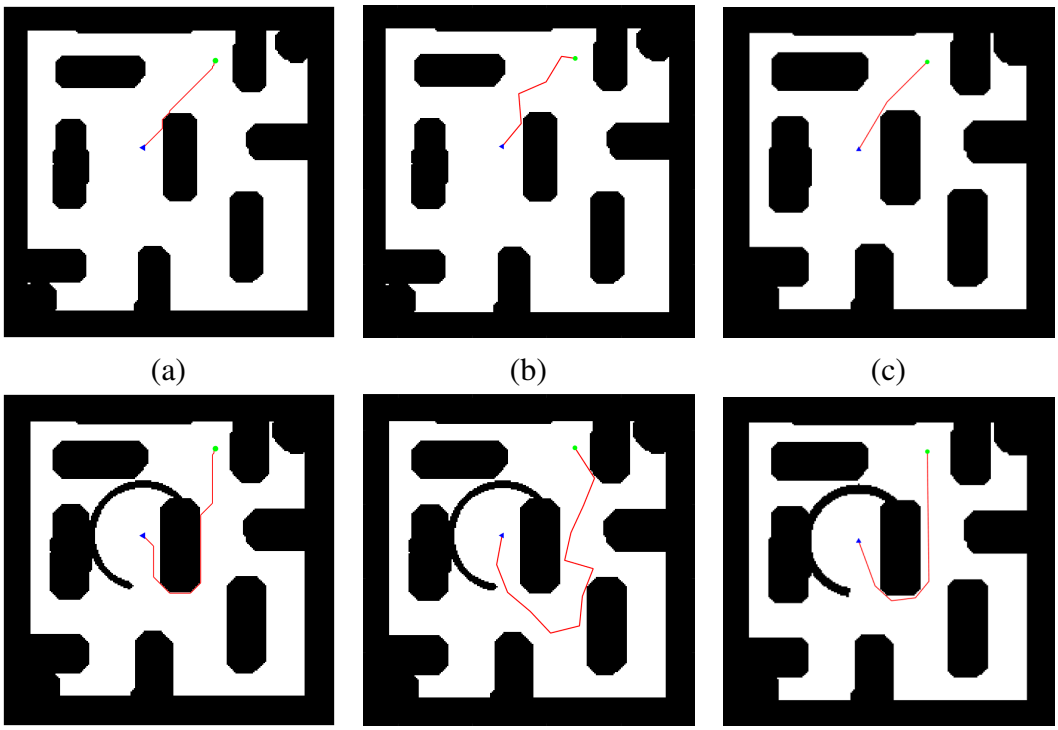
3.2.4. Simulation results on unknown static obstacle situation

In addition to real-world tests, the system was further validated in a Gazebo simulation environment. As shown in Table 3 and Figure 13, all planners, which are A*, RRT*, and PPT, successfully reached the goal both before and after static obstacle insertion. Prior to re-planning, PPT achieved the shortest and smoothest path (78.30 px with 1 turn). After re-planning, PPT again produced the shortest path (164.84 px) and fewest

turns (4), outperforming A* and RRT*, which generated longer and more segmented paths. Although RRT* maintained the lowest computation time (0.0706 s), its path was the longest and most irregular. These results confirm that PPT consistently yields efficient and smooth trajectories in both static and dynamic environments under the modified Right-of-Way rule.

Table 3. Numerical results of path planning before and after static obstacle detection and re-planning of robot in simulation on Gazebo.

Before re-planning			
Path Planner	**A***	**RRT***	**PPT**
Path Length (px)	79.3	89.24	**78.30**
Computational Time (s)	0.205	0.211	**0.085**
Number of Turns	8	4	**1**
After re-planning			
Path Planner	**A***	**RRT***	**PPT**
Path Length (px)	165.50	243.40	**164.84**
Computational Time (s)	0.211	**0.0706**	0.779
Number of Turns	7	9	**4**



(a) (b) (c)

(d) (e) (f)

Figure 13. Re-planning behavior of robot in simulation under static obstacle detection using the modified Right-of-Way rule by A*, RRT* and PPT algorithms from the start (blue) to the goal (green) via solution path (red). **(a)** A* – Initial path; **(b)** RRT* – Initial path; **(c)** PPT – Initial path; **(d)** A* – Replanned path; **(e)** RRT* – Replanned path; **(f)** PPT – Replanned path.

