2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

COLSON: Controllable Learning-Based Social Navigation via Diffusion-Based Reinforcement Learning

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

***Abstract*— Navigation of mobile robots in dynamic environments with pedestrian traffic poses a significant challenge in the development of autonomous mobile service robots. Recently, deep reinforcement learning-based methods have been actively studied and have outperformed traditional rule-based approaches, owing to their optimization capabilities. Among these methods, those assuming continuous action spaces typically use Gaussian distributions, limiting the flexibility of action generation. By contrast, the application of diffusion models to reinforcement learning has advanced, allowing more flexible action distributions than Gaussian policy-based approaches. In this study, we used a diffusion-based reinforcement learning approach to social navigation and validated its effectiveness. Furthermore, using the characteristics of diffusion models, we propose extensions that allow adaptation to previously unseen scenarios without additional training. As concrete scenario examples, we show adaptability to scenarios in which static obstacles exist in an environment that was not present during training, as well as scenarios in which the objective differs from training, such as accompanying a target pedestrian while avoiding other pedestrians to reach a destination.**

I. INTRODUCTION

Social navigation, which allows safe and efficient navigation in dynamic environments with pedestrian traffic, is crucial for developing autonomous mobile service robots. In recent years, deep reinforcement learning-based methods have been extensively studied owing to the advancements in machine learning. Among these, methods that assume continuous action spaces require the modeling of the action distribution with predefined functions, such as Gaussian distributions. Generative models have advanced rapidly in recent years, and diffusion models have achieved remarkable success in tasks such as image generation. Their application to reinforcement learning has also been investigated, demonstrating a strong performance in continuous control tasks owing to their expressive capacity and lack of reliance on Gaussian distribution assumptions. Motivated by this, we use a diffusion-based reinforcement learning approach to mobile-robot navigation in dynamic environments with pedestrian traffic. This method outperforms Gaussian policy-based approaches, maintaining high performance even as the number of pedestrians in the environment or their behavioral patterns change.

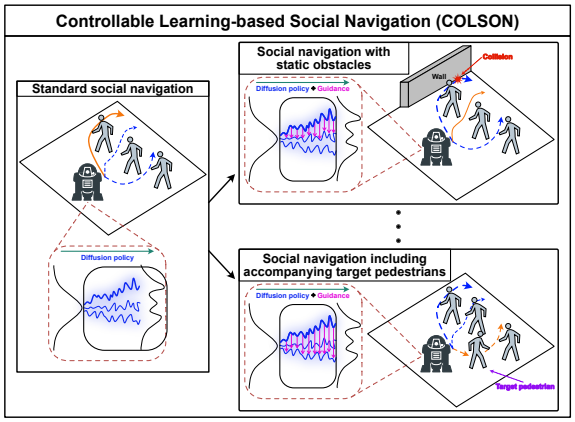

Fig. 1. Conceptual diagram of the proposed method. In standard social navigation, the proposed method uses the multimodality of diffusion models to generate various actions that allow robots to avoid pedestrians. The blue dashed arrows represent examples of candidate actions generated in this situation, whereas the solid orange line indicates the actual action selected. In environments with static obstacles, actions generated solely by the diffusion model may result in collisions with walls. Therefore, guidance is applied to steer the selection toward alternative candidates that avoid wall collisions. In tasks involving pedestrian following, diffusion models alone tend to generate behaviors that move away from pedestrians. Therefore, guidance is used to generate actions that follow the target pedestrian.

Furthermore, the proposed method leverages the diverse action generation capabilities of policies obtained through diffusion-based reinforcement learning to adapt to environments and tasks that were not considered during training and require no additional training. Two scenarios are considered in this study. The first scenario involved social navigation in environments with static obstacles that were not present during the training phase. The second scenario involved the companion tasks. This objective was not considered during training. The task required agents to follow a designated pedestrian while avoiding other pedestrians to achieve their objectives. Considering these adaptability features, the proposed method is called **co**ntrollable **l**earning-based **so**cial **n**avigation (COLSON). A conceptual diagram of COLSON is shown in Fig. 1.

