2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Crowd-Aware Robot Navigation with Switching Between Learning-Based and Rule-Based Methods Using Normalizing Flows

Kohei Matsumoto[1†], Yuki Hyodo[2†], and Ryo Kurazume[1]

***Abstract*— Mobile robot navigation in crowded environments with pedestrians is a crucial challenge in realizing service robots that can assist people in their daily lives. Navigation methods for mobile robots in environments employing deep reinforcement learning have been extensively studied. However, addressing such unexpected situations is a significant challenge. This study presents an approach that discerns whether a situation has been supposed to utilize a normalizing flow and dynamically switches between learning- and rule-based methods. Specifically, the proposed method achieves a higher success rate than employing only a learning-based approach and reaches the destination faster than employing only a rule-based approach in unexpected situations. Experiments are conducted to validate the performance enhancement achieved with the proposed switching method in both simulated and real-world settings.**

I. Introduction

Mobile robot navigation is essential for service robots to operate in human-inhabited environments. Robots must move smoothly, even in environments crowded by pedestrians. Navigation methods for crowded environments have been extensively studied, particularly in recent years, with significant results achieved with deep reinforcement learning (DRL). DRL-based methods have significant potential because they can outperform rule-based methods through learning, particularly mobile robot navigation in crowded environments, including predicting pedestrian behavior, which can be addressed through learning. However, learning-based methods are difficult to handle when faced with unlearned situations. In addition, it is difficult to collect data for all possible situations or create a simulation environment for mobile robot navigation in crowded environments. Mobile robots are expected to be applied in various environments, and pedestrian behavior is expected to be influenced and changed by various environmental factors. This is difficult to operate without the possibility of encountering situations that the robot may not assume.

This study attempts to make the robot respond to such problems by judging the situation and taking learning-based actions if it can respond to the situation based on learning. Otherwise, it will take safe rule-based actions to cope with an environment not assumed during training. In the example

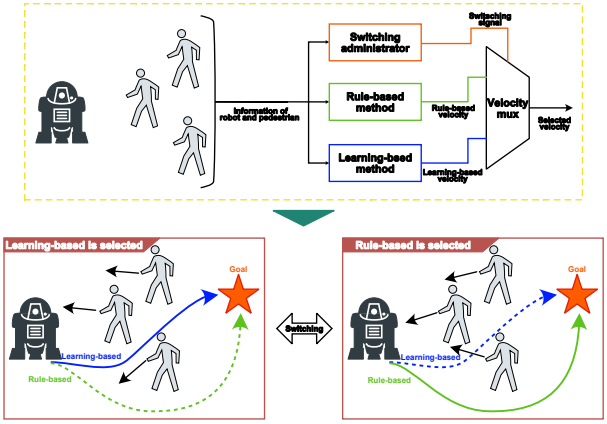


Fig. 1. Conceptual diagram of the proposed method. The schematic configuration of proposed method is illustrated in the yellow dashed rectangular window. Output speeds from rule-based and learning-based methods are input to speed multiplexer, and either one is selected by signal from switching administrator. Brown rectangular windows indicate status of rule-based and learning-based selections. Solid colored lines indicate trajectory of selection. In left window, learning-based action is selected because it can reach destination faster, and in right window, rule-based action is selected because learning-based action may cause collision.

of mobile robot navigation in a crowded environment, the robot usually moves while avoiding pedestrians, adopting a learning-based method, which is more efficient than a rule-based method, and acts in a rule-based manner to avoid a collision if a collision is likely to occur in a learning-based manner. Following this idea, this research attempts to realize a method that employs both learning-based and rule-based methods in environments that were not envisioned during training, has a lower collision rate than if it acted only on a learning-based method, and reaches its destination more quickly than if it acted only on a rule-based method. Specifically, this study achieves this goal by switching between a learning-based method trained by offline reinforcement learning from a limited amount of data and a rule-based method with likelihood estimation through a graph normalizing flow. The main contributions of this study are as follows.

- Calculation method of switching score applying graph normalization flow to mobile robot navigation in crowded environments;
- mobile robot navigation method by switching between a learning-based method and a rule-based method in crowded environments;
- demonstrating that the proposed switching method improves performance in unexpected environments and

*This work was partially supported by JSPS KAKENHI Grant Number JP20H00230 and a collaborative research with KYUDENKO Corporation

[1]Kohei Matsumoto and Ryo Kurazume are with the Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan {matsumoto, kurazume}@ait.kyushu-u.ac.jp

[2]Yuki Hyodo is with the School of Engineering, Kyushu University, Fukuoka 819-0395, Japan hyodo@irvs.ait.kyushu-u.ac.jp

†These two authors contributed equally to this work.

the necessity of switching at appropriate timings by comparing proposed switching method with the method of switching at random by simulation experiments;
- application of the proposed method to an actual robot system and conducted experiments in a real-world environment.

