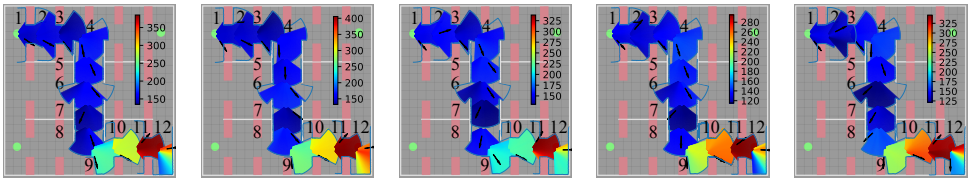
Deep Reinforcement learning with Multiple Unrelated Rewards for AGV Mapless Navigation: supplemental document

This document provides a detailed analysis of the experimental results for the Multi-Featured Policy Gradients (MFPG) algorithm as well as the baseline algorithms. Firstly, a horizontal comparison of MFPG with different reward weights is presented. Secondly, the algorithm is tested in a more complex environment, and trajectory data are visualized, including both a failure case and a successful trajectory. Lastly, the preference of the algorithms with respect to different actions is visualized using other given trajectories to identify the algorithms' preferences under certain circumstances.

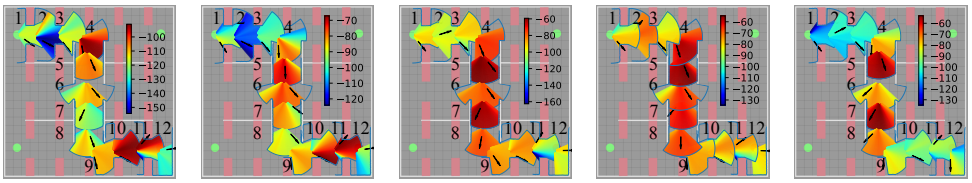
1. HORIZONTAL COMPARISON OF GRADIENT WEIGHTS.

In this section, we compare the preference of the MFPG with respect to different gradient weights Φ. The details of the picture are shown in Figure S1 and S2.



(a) Q (Nav) with gradient weight Φ= [5, 1].
(b) Q (Nav) with gradient weight Φ= [3, 1].
(c) Q (Nav) with gradient weight $\Phi = [1, 1]$.
(d) Q (Nav) with gradient weight $\Phi = [1, 3]$.
(e) Q (Nav) with gradient weight Φ= [1, 5].

Fig. S1. Visulization of the changes of Q (Nav) values with respect to different gradient weights.



(a) Q (CA) with gradient weight Φ= [5, 1].
(b) Q (CA) with gradient weight Φ= [3, 1].
(c) Q (CA) with gradient weight $\Phi = [1, 1]$.
(d) Q (CA) with gradient weight $\Phi = [1, 3]$.
(e) Q (CA) with gradient weight Φ= [1, 5].

Fig. S2. Visualization of the changes of Q (CA) values with respect to different gradient weights.

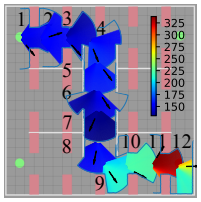
Obviously, the selection of actions by the policy is not simply according to the distribution of Q^{CA} or Q^{Nav}. Upon analyzing the changes in gradient weights, we observe that an increase in the proportion of the collision avoidance Q network gradients causes a shift in the policy's focus from target navigation to collision avoidance. At the initial stage, especially waypoints 1, 2 and 3, the policy did not choose actions that can maximize the collision avoidance Q values in Figure S2a, S2b and S2c. As the gradient weight for collision avoidance increases, the policy becomes more inclined to select actions that would avoid collisions.

2. PERFORMANCE OF THE ALGORITHMS IN A MORE COMPLEX ENVIRONMENT.

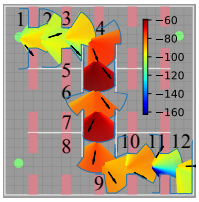
In Section V.C, we carry out a trajectory preference analysis in a complex environment for each algorithm. Here, we present the analyses of trajectories in the warehouse as shown in Figure S3.