The main contributions of this study are as follows:

- We integrate a diffusion-based reinforcement learning method with a graph neural network (GNN) architecture and use it for social navigation tasks. To the best of our knowledge, this is the first application of diffusion-based reinforcement learning to social navigation.
- We propose an annealing method to improve the performance of diffusion-based reinforcement learning in

*This study was not supported by any organization

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

[2]Yuki Tomita and Yuki Hyodo are with the Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan tomita@irvs.ait.kyushu-u.ac.jp, hyodo@irvs.ait.kyushu-u.ac.jp

†Authors contributed equally

social navigation.

- We show that the proposed method can adapt to environments with static obstacles and companion tasks without requiring additional training.
- Through comparative experiments with conventional methods, we show that the proposed method outperforms existing approaches and shows high performance, even in environments where the number of pedestrians differs from that in the training phase, highlighting its scalability.
- We conduct a real-world demonstration to validate that the proposed method is applicable to an actual robot in practical situations.

II. RELATED WORK

A. Social Navigation

Numerous deep reinforcement learning-based social navigation methods have been proposed. Chen et al. [1] introduced a value-based learning model for multiagent collision avoidance, marking a pioneering achievement in deep reinforcement learning approaches aimed at allowing autonomous mobile robots to safely navigate dynamic environments. Based on this, Chen et al. [2] proposed a method that incorporated social norms into the reward function to improve pedestrian collision avoidance and further addressed the challenge of pedestrian switching in multiagent scenarios by sharing weights in the pedestrian information processing layer and using max pooling. In addition to the direct application of deep reinforcement learning, architectural improvements in neural networks have been explored to improve performance and develop methods that can accommodate an arbitrary number of pedestrians [3], [4], [5], [6], [7], [8]. Other studies have expanded the scope beyond performance improvement and scalability to pedestrian density by investigating approaches that leverage human gaze information [9], using map-based processing to allow navigation in environments with both static and dynamic obstacles [10], [11], and incorporating continuous action spaces [12]. More recently, hybrid approaches that integrate reinforcement learning-based methods with rule-based methods have been proposed to further improve the robustness and adaptability [13], [14].

Despite extensive research on social navigation, no studies have used diffusion-based reinforcement learning approaches to generate both high-performance and diverse behaviors that can be adaptively guided based on specific situations or adapted to novel environments and emerging applications.

B. Diffusion Models for Action Generation

Diffusion models have achieved remarkable success in various data generation tasks, including image synthesis [15], [16], [17]. Recent studies have extended their application beyond vision-based data to action generation in robotics through imitation and reinforcement learning. In the field of imitation learning, methods based on behavior cloning (BC) have been proposed [18], [19], [20], in which diffusion models are trained to generate actions that replicate target demonstration data, making it the most fundamental approach for applying diffusion models to action generation. In addition to imitation learning, diffusion models have been applied to offline reinforcement learning [21], [22], [23], [24], where exploration is not required, thereby allowing the models to learn distributions from datasets in a manner similar to imitation learning. More recently, efforts have been directed toward applying diffusion models to online reinforcement learning [25], particularly in settings that differ from conventional diffusion model formulations, such as learning without access to presampled data from an optimal target distribution [26] and approaches based on weighted regression [27].

Furthermore, in the context of applying diffusion models to mobile robots, methods using diffusion policy for goal-conditioned navigation and undirected exploration have been proposed [28]. Other approaches use models trained on ground truth data from the A* algorithm for global path planning [29], [30].

From the perspective of diffusion model guidance, methods capable of generating actions that simultaneously satisfy multiple constraints or combine multiple skills [31], controllable learning-based pedestrian simulation [32], and a visual navigation method that uses depth estimation-based cost guidance have been proposed, demonstrating high performance even in unseen environments [33].