II. Related Work

DRL-based mobile robot navigation methods have been actively studied in crowded environments. Chen *et al.* [1] introduced a value learning model of DLR for decentralized non-communicating multiagent collision avoidance. Chen *et al.* [2] proposed DRL-based method with reflecting social norms. Everett *et al.* [3] proposed a method that employs long-term memory (LSTM) to handle situations where the number of pedestrians in the environment changes. Chen *et al.* [4] proposed a pooling module to learn the relationships among the attention mechanism, changes in human behavior, and surrounding pedestrians, and proposed a method to learn robot behavior, including human–robot and human–human interactions. Chen *et al.* [5] used a graph convolutional network based on human gaze data to predict human attention to different agents in a crowd and applied it to a graph-based reinforcement learning architecture. Chen *et al.* [6] proposed an excellent navigation method that adopts a graph convolutional network and an attention mechanism based on the human gaze. Chen *et al.* [7] proposed a DRL method with learning relation between robot and each pedestrian and between pedestrians using a graph neural network to achieve efficient navigation. Liu [8] proposed a DRL method that focused on robot navigation in crowded environments with static obstacles. Zhang *et al.* [9] proposed a DRL method method based on relational graphs learning that considers a continuous action space. Yao *et al.* [10] proposed a method that applies two maps, a sensor map and a pedestrian map, as inputs to a neural network. Matsumoto *et al.* [11] proposed a method based on a predictive state representation to consider the effects of robot actions on pedestrians. Yang *et al.* Matsuzaki *et al.* [12] developed a method for estimating parameters of social force model from real pedestrian data to construct simulations that more closely resemble actual pedestrians, and have proposed a more efficient learning method using distributed deep reinforcement learning. Katyal *et al.* [13] proposed group-aware navigation policies based on dynamic group formation using DRL. Wang *et al.* [14] proposed hybrid experiential learning method consisting of exploration by curiosity and expert demonstration. Yang *et al.* [15] proposed a method utilizing a spatial-temporal state transformer to extract both the spatial and temporal features of human–robot and human–human interactions. Although these methods have significantly improved the performance of learning-based methods in crowd-aware mobile-robot navigation, they do not address handling situations that are not anticipated during training. Liu *et al.* [16] proposed a novel recurrent graph neural network with attention mechanisms to capture heterogeneous interactions between agents, and incorporated it into a model-free RL framework.

Apart from DRL-based methods, researchers have investigated various other approaches those based on partially observable Markov decision processes (POMDP) [17], [18], generative adversarial imitation learning (GAIL) [19], and generative adversarial networks (GAN) [20]. The utilization of a large-scale dataset [21] and a simulation environment based on ROS2 [22] were also proposed.

Normalizing flows (NFs) are a family of generative models with sequences of invertible transformations. Like other generative models, NFs have been actively studied for applications in image generation and applied to SLAM and manipulation in the field of robotics [23], [24]. Rudolph *et al.* [25] proposed an anomaly-detection method employing NFs. Inspired by this study, we propose a method that switches between learning-based and rule-based methods based on likelihood calculations employing NFs that can handle graph structures [26].

The most relevant work is that of Wu *et al.* [27], however, it does not address the timing of switching, and no real-world experiments were conducted.

III. Preliminaries

A. Problem Formulation

This study deals with the problem of a robot reaching a goal in the X-Y plane while avoiding collisions with surrounding pedestrians. The observable information includes the robot's position $\boldsymbol{p}^r = \left[p_x^r, p_y^r\right]$, velocity $\boldsymbol{v}^r = \left[v_x^r, v_y^r\right]$, orientation θ and goal position $\boldsymbol{p}^g = [g_x, g_y]$, as well as the position $\boldsymbol{p}^h = \left[p_x^n, p_y^n\right]$ and velocity $\boldsymbol{v}^n = \left[v_x^n, v_y^n\right]$ of surrounding pedestrians. In addition, we assume holonomic kinematics for the robot, which implies that the robot can move in any direction, and the action information is the velocity value in the X- and Y-directions, $\boldsymbol{a} = \left[v_x^c, v_y^c\right]$. Utilizing this information, the robot's state obtains its position from the goal, velocity, and orientation information, $\boldsymbol{s}^r = [|\boldsymbol{p}^g - \boldsymbol{p}^r|, \boldsymbol{v}^r, \theta]$. By contrast, the position and velocity information of each pedestrian is converted to a robot-centered coordinate system and adopted as a state $\boldsymbol{s}^n = \left[{}^{rc}\boldsymbol{p}^n, {}^{rc}\boldsymbol{v}^n\right]$. Furthermore, the set of states of pedestrians in the environment is described as $\boldsymbol{s}^h = \left[\boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^n\right]$.

The robot obtained rewards after acting. R_t denotes the reward function at time t. This function follows the equation below.

$$R_t = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \boldsymbol{p}_t^r = \boldsymbol{p}_t^g \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where d_t is the minimum separation distance between the robot and the pedestrians.

B. Normalizing Flows

NFs are generative models that transform latent variables generated from a normal distribution into samples within a target distribution through reversible transformations. Compared to other generative models, NFs are characterized by their ability to calculate exact likelihoods and transform

data into latent variables through reversible transformations. Fig. 2 presents a conceptual diagram of NFs. As depicted in

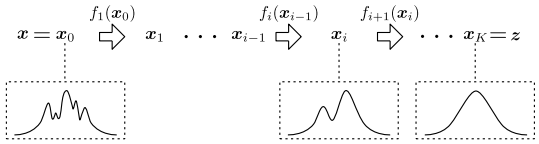


Fig. 2. Conceptual diagram of normalizing flows.

the figure, the NFs transform the target sample $\boldsymbol{x}$ into the latent variable $\boldsymbol{z}$ by recursively applying an invertible transformation f. The inverse transformation f^{-1} is recursively applied to the latent variable $\boldsymbol{z}$ during its generation. This transformation follows the following equation:

$$\begin{cases} \boldsymbol{x}_{n+1}^{1:d} & = \boldsymbol{x}_n^{1:d} \\ \boldsymbol{x}_{n+1}^{d+1:D} & = \boldsymbol{x}_n^{d+1:D} \odot \exp(f^s(\boldsymbol{x}_n^{1:d})) + f^t(\boldsymbol{x}_n^{1:d}) \end{cases}, \quad (2)$$

where f^s and f^t are the scale and translation functions, respectively, and $\odot$ is the elementwise product.

$$y = b \odot x + (1-b) \odot (x \odot \exp(s(b \odot x)) + t(b \odot x)). \quad (3)$$

For the target sample $\boldsymbol{x}$, the log-likelihood can be calculated as:

$$\log p_X(\boldsymbol{x}) = \log p_Z(\boldsymbol{z}) + \sum_{i=1}^{K} \log \left| \det \frac{df_i(\boldsymbol{x}_{i-1})}{d\boldsymbol{x}_{i-1}} \right|, \quad (4)$$

and the training was performed by minimizing the negative log-likelihood. This is expected to result in comparatively lower likelihoods for data not assumed during training. In this study, we aim to leverage this characteristic to determine the necessity of switching.