For the MFPG-based algorithms, the navigation Q distribution (as shown in Figure S3a and S3c) remains at a low level before Waypoint 8, and then increases to the maximum value. Additionally, a sharp jump of the navigation Q value is observed from Waypoint 10 to 11 in Figure S3a, S3c, S3f and S3g. We believe this happens because the laser scan detects nothing between the target and agent, which increased their confidence of corresponding action. In Waypoint 12 of the navigation reward, the Q value distribution shows the trend for separate actions, when the agent approaches the target location. Since the left side of the vehicle is closer to the target, actions that approach to the left are assigned a high Q value, which is opposite to the right side. On the other hand, the collision avoidance state-action values shows an homogeneous trend during the navigation tasks. First of all, it is apparent that the agents trained under the social norm collision avoidance reward obeys the preset right-hand rule. Specifically, for Waypoint 1, in Figure S3b, S3d, S3e, S3g, the distribution of the Q value indicates the policy prefers more to turning right than turning left. However, similar situations that can be observed Waypoint 11 of Figures S3b, S3d, S3e, produce different actions. It happens because 1) the vehicle is closer to obstacles on the right than that on the left; 2) the laser-sensor reading reveals that there is no obstacle-free area on the right-hand side. The two factors made the policy more pessimistic on actions of turning right. Furthermore, the highest Q values for collision avoidance are located in Waypoints 5 and 7 in Figures S3b and S3d, which show its robustness in ensuring safety under the corresponding states.

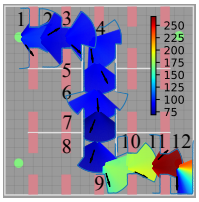
With the other three algorithms ofTD3 (Figure S3e), DDPG (Figure S3f) and SAC (Figure S3g), only one Q value is available for each of them. Therefore, the navigation behaviours, as well as the collision avoidance behaviours, would be shown on the same Q value distributions. The TD3 (Figure S3e) presents the most limited range of Q values among all the algorithms, ranged from -20 to 20. The maximum Q value for TD3 appears at Waypoint 4, while the minimum value appears at the target location (Waypoint 12). It is also worth noting that TD3 is the only algorithm with Q values that range evenly on both sides of 0. With DDPG (Figure S3f), it can be observed that the change of Q value increases along the path towards the target goal. As the robot moves closer to the target, actions are assigned higher Q values. Additionally, it is noticeable that the Q value distribution is smoother compared to the other algorithms, indicating that the DDPG algorithm is less sensitive to changes in the robot's position and laser scan information.



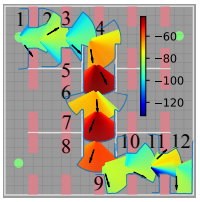
(a) Q (Nav) value of MFPG.



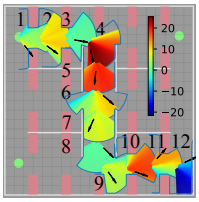
(b) Q (CA) value of MFPG.



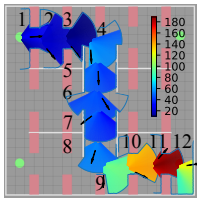
(c) Q (Nav) value of Ada-MFPG.



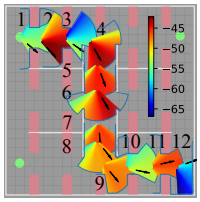
(d) Q (CA) value of Ada-MFPG.



(e) Q value of TD3.



(f) Q value of DDPG.



(g) Q value of SAC.

Fig. S3. Comparison of state-action value distribution for different algorithms based on a given trajectory.

3. EXPERIMENTS ON REAL ROBOT

In this section, we conduct further experiments with real robots in a lab environment to verify the viability of our algorithm in the real world. The experimental environment comprises a long corridor and two rooms connected by two narrow doors, as displayed in Figure S4. In the map, we selected three positions as the candidate start and target positions, denoted by A, B, and C. Besides, two doors within the map are also shown in Figure S4. The maximum traveling distance is 15m from Position A to Position C.