However, the application of diffusion-based reinforcement learning to social navigation remains largely unexplored.

III. PRELIMINARIES

This study addresses the issue of two-dimensional (2D) mobile-robot navigation while ensuring collision avoidance with pedestrians. The state of the robot includes its position from the goal, velocity, and orientation information $\boldsymbol{s}^r = [{}^g\boldsymbol{p}^r, \boldsymbol{v}^r, \theta]$. By contrast, the position and velocity information of each pedestrian is converted to a robot-centered coordinate system and used as a state $\boldsymbol{s}^n = [{}^{rc}\boldsymbol{p}^n, {}^{rc}\boldsymbol{v}^n]$. The set of pedestrian states in an environment is denoted by $\boldsymbol{s}^h = \left[\boldsymbol{s}^1, \boldsymbol{s}^2, \ldots, \boldsymbol{s}^n\right]$. The robot's action includes velocity values in the X- and Y-directions, $\boldsymbol{a} = \left[v_x^c, v_y^c\right]$. The robot obtains rewards after performing these actions. In this study, we used the form of the reward function from previous research [4]. R_t denotes the reward function at time t as follows:

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

where d_t denotes the minimum separation distance between the robot and pedestrians, $\boldsymbol{p}_t^r$ denotes the position of the robot, and $\boldsymbol{p}_t^g$ denotes the goal position at time t.

IV. APPROACH

This section presents the model architecture, training methodology, and guidance methods for the static obstacle avoidance and companion tasks.

A. Model Architecture

The architecture of the proposed method is shown in Fig. 2. This model uses an actor–critic framework in which both the actor and critic include GNNs. Each GNN employs an attention-based mechanism, as in a previous study [5]. The actor uses a diffusion model that considers the output of the GNN as a condition and generates actions. The critic network inputs the output of the GNN into a multilayer perceptron to compute the Q-values.

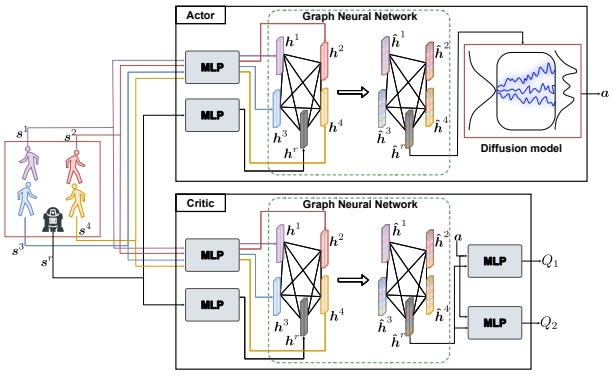

Fig. 2. Architecture of the proposed method. $\boldsymbol{s}^r$ and $\boldsymbol{s}^n$ denote the states of each robot and pedestrian, respectively. $\boldsymbol{h}^n$ denotes the features of each robot and pedestrian. $\hat{\boldsymbol{h}}^n$ denotes the features after the GNN is applied.

B. Q-Score Matching and Annealing

In this study, we use Q-score matching (QSM) [26] as the training framework. Diffusion models are typically trained using score functions for the target distribution. However, in reinforcement learning, where the objective is to learn an optimal policy, the optimal policy is unknown. Consequently, neither the score function nor the samples from the optimal policy can be precomputed. To address this issue, QSM facilitates reinforcement learning with diffusion models by using the action gradient of the Q-function as a score. The training procedure using QSM is presented in Algorithm 1. The coefficient α denotes the inverse temperature parameter. Although smaller values of α improve exploration during reinforcement learning, they also make it more difficult for the final action distribution to converge. Conversely, larger values of α have the opposite effect. While the previous study fixed this coefficient during training, we propose an annealing technique that gradually varied α during training. In this study, we set the target value $\hat{\alpha}$ to the value that achieves the highest return in a preliminary experiment. The proposed method linearly anneals α from one to the target value $\hat{\alpha}$ during the training. This schedule allows the agent to explore broadly in the early stages of training while promoting convergence toward optimal actions in later stages.