IV. Approach

A conceptual diagram of the proposed method is presented in Fig. 1. Command velocities were generated from information from the robot and pedestrians employing rule- and learning-based methods, respectively. These velocities are then switched by switching the signals sent by the switching administrator. ORCA[28] was adopted as the rule-based method, and the policy learned by advantage weighted actor-critic (AWAC) in an offline setting was employed as the learning-based method. A normalization flow for the graph structure is adopted as the basis for the switching administrator.

A. Learning-based Method

The architecture of the learning-based model is illustrated in Fig. 3. This study adopts an actor–critic structure and employs a probabilistic policy as actor π_θ. In addition, the structure has two Q-functions, Q_{ϕ^1} and Q_{ϕ^2} for the min-double-Q trick. The trained actor outputs the mean and variance of the normal distributions. The actor and critic each have a graph neural network to learn the relationship between the robot and the pedestrian. The structure of the graph neural network is an embedded Gaussian structure, which has been

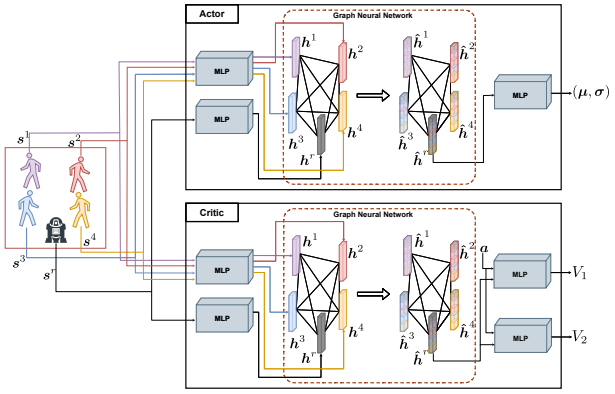


Fig. 3. Architecture of actor-critic part of proposed method. $\boldsymbol{s}^r$ and $\boldsymbol{s}^n$ indicate states of each robot and pedestrian. $\boldsymbol{h}^n$ indicates features of each robot and pedestrian. $\hat{\boldsymbol{h}}^n$ indicates features after GNN is applied.

applied in previous studies [7], [9]. This network is trained with AWAC by employing a clipping advantage [9]. The actor was trained with the following equation:

$$L_{\text{actor}} = -\log \pi_\theta(\boldsymbol{s}, \boldsymbol{a}) \exp\left(\frac{1}{\lambda} W\right), \quad (5)$$

where λ is the hyperparameter. The weight value W is calculated with the following equation:

$$W = \max\left(0, \min_{i=1,2} Q_{\phi^i}(\boldsymbol{s}, \pi_\theta(\cdot \mid \boldsymbol{s})) - \min_{i=1,2} Q_{\phi^i}(\boldsymbol{s}, \boldsymbol{a})\right). \quad (6)$$

The learning-based method is trained based on offline RL, which means it does not involve additional data collection through exploration. The training algorithm is presented in Algorithm 1.

B. Switching Administrator

Fig. 4 illustrates the switching administrator architecture. The switching administrator is based on normalizing flows, where the pedestrian state is encoded by the MLP and then input into the normalizing flow, where the likelihood is computed. The structure of the normalizing flow is a graph-normalizing flow [26]. Graph-normalizing flow is based on a real-valued non-volume preserving network [29] and is realized by replacing the nonlinear functions in the affine coupling layer with graph neural networks. The Graph Attention Network v2 (GATv2) [30] was adopted as the graph neural network. We employ partition utilizing a binary mask. Thus, the transformation of the flow block in the switching administrator is described as:

$$\begin{aligned} \bar{\boldsymbol{h}}^n &= \boldsymbol{b} \odot \boldsymbol{h}^n + (1-\boldsymbol{b}) \\ &\quad \odot \left(\boldsymbol{h}^n \odot \exp(f_g^s(\boldsymbol{b} \odot \boldsymbol{h}^n)) + f_g^t(\boldsymbol{b} \odot \boldsymbol{h}^n)\right) \quad (7) \\ \hat{\boldsymbol{h}}^n &= \tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n + (1-\tilde{\boldsymbol{b}}) \\ &\quad \odot \left(\bar{\boldsymbol{h}}^n \odot \exp(f_g^s(\tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n)) + f_g^t(\tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n)\right), \quad (8) \end{aligned}$$

where $\boldsymbol{h}^n \in \mathbb{R}^d$ is a feature value encorded by MLP. In addition, $\boldsymbol{b}$ is a binary mask and $\tilde{\boldsymbol{b}}$ is an inversion of $\boldsymbol{b}$. f_g^s and f_g^t are the scale and translation functions, respectively, based on GATv2.