4. Discussion

In Table 1, which presents the experimental results from Robot 1, the PPT consistently achieved the shortest path length and zero turns before re-planning. After re-planning, it maintained a low number of turns (only 2) compared to A* (20) and RRT* (12), demonstrating its smoothness and adaptability in dynamic environments. Figure 11f visually confirms these results, showing that PPT produces a smooth trajectory even in the presence of new obstacles. Similarly, Table 2 presents results for Robot 2. Although A* generated the shortest path after re-planning, it required 31 turns, indicating a more zigzag route. In contrast, PPT achieved a slightly longer but smoother trajectory with only 2 turns. This aligns with the visual representation in Figure 12f, where PPT avoids unnecessary angular deviations.

In Table 3, simulation results further validate the effectiveness of PPT. Before re-planning, PPT showed the shortest path and lowest number of turns. After static obstacle detection and re-planning, although the path length increased due to obstacle avoidance, PPT maintained a balance between path efficiency (164.84 px) and trajectory smoothness (4 turns). This is evident in Figure 13f, where the path avoids obstacles with minimal curvature. These findings consistently highlight the strength of the PPT in producing smooth paths, particularly under re-planning scenarios with static obstacles and multiple robots.

5. Conclusion

In this study, a novel learning-based path planning framework was introduced for mobile robots operating in structured environments. The core of the system is the PPT, which was trained using expert trajectories generated by the IRRT*-RRMS algorithm. The proposed model demonstrated the ability to directly infer smooth and feasible paths from occupancy maps without the need for conventional SLAM during deployment. To address dynamic and multi-robot scenarios, a modified right-of-way rule was incorporated into the system to support decentralized re-planning based on virtual obstacle augmentation.

The proposed framework was evaluated through extensive simulations and real-world robot experiments using a dataset of 10,000 samples. The model converged after 429 training epochs with a best validation loss of 186, reflecting strong generalization. Comparative experiments with classical A* and RRT* planners confirmed the advantages of the proposed method, especially in terms of path smoothness and efficiency. Notably, the PPT consistently produced smoother trajectories with fewer turns both before and after encountering dynamic obstacles, while maintaining computational feasibility for real-time execution. In contrast, A* and RRT* generated more fragmented paths with increased turn counts after re-planning, particularly in obstacle-rich scenarios.

This work demonstrates how deep learning can complement classical planning by improving path prediction in both static and dynamic settings. The integration of the modified right-of-way rule allows for decentralized operation in multi-robot systems.

6. Future work

In future work, we plan to extend our current 2D PPT-based path planning framework to operate in more complex 3D environments. This is particularly important for indoor scenarios involving multi-floor buildings or environments with vertical structures such as stairs, ramps, or elevated platforms.

To achieve this, we propose using voxel-based map representations, such as OctoMap [19], which support hierarchical and memory-efficient storage of 3D occupancy data. The input to the transformer would be adapted from 2D occupancy grids to 3D voxel grids [20], while the decoder would be modified to output sequences of 3D waypoints (x, y, z).

This voxel-based extension will allow our framework to scale toward more realistic robotic deployments in industrial and service environments with layered or stacked navigation spaces.

To scale beyond two robots, we plan to implement a priority-based arbitration system in which each priority of the robot is dynamically determined (e.g., based on task urgency, distance to the goal, or predefined roles).

Additionally, we intend to integrate conflict-based search (CBS) or reservation table mechanisms into the re-planning pipeline to prevent livelocks and cyclic path conflicts.

Data availability statement

The datasets generated in this study are available on GitHub at https://github.com/technologyELTE/Path-Planning-Transformers-PPT.

Declaration of Generative AI and AI-assisted Technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

The authors gratefully acknowledge the support of the Komondor High Performance Computing infrastructure provided by KIFÜ (Kormányzati Informatikai Fejlesztési Ügynökség) at https://hpc.kifu.hu, which was used for training and evaluating the Path Planning Transformer model.