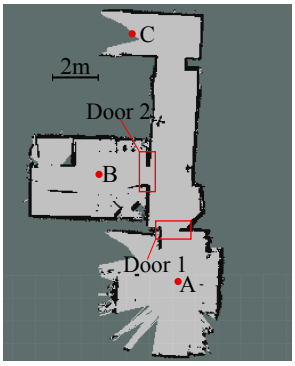
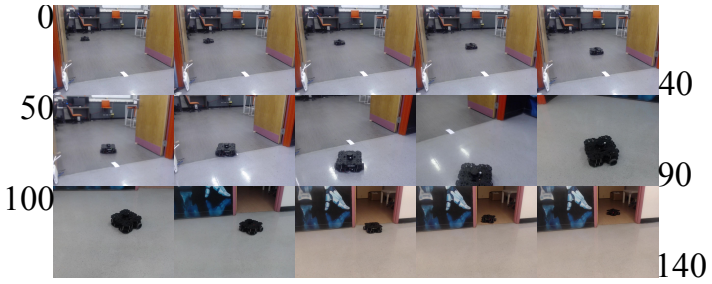


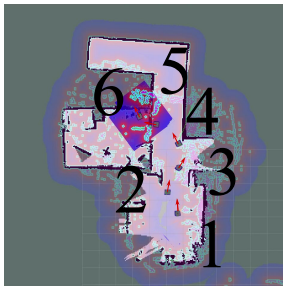
Fig. S4. The map of the experiment on real robot.

The robot's trajectories and their corresponding captured images are shown in Figures S5, S6, and S7. All the settings and control methods are identical to those described in Section V.D.

Figure S5 depicts the robot's movement from position A to position B, crossing two narrow doors. The robot began at Step 1 and detected an obstacle on the right-hand side. Therefore, it decided to move towards the right front. Upon reaching Step 2, the robot detected a dangerous situation from the laser scan on its left front and reacted by turning right. From Step 3 to Step 4, the robot attempted to pass through the first narrow door and turned toward its target position. However, due to the slippery floor, the odometry errors caused a discrepancy between the real robot trajectory and the poses estimated by Lidar-based SLAM (Simultaneous Localization and Mapping), resulting in inaccurate localization. After Step 4, the robot headed towards the target location and arrived at Step 6 after passing through the second door.



(a) Trajectory view from camera..

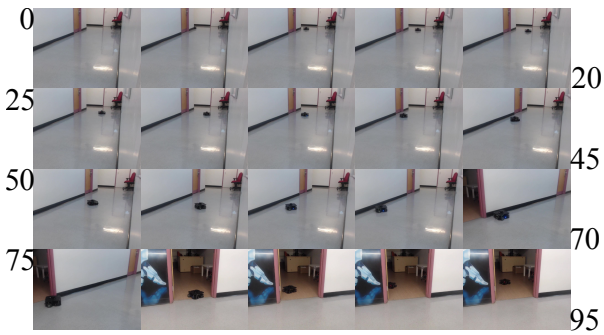


(b) Trajectory view in rviz.

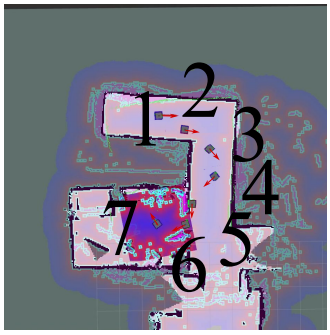
Fig. S5. Trajectory 1 of real robot navigation in corridor.

Figure S6 illustrates the robot's trajectory from position C to position B. The robot had to steer around a block and cross a narrow door to reach the target location. In Step 1, the robot began driving towards the right front of its current position. As the robot approached the corner, it attempted to move straight ahead to avoid colliding with the wall. After passing the corner, the robot remained stationary at Step 3 before being attracted to the target at Step 4. However, its action towards the target was blocked by the wall. At Step 5, the robot attempted to navigate along the wall while maintaining a small gap. Eventually, the robot reached the door and began turning right to approach the target location.

In Figure S7, the robot navigates from position A to position C through the long corridor. Beginning from Step 1, the robot attempted to move through the middle of the narrow door. In Step 4, the robot's heading is misaligned with the target orientation, and the policy produced subsequent actions to adjust its heading in Step 5 accordingly. After Step 6, the agent followed the wall on its left and finally approached the corner at Step 9. Finally, after turning left at the corner, the agent reached its target at Step 11.