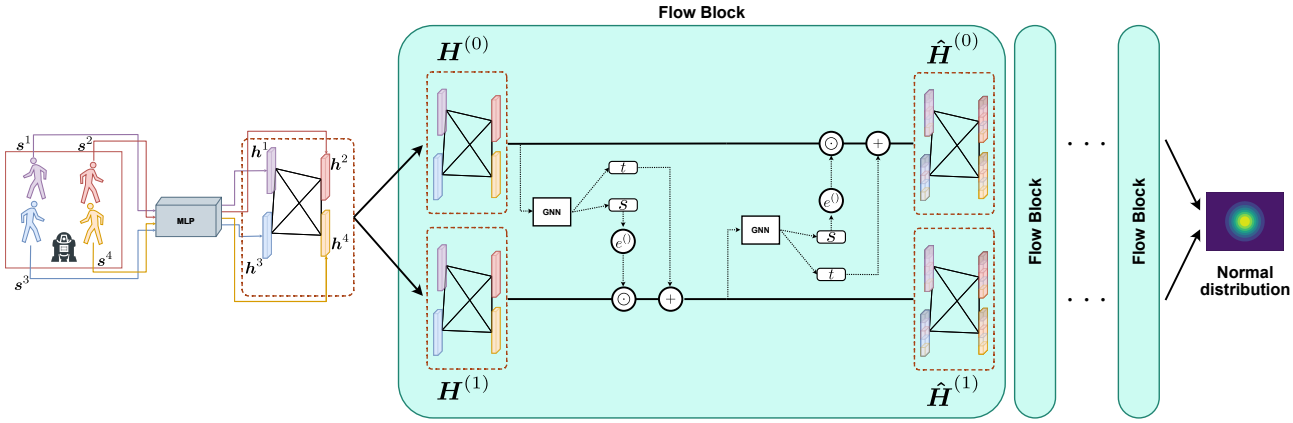


Fig. 4. Architecture of switching administrator. $\boldsymbol{s}^n$ indicates state of each pedestrian. $\boldsymbol{h}^n$ indicates features of each robot and pedestrian. $\boldsymbol{H}$ indicates a set of features of each pedestrian. $\hat{\boldsymbol{H}}$ is a set of features of each pedestrian after applying a flow block.

Algorithm 1: Training algorithm for learning-based method

Initialize actor π_θ and Q-functions Q_{ϕ^1} and Q_{ϕ^2}
Set the parameter values of the target critics $Q_{\phi',1}$ and $Q_{\phi',2}$ equal to the main critics
for $i = 1$ **to** E **do**
  Sample a batch $\mathcal{B} = \{(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')\}$ from the dataset $\mathcal{D}$
  Sample actions for computing targets $\hat{\boldsymbol{a}}' \sim \pi_\theta(\cdot \mid \boldsymbol{s}')$
  Targets for the Q-functions were calculated. $y(r, \boldsymbol{s}') = r + \gamma\left(\min_{i=1,2} Q_{\phi',i}(\boldsymbol{s}', \hat{\boldsymbol{a}}')\right)$
  Update Q-functions by minimizing $L_{\text{critic}} = \frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in\mathcal{B}} \left(Q_{\phi^i}(\boldsymbol{s}, \boldsymbol{a}) - y(r, \boldsymbol{s}')\right)^2$ for $i = 1, 2$
  Sample actions for computing the weight values $\hat{\boldsymbol{a}} \sim \pi_\theta(\cdot \mid \boldsymbol{s})$
  Calculate the regression weights $W = \max\left(0, \min_{i=1,2} Q_{\phi^i}(\boldsymbol{s}, \boldsymbol{a}) - \min_{i=1,2} Q_{\phi^i}(\boldsymbol{s}, \hat{\boldsymbol{a}})\right)$
  Update π_θ by minimizing $L_{\text{actor}} = -\frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}')\in\mathcal{B}} \log \pi_\theta(\boldsymbol{s}, \boldsymbol{a}) \exp\left(\frac{1}{\lambda} W\right)$
  Update the target critic by polyak averaging $\phi'^{,i} \leftarrow \rho\phi'^{,i} + (1-\rho)\phi_i$ for $i = 1, 2$
end

C. Switching Process

Switching between rule-based and learning-based methods is performed when the likelihood calculated by the learning-switching manager falls below a threshold value, θ_{th}. Actions at each time point were generated with the following equation:

$$\boldsymbol{a}_t = \begin{cases} \boldsymbol{a}_t^r & \text{for } \tau(\boldsymbol{s}_t^h) > \theta_{th} \\ \boldsymbol{a}_t^l & \text{for } \tau(\boldsymbol{s}_t^h) \leq \theta_{th} \end{cases}, \tag{9}$$

where $\boldsymbol{a}_t^r$ and $\boldsymbol{a}_t^l$ denote the actions generated by the rule-based and learning-based methods at time t, respectively. The switching score $\tau(\boldsymbol{s}^h)$ is calculated as:

$$\tau(\boldsymbol{s}^h) = -\log p_Z\left(f_{\text{sa}}(\boldsymbol{s}^h)\right). \tag{10}$$

Although determining the threshold remains a major open challenge, we utilized the average value of the likelihoods calculated for the training data by a trained switching administrator, with the aim that the threshold can be determined independently of the amount of training. The flow of mobile robot navigation adopting the proposed method is presented in Algorithm 2.