C. Guidance for Static Obstacle Avoidance

In this section, we describe the first example of action generation guidance for realizing social navigation with static obstacles. The training environment in this study only included pedestrians, implying that the trained policy does not inherently have the capability to avoid static obstacles.

To address this issue, we incorporate guidance into the diffusion model to steer actions away from static obstacles, thereby allowing the robot to avoid both pedestrians and static obstacles. In particular, the guidance strength increases as the robot approaches static obstacles, directing its movement away from them. The execution procedure for this guidance is shown in Algorithm 2.

In this algorithm, $\mathcal{J}_w(\hat{\boldsymbol{x}}_0)$ is calculated using the distance from the nearest point of the static obstacles $\boldsymbol{p}^w$ before the

Algorithm 1: Training algorithm using QSM

Initialize the score network Ψ_θ and the critic networks Q_{ϕ^1} and Q_{ϕ^2}
Set the parameter values of the target critics $Q_{\phi',1}$ and $Q_{\phi',2}$ equal to those of the main critics
for $i = 1$ **to** E **do**
  Explore using the policy until finishing an episode
  After finishing an episode, store the trajectory of $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}'_t)$ to the replay buffer $\mathcal{D}$
  Sample batch $\mathcal{B} = \{(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')\}$ from replay buffer $\mathcal{D}$
  Sample actions for computing targets $\hat{\boldsymbol{a}}' \sim \pi_\theta(\cdot \mid \boldsymbol{s}')$
  Calculate targets for the Q-function
  $$y(r, \boldsymbol{s}') = r + \gamma \left(\min_{i=1,2} Q_{\phi',i}(\boldsymbol{s}', \hat{\boldsymbol{a}}') \right)$$
  Update the critics by minimizing
  $$L_{\text{critic}} = \frac{1}{|\mathcal{B}|} \sum (Q_{\phi^i}(\boldsymbol{s}, \boldsymbol{a}) - y(r, \boldsymbol{s}'))^2 \quad \text{for } i = 1, 2$$
  Update the score network by minimizing
  $$L_{\text{QSM}} = \frac{1}{|\mathcal{B}|} \sum \|\Psi_\theta(\boldsymbol{s}, \boldsymbol{a}) - \alpha \nabla_a Q(\boldsymbol{s}, \boldsymbol{a})\|^2$$
  Update the target critic using polyak averaging
  $\phi'^{,i} \leftarrow \rho \phi'^{,i} + (1-\rho)\phi_i \quad \text{for } i = 1, 2.$
  ;
end

Algorithm 2: Action generation with guidance for static obstacle avoidance

Input: Trained score network Ψ_θ, states $\boldsymbol{s}_t = [\boldsymbol{s}_t^r, \boldsymbol{s}_t^h]$ at time t, current robot position $\boldsymbol{p}_t^r$ at time t, robot radius d_r, nearest point of the static obstacles $\boldsymbol{p}^w$, number of denoising steps T, and noise scheduling parameters α_τ, $\bar{\alpha}_\tau$, β_τ, and $\bar{\beta}_\tau$