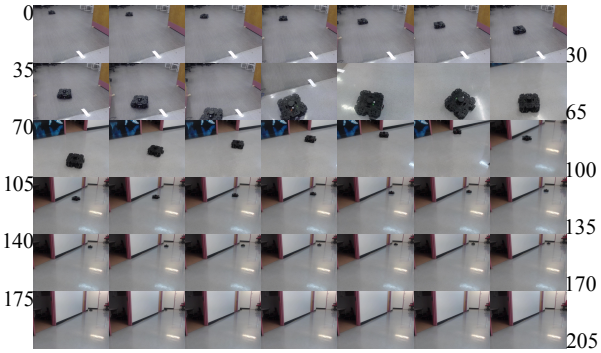


(a) Trajectory view from camera.

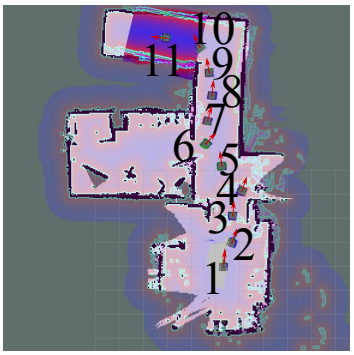


(b) Trajectory view in rviz.

Fig. S6. Trajectory 2 of real robot navigation in corridor.



(a) Trajectory view from camera.



(b) Trajectory view in rviz.

Fig. S7. Trajectory 3 of real robot navigation in corridor.

4. SUMMARY

In the supplementary material, we show additional comparative studies of the algorithms with different reward weights or gradient weights. Table S1 and Table S2 illustrate the state-action value distributions of the different combinations of reward/gradient weights. Extra experimental analyses of real robot performance are also included, demonstrating the viability of the proposed algorithm to be deployed on real robots.

Table S1. Comparison of all algorithms with different reward/gradient weights in a complex environment