Algorithm 2: Navigation with proposed method

Input: Trained policy π_θ and trained switching administrator f_{sa}
while $\boldsymbol{p}_t^r \neq \boldsymbol{p}_t^g$ **do**
  Observe the states $\boldsymbol{s}_t = \left[\boldsymbol{s}_t^r, \boldsymbol{s}_t^h\right]$ at time t
  Obtain the $\boldsymbol{a}_t^r$ and $\boldsymbol{a}_t^l$ employing ORCA and π_θ with $\boldsymbol{s}_t$
  Obtain switching score employing $\tau(\boldsymbol{s}_t^h) = -\log p_Z\left(f_{\text{sa}}(\boldsymbol{s}_t^h)\right)$
  Select action by $\boldsymbol{a}_t = \begin{cases} \boldsymbol{a}_t^r & \text{for } \tau(\boldsymbol{s}_t^h) > \theta_{th} \\ \boldsymbol{a}_t^l & \text{for } \tau(\boldsymbol{s}_t^h) \leq \theta_{th} \end{cases}$
  Execute the action $\boldsymbol{a}_t$
end

V. Experiments

Evaluation experiments were conducted in a simulation environment to verify the effectiveness of the proposed method.

A. Simulation Environment

We employed the square- and circle-crossing scenarios in the CrowdNav environment adopted in some related studies [4], [7]. Trajectories for example cases of these two scenarios are illustrated in Fig. 5. The square-crossing scenario was employed for training, and the circle-crossing scenario was employed for testing unexpected situations. In this environment, pedestrians are controlled with ORCA [28]. The learning-based method and switching administrator were trained with only 200 episodes of data in a square-crossing scenario, and an evaluation was conducted with 500 test cases in a circle-crossing scenario. In the training and

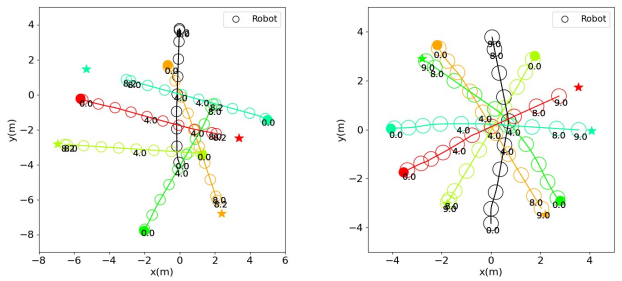


Fig. 5. Trajectories of examples of simulation environment. Left figure describes square-crossing scenario, and right figure describes circle-crossing scenario. Colored filled circles indicate start positions of each pedestrian, and colored stars indicate goal positions of each pedestrian.

testing phases of all the experiments, we set the number of pedestrians to 5.

B. Results of Performance Improvement

Experiments were conducted to confirm whether the proposed switching method improves performance. The trajectories of the results of the rule-based only, learning-based only, and the proposed methods for the four episodes are illustrated in Fig. 6. Although the learning-only method tended to reach the destination faster than the rule-based method, it caused more collisions. The proposed method succeeded even in episodes where the learning-based method caused collisions and reached the goal faster than the rule-based method (especially in Episode 4). For Episode 4, Fig. 7 presents the trajectory of the robot's movement and the timing of switching for the learning-based only and the proposed method. This figure demonstrates that collision can be avoided by switching to the rule-based method before the collision occurs with the learning-based method alone.

Then, Fig. 8 presents a heatmap about positions where switching happens. This figure depicts that switching occurs mostly near the center of the environment. This is because there are many situations where all pedestrians and robots gather near the center in the circle-crossing scenario adopted for evaluation, whereas in the square-crossing scenario adopted for training, there are few situations. This result indicates a tendency for the switching administrator to transition from the navigation method to rule-based methods in unexpected situations during training. Subsequently, a numerical comparison is presented. Table I presents the evaluation of the rule-based and learning-based methods in the square-crossing scenario. Table II presents the evaluation of the rule-based, learning-based, and proposed methods and the results of the evaluation of the rule-based, learning-based, random switching, and proposed methods in the circle-crossing scenario. In this section, the comparison focuses on the results of the rule-based, learning-based, and proposed methods.

Comparing the results of the learning-based method in the square-crossing scenario with those in the circle-crossing scenario, the success rate in the square-crossing scenario was greater than 90%, whereas in the circle-crossing scenario, it was only 60%. This result indicates that even models trained to perform well in square-crossing scenarios do not perform well in circle-crossing scenarios.

The results of the circle-crossing scenario reveal that the learning-based method has a success rate of 63.8%, whereas the proposed switching method has a success rate of 97.8%, which is a significant improvement. The proposed method, which switches between the learning and rule bases, reaches the destination in approximately 8.8 s, whereas the rule base alone takes approximately 10.0 s to reach the destination.

TABLE I

NUMERICAL COMPARISON IN SQUARE-CROSSING SCENARIO

Method	Success [%]	Collision [%]	Exec. time [s]
Rule-based (ORCA)	100.0	0.0	8.49 $\pm$ 1.13
Learning-based	91.2	7.8	8.29 $\pm$ 0.31