$\boldsymbol{x}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$,
for $\tau = T$ **to** 1 **do**
  Denoise by score and guidance using
  $$\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_\tau}} \boldsymbol{x}_T - \frac{\sqrt{1-\bar{\alpha}_\tau}}{\sqrt{\bar{\alpha}_\tau}} \Psi_\theta(\boldsymbol{s}_t, \boldsymbol{x}_\tau, \tau)$$
  $\sigma_\tau = \sqrt{\frac{1-\bar{\alpha}_{\tau-1}}{1-\bar{\alpha}_\tau}}\, \beta_\tau$, Compute guidance cost $\mathcal{J}_w(\hat{\boldsymbol{x}}_0)$:
  $$\boldsymbol{v}_{\text{before}} = \boldsymbol{p}^w - \boldsymbol{p}_t^r, \quad \boldsymbol{v}_{\text{after}} = \boldsymbol{p}^w - (\boldsymbol{p}_t^r + \Delta t \hat{\boldsymbol{x}}_0)$$
  $$c = \frac{\boldsymbol{v}_{\text{before}} \cdot \boldsymbol{v}_{\text{after}}}{\|\boldsymbol{v}_{\text{before}}\| \|\boldsymbol{v}_{\text{after}}\| + \epsilon}$$
  $$\xi = \|\boldsymbol{v}_{\text{after}}\| - d_r$$
  $$\mathcal{J}_w(\hat{\boldsymbol{x}}_0) = \begin{cases} 0, & \boldsymbol{v}_{\text{before}} \cdot \hat{\boldsymbol{x}}_0 < 0 \ \ \& \ \ c > 0 \\ c \cdot \exp(-\xi^2), & \boldsymbol{v}_{\text{before}} \cdot \hat{\boldsymbol{x}}_0 \geq 0 \ \ \& \ \ c > 0 \\ -c \cdot (|\xi| + 1), & c \leq 0 \end{cases}$$
  $\hat{\boldsymbol{x}}_0 \leftarrow \hat{\boldsymbol{x}}_0 - \sigma_\tau \nabla_{\boldsymbol{x}_\tau} \mathcal{J}_w(\hat{\boldsymbol{x}}_0)$
  $$\boldsymbol{\mu}_\tau = \frac{\beta_\tau \sqrt{\bar{\alpha}_{\tau-1}}}{1-\bar{\alpha}_\tau} \hat{\boldsymbol{x}}_0 + \frac{\sqrt{\alpha_\tau}\left(1-\bar{\alpha}_{\tau-1}\right)}{1-\bar{\alpha}_\tau} \boldsymbol{x}_\tau$$
  $\boldsymbol{x}_{\tau-1} \sim \mathcal{N}(\boldsymbol{\mu}_\tau, \sigma_\tau \boldsymbol{I})$
end
Execute the action $\boldsymbol{a}_t = \boldsymbol{x}_0$

TABLE I

NUMERICAL COMPARISON IN CIRCLE CROSSING SCENARIO WITH FIVE ORCA PEDESTRIANS

	Visible				Invisible			
Method	Success [%] ↑	Collision [%] ↓	Exec. time [s] ↓	Return ↑	Success [%] ↑	Collision [%] ↓	Exec. time [s] ↓	Return ↑
ORCA (for reference)	100.00 ± 0.00	0.00 ± 0.00	10.02 ± 0.00	0.542 ± 0.000	42.80 ± 0.00	56.80 ± 0.00	10.93 ± 0.00	0.081 ± 0.000
BC	77.28 ± 0.17	22.56 ± 0.18	10.06 ± 0.61	0.377 ± 0.001	5.08 ± 0.21	94.92 ± 0.21	10.24 ± 0.81	-0.182 ± 0.002
AWR	87.60 ± 1.96	12.36 ± 1.97	11.44 ± 1.80	0.466 ± 0.009	42.44 ± 4.78	57.56 ± 4.78	11.48 ± 2.16	0.122 ± 0.030
CAWR	61.00 ± 0.57	35.28 ± 0.45	11.98 ± 8.13	0.236 ± 0.001	12.48 ± 2.11	86.68 ± 2.34	12.96 ± 10.21	0.122 ± 0.030
QVWR	62.48 ± 3.84	34.24 ± 3.86	11.44 ± 5.63	0.245 ± 0.020	6.68 ± 1.00	92.92 ± 1.10	11.44 ± 6.04	-0.173 ± 0.007
AWAC	71.08 ± 2.91	28.92 ± 2.91	11.32 ± 5.83	0.327 ± 0.011	29.76 ± 7.20	70.24 ± 7.20	11.31 ± 5.69	0.010 ± 0.038
DIPO	80.80 ± 1.97	17.52 ± 1.