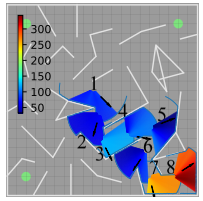
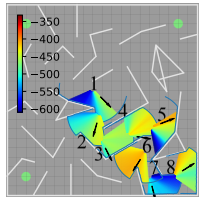
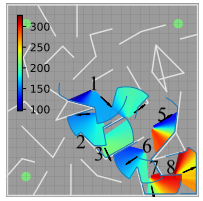
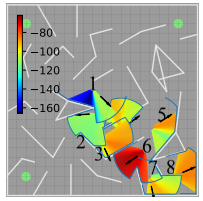
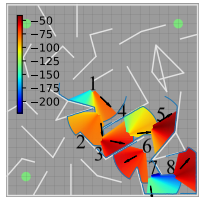
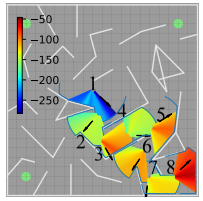
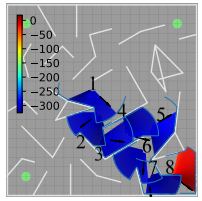
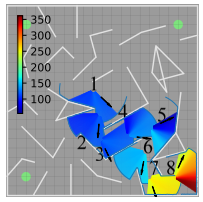
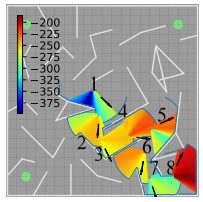
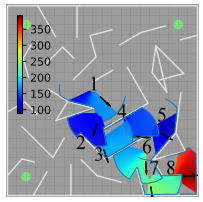
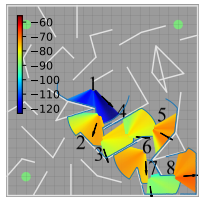
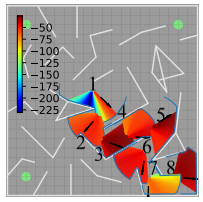
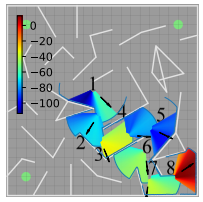
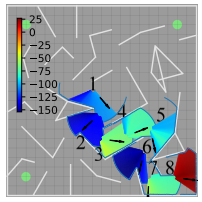
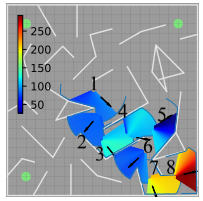
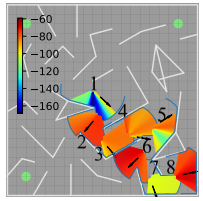
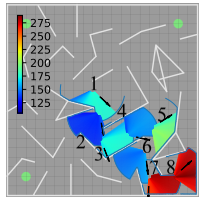
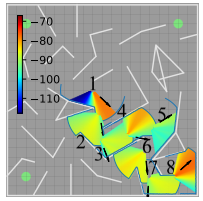
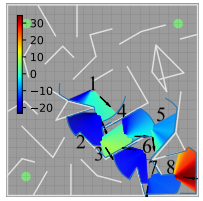
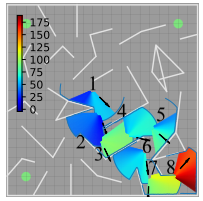
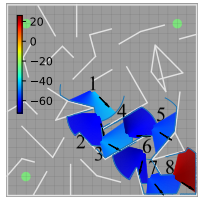
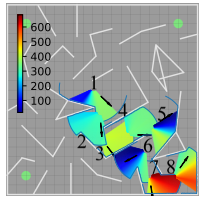
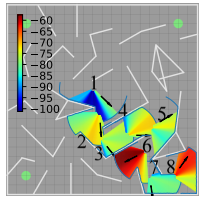
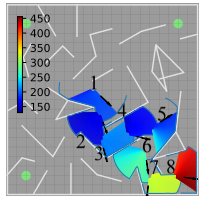
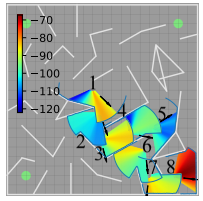
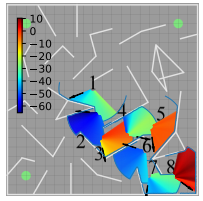
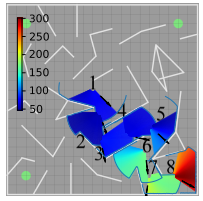
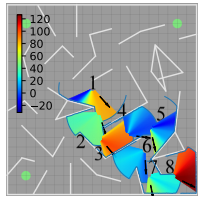
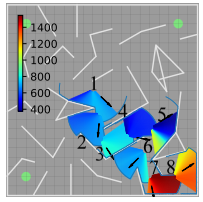
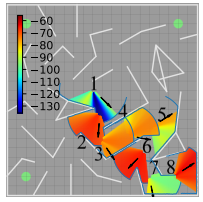
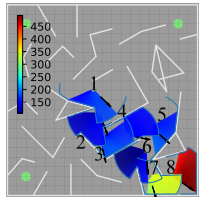
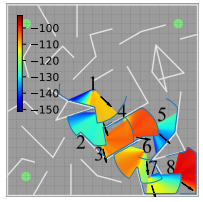
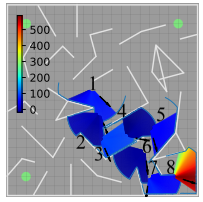
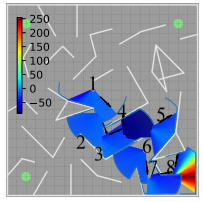
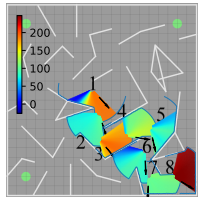
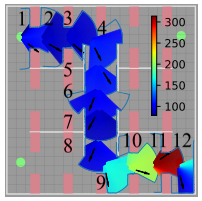