TABLE II

NUMERICAL COMPARISON IN CIRCLE-CROSSING SCENARIO

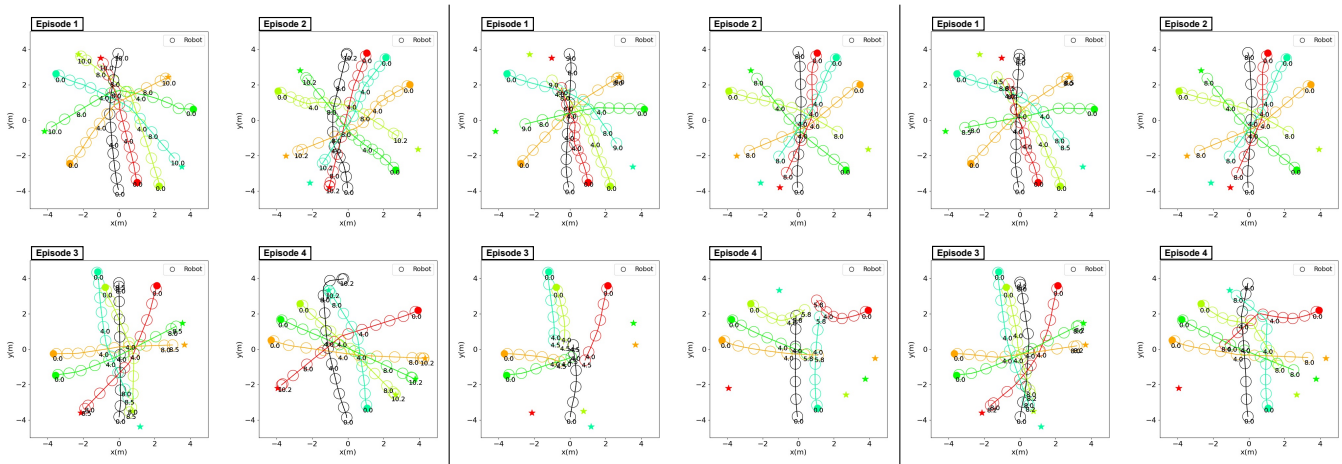
Method	Success [%]	Collision [%]	Exec. time [s]
Rule-based (ORCA)	100.0	0.0	10.02 $\pm$ 0.96
Learning-based	63.8	36.2	8.70 $\pm$ 0.87
Random switching	77.8	22.2	8.98 $\pm$ 0.70
Proposed	97.8	2.2	8.84 $\pm$ 0.84

Next, we compared the proposed method with a case where the rule and learning bases were randomly switched. In the evaluation scenario, the percentage of switching from the learning base to the rule base was 60.4% when employing the proposed method. Therefore, we demonstrated the difference between the proposed method and random switching to the rule-based method with a probability of 60.4%. The numerical results are presented in the random switching row in Table II. This results are the average of the results tested with 100 random seed values. Although the success rate improved compared with that of the learning-based method alone, the improvement was limited compared with that of the proposed method.

Furthermore, Table III presents how the results of each learning-based scenario changed with the proposed method and random switching and how the results of each scenario changed when the switch occurred. Each column includes the following information:

- S-S : Number of successful episodes with the learning-based alone and with switching.
- S-F : Number of episodes that succeed with the learning-based alone but fail after switching is attached.
- F-S : Number of episodes that fail in the learning-based alone and succeed after switching is attached.
- F-F : Number of episodes that fail in the learning-based alone and also fail after switching is attached.

As presented in the table, only one episode of S-F was observed in the proposed method, accounting for approximately 0.6% of the scenarios in which switching changed the results, whereas 59 episodes of S-F were observed in the random switching, accounting for 31%. This indicates that even scenarios that were successful only with the learning-based method often failed in the case of random switching.



(a) Trajectories of the rule-based method. (b) Trajectories of the learning-based method. (c) Trajectories of the proposed method.

Fig. 6. Trajectories of results of each method in four episodes. Black trajectories describe robot, and colored trajectories describe pedestrians.

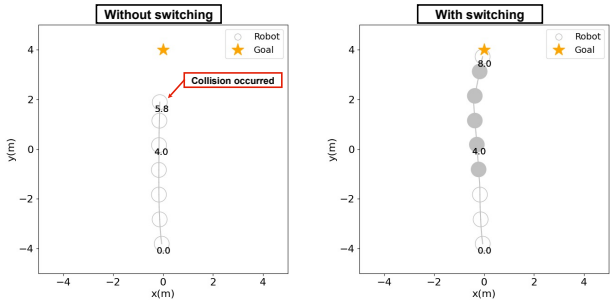


Fig. 7. Comparison between with and without switching. Gray circles indicate positions where learning-based method is adopted, and filled circles indicate positions where rule-based method is adopted.

TABLE III

NUMERICAL COMPARISON OF NUMBER OF CHANGE IN RESULTS

Method	S-S	S-F	F-S	F-F
Random switching	260	59	129	52
Proposed	318	1	171	10

This result indicates that random switching is ineffective for harnessing the performance of the learning-based method, whereas the proposed method is more effective and highlights the necessity of switching at appropriate timings.

VI. REAL-WORLD EXPERIMENTS

A real-world experiment was conducted to verify whether the proposed method could be adopted in practical situations.

A. System configuration

We developed a real robot system to perform experiments. The system is based on the mecanum rover (Lynxmotion - A4WD3). The system has 2D-LiDAR (UST-20LX) and a computer (Jetson AGX Orin) for processing. The software systems were created utilizing ROS2 Humble, and pedestrian detection was carried out by implementing DR-SPAAM

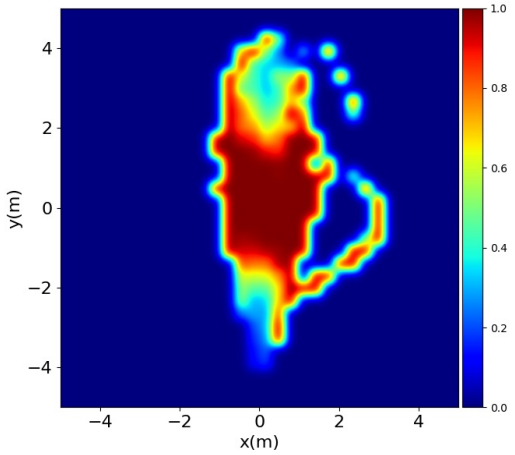


Fig. 8. Heatmap of positions where switching happens. Note that normalization by number of visits to each location is applied.

[31], [32]. The velocity of pedestrians was determined by analyzing the change in their positions and matching them across frames using the Hungarian method.