Authors' contribution

Conceptualization, A.L. and J.B.; methodology, A.L.; writing—original draft preparation, A.L.; revised manuscript preparation, J.B.; data curation, A.L. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

János Botzheim holds the position of Associate Editor for *Robot Learning* and has not peer reviewed or made any editorial decisions for this paper.

References

[1] Liu L, Wang X, Yang X, Liu H, Li J, *et al.* Path planning techniques for mobile robots: review and prospect. *Expert Syst. Appl.* 2023, 227:120254.

[2] Thrun S. *Probabilistic robotics*, 1st ed. Cambridge: MIT Press, 2005.

[3] Bouman A, Ott J, Kim SK, Chen K, Kochenderfer MJ, *et al.* Adaptive coverage path planning for efficient exploration of unknown environments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, October 23–27, 2023, pp. 11916–11923.

[4] Mohajerin N, Rohani M. Multi-step prediction of occupancy grid maps with recurrent neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, USA, June 16–20, 2019, pp. 10592–10600.

[5] Xu H, Chen J, Meng S, Wang Y, Chau LP. A survey on occupancy perception for autonomous driving: the information fusion perspective. *Inf. Fusion* 2025, 114:102671.

[6] Zhang Y, Zhang J, Wang Z, Xu J, Huang D. Vision-based 3D occupancy prediction in autonomous driving: a review and outlook. *arXiv* 2024, arXiv:2405.02595.

[7] Zhang T, Fan J, Zhou N, Gao Z. Highly self-adaptive path-planning method for unmanned ground vehicle based on transformer encoder feature extraction and incremental reinforcement learning. *Machines* 2024, 12(5):289.

[8] Önen Pandharipande A, Joseph G, Myers NJ. Occupancy grid mapping for automotive driving exploiting clustered sparsity. *IEEE Sens. J.* 2024, 24(7):9240–9250.

[9] Banfi J, Woo L, Campbell M. Is it worth to reason about uncertainty in occupancy grid maps during path planning? *arXiv* 2022, arXiv:2205.14251.

[10] Chaplot DS, Pathak D, Malik J. Differentiable spatial planning using transformers. *arXiv* 2021, arXiv:2112.01010.

[11] Wang H, Tan AH, Nejat G. NavFormer: a transformer architecture for robot target-driven navigation in unknown and dynamic environments. *IEEE Rob. Autom. Lett.* 2024, 9(8):6808–6815.

[12] Zhou C, Huang B, Fränti P. An advantage actor-critic algorithm for robotic motion planning in dense and dynamic scenarios. *arXiv* 2021, arXiv:2102.03138.

[13] Li Q, Gama F, Ribeiro A, Prorok A. Graph neural networks for decentralized multi-robot path planning. *arXiv* 2020, arXiv:1912.06095.

[14] Liu S, Chang P, Huang Z, Chakraborty N, Hong K, *et al.* Intention aware robot crowd navigation with attention-based interaction graph. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, May 29–June 2, 2023, pp. 12015–12021.

[15] Lonklang A, Botzheim J. Improved rapidly exploring random tree with bacterial mutation and node deletion for offline path planning of mobile robot. *Electronics* 2022, 11(9):1459.

[16] Lone MA, Khanday OM, Szilveszter K. Implementation guidelines for ethologically inspired fuzzy behaviour-based systems. *Infocommunications J.* 2024, 16(3):43–56.

[17] Lonklang A, Jara PAC, Xu H, Zhang W, Gyöngyössy NM, *et al.* LLM-based task planning for mobile robots using a novel SLAM method for materials distribution. In *2025 11th International Conference on Automation, Robotics, and Applications (ICARA)*, Zagreb, Croatia, February 12–14, 2025, pp. 235–239.

[18] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, *et al.* An image is worth 16×16 words: transformers for image recognition at scale. *arXiv* 2021, arXiv:2010.11929.

[19] Mao Y, Liu S. A-OctoMap: an adaptive OctoMap for online path planning. *arXiv* 2025, arXiv:2406.13910.

[20] Jang M, Kim D, Lee J. A voxel-based optimal path planning method for UAV navigation in smart cities. *ISPRS Int. J. Geo-Inf.* 2025, 14(12):457.