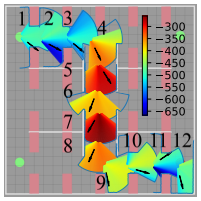
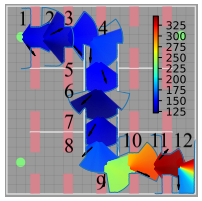
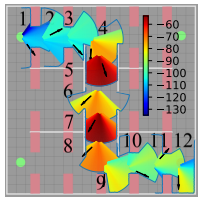
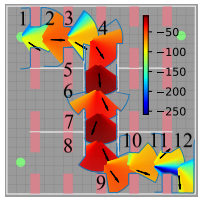
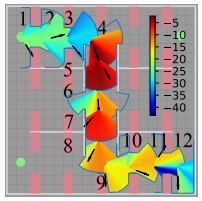
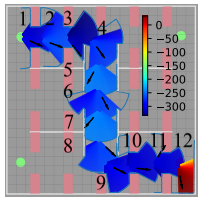
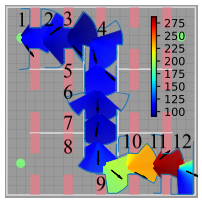
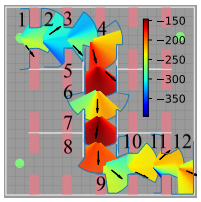
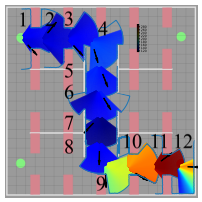
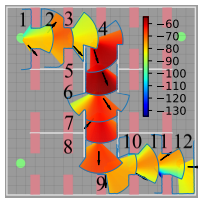
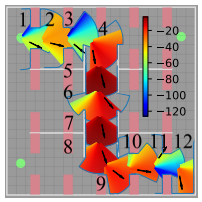
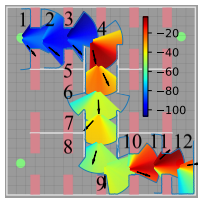
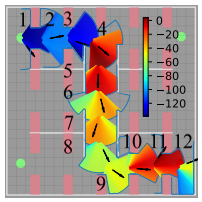
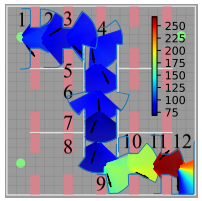
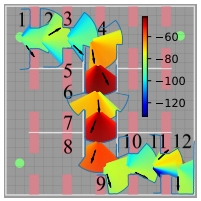
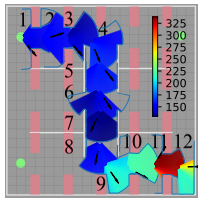
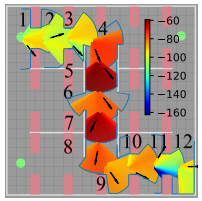
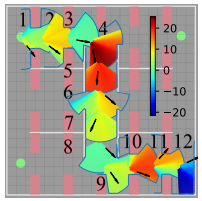
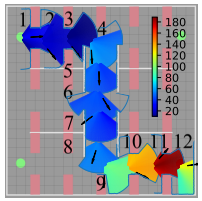
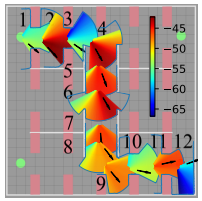
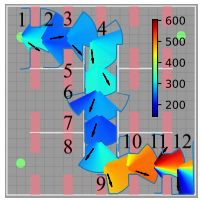
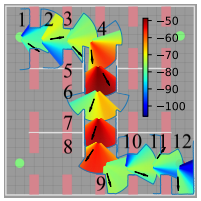
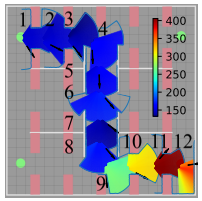
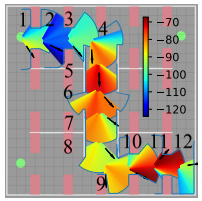
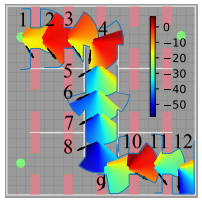
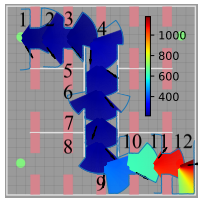
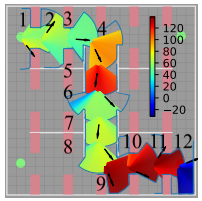
reward/gradient weight	Ada-MFPG (Nav)	Ada-MFPG (CA)	MFPG (Nav)	MFPG (CA)	TD3	DDPG	SAC
(1,5)							
(1, 3)							
(1, 1)							
(3, 1)							
(5, 1)							

Table S2. Comparison of all algorithms with different reward/gradient weights in a warehouse environment

reward/gradient weight	Ada-MFPG (Nav)	Ada-MFPG (CA)	MFPG (Nav)	MFPG (CA)	TD3	DDPG	SAC
(1, 5)							
(1, 3)							
(1, 1)							
(3, 1)							
(5, 1)