B. Experimental setup

The experimental environment was a room with dimensions of around 7.5 meters for both length and width. Five pedestrians were assigned to walk across the space, following the circle-crossing scenario. Each of the three methods were tested with 25 episodes. The robot was considered successful when it reached the edge of the environment without colliding with pedestrians.

C. Experimental results

Fig. 9 shows the experiment and the miracle of the robot when the proposed method is used in the experiment. As illustrated in the figure, the proposed method enabled successful navigation of real robot and pedestrian avoidance in a real-world environment. Regarding the success rate, the rule-based method was successful 21 times, the learning-based

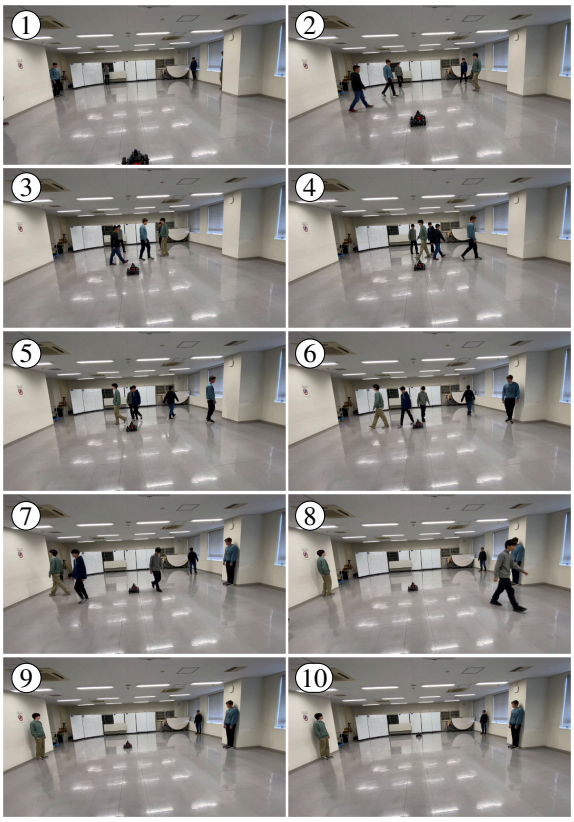


Fig. 9. Scenes depicting the robot navigating to its destination while avoiding pedestrians using the proposed method in the real-world experiment.

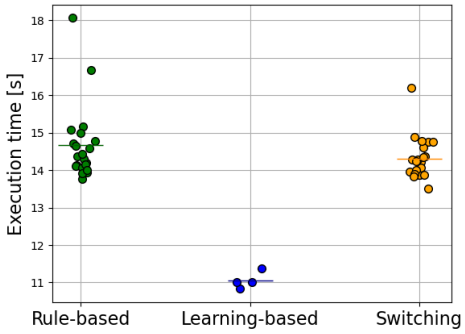


Fig. 10. Individual value plot of execution time for successful episodes for each method. Note that the velocity is limited by maximum velocity of actual robot system. The colored horizontal lines represent the average value for each method.

method 4 times, and the proposed method achieved success 21 times. The results validated that beyond simulation environments, the proposed switching method improved success rates in real-world experiments compared to relying solely on the learning-based method. Fig. 10 shows individual value plot of execution time for successful episodes for each method. The results revealed that while the proposed method marginally outperformed the rule-based approach in terms of median performance, no significant difference was observed. Although the learning-based approach achieved only 4 successful episode, it reached the goal faster compared to the other two methods.

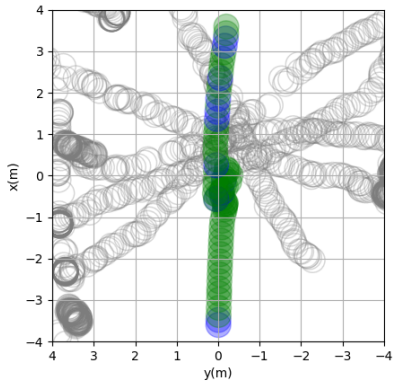


Fig. 11. Example of trajectories of the robot in the real-world experiment. Blue markers indicate positions where learning-based method is adopted, and green markers indicate positions where rule-based method is adopted. The gray circles represent trajectories of detected pedestrians.

D. Discussion

In the real-world experiments, the proposed method demonstrated a reduction in collisions compared to solely utilizing the learning-based approach. However, the improvement in execution time was not as pronounced as observed in the simulation environment when compared to the rule-based method alone. One potential factor behind this is the lack of real-world data used for training. As evidenced by the robot trajectories shown in Fig. 11, the rule-based approach appeared to dominate a larger proportion of the overall operation compared to the simulation experiments. Quantitative analysis revealed that 87.8% of the actions were governed by the rule-based method. Since the proposed method evaluates the likelihood of the learning data as a criterion, practical deployment would necessitate the use of data collected from real-world environments. Additionally, detection errors of pedestrian are considered a contributing factor. While only 5 pedestrians were present during the experiments, instances of up to 12 pedestrians being detected occurred. As it is inherently challenging to avoid detection errors caused by noise and other factors in real-world settings, it is crucial to address such aspects moving forward.

VII. Conclusion

This study proposed a method for mobile robot navigation in crowded environments with pedestrians that can switch between rule-based and learning-based methods by employing a normalizing flow and can handle environments that are unexpected during training. Experimental results confirmed that the proposed method achieves a higher success rate than the learning-based method and reaches the destination faster than the rule-based method. A comparison experiment with the random-switching method revealed that the proposed method performs better. Furthermore, the proposed method was confirmed to switch to rule-based switching while leveraging learning-based performance more effectively than random switching. Additionally, the real-world experiment showed that the proposed method reduces collisions compared to solely utilizing the learning-based method.

Our future work includes conducting more comprehensive experiments in real-world environments. Additionally, as discussed in Section VI-D, we aim to address the challenges that may arise in practical scenarios. Furthermore, since the switching administrator evaluates the likelihood to the learned dataset, it is impossible to directly assess whether a situation can be handled by the learning-based method. Consequently, we will develop an approach that considers the performance of learning-based methods during the switching.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI Grant Number JP20H00230 and a collaborative research with KYUDENKO Corporation.

REFERENCES

[1] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized Non-communicating Multiagent Collision Avoidance with Deep Reinforcement Learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 285–292, 2017.

[2] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially Aware Motion Planning with Deep Reinforcement Learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1343–1350, 2017.

[3] M. Everett, Y. F. Chen, and J. P. How, "Motion Planning among Dynamic, Decision-Making Agents with Deep Reinforcement Learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3052–3059, 2018.

[4] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019.

[5] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot Navigation in Crowds by Graph Convolutional Networks With Attention Learned From Human Gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[6] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot Navigation in Crowds by Graph Convolutional Networks with Attention Learned from Human Gaze," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2754–2761, 2020.

[7] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational Graph Learning for Crowd Navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10007–10013, 2020.

[8] L. Lucia, D. Daniel, C. Gianluca, S. Roland, and D. Renaud, "Robot Navigation in Crowded Environments Using Deep Reinforcement Learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5671–5677, 2020.

[9] X. Zhang, W. Xi, X. Guo, Y. Fang, B. Wang, W. Liu, and J. Hao, "Relational Navigation Learning in Continuous Action Space among Crowds," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3175–3181, 2021.

[10] S. Yao, G. Chen, Q. Qiu, J. Ma, X. Chen, and J. Ji, "Crowd-Aware Robot Navigation for Pedestrians with Multiple Collision Avoidance Strategies via Map-based Deep Reinforcement Learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8144–8150, 2021.

[11] K. Matsumoto, A. Kawamura, Q. An, and R. Kurazume, "Mobile Robot Navigation Using Learning-Based Method Based on Predictive State Representation in a Dynamic Environment," in *Proceedings of the IEEE/SICE International Symposium on System Integration (SII)*, pp. 499–504, 2022.

[12] S. Matsuzaki and Y. Hasegawa, "Learning Crowd-Aware Robot Navigation from Challenging Environments via Distributed Deep Reinforcement Learning," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pp. 4730–4736, 2022.

[13] K. Katyal, Y. Gao, J. Markowitz, S. Pohland, C. Rivera, I.-J. Wang, and C.-M. Huang, "Learning a Group-Aware Policy for Robot Navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11328–11335, 2022.

[14] R. Wang, W. Wang, and B.-C. Min, "Feedback-efficient Active Preference Learning for Socially Aware Robot Navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11336–11343, 2022.

[15] Y. Yang, J. Jiang, J. Zhang, J. Huang, and M. Gao, "ST^2: Spatial-Temporal state transformer for Crowd-Aware autonomous navigation," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 912–919, 2023.

[16] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. Livingston McPherson, J. Geng, and K. Driggs-Campbell, "Intention Aware Robot Crowd Navigation with Attention-Based Interaction Graph," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12015–12021, 2023.

[17] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-Aware Online POMDP Planning for Autonomous Driving in a Crowd," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 454–460, 2015.

[18] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "PORCA: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3418–3425, 2018.

[19] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially Compliant Navigation Through Raw Depth Inputs with Generative Adversarial Imitation Learning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1111–1117, 2018.

[20] C.-E. Tsai and J. Oh, "A Generative Approach for Socially Compliant Navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2160–2166, 2020.

[21] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially CompliAnt Navigation Dataset (SCAND): A Large-Scale Dataset of Demonstrations for Social Navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11807–11814, 2022.

[22] N. Pérez-Higueras, R. Otero, F. Caballero, and L. Merino, "HuNavSim: A ROS 2 Human Navigation Simulator for Benchmarking Human-Aware Robot Navigation," *IEEE robotics and automation letters*, vol. 8, no. 11, pp. 7130–7137, 2023.

[23] Q. Huang, C. Pu, D. Fourie, K. Khosoussi, J. P. How, and J. J. Leonard, "NF-iSAM: Incremental smoothing and mapping via normalizing flows," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1095–1102, 2021.

[24] S. A. Khader, H. Yin, P. Falco, and D. Kragic, "Learning Stable Normalizing-Flow Control for Robotic Manipulation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1644–1650, 2021.

[25] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1906–1915, 2021.

[26] J. Liu, A. Kumar, J. Ba, J. Kiros, and K. Swersky, "Graph Normalizing Flows," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 13556–13566, 2019.

[27] J. Wu, Y. Wang, H. Asama, Q. An, and A. Yamashita, "Risk-Sensitive Mobile Robot Navigation in Crowded Environment via Offline Reinforcement Learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7456–7462, 2023.

[28] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-Body Collision Avoidance," in *Proceedings of the International Symposium of Robotic Research*, pp. 3–19, 2011.

[29] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Apr. 2017.

[30] S. Brody, U. Alon, and E. Yahav, "How Attentive are Graph Attention Networks?," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[31] D. Jia, A. Hermans, and B. Leibe, "DR-SPAAM: A Spatial-Attention and auto-regressive model for person detection in 2D range data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10270–10277, 2020.

[32] D. Jia, M. Steinweg, A. Hermans, and B. Leibe, "Self-Supervised person detection in 2D range data using a calibrated camera," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13301–13307, 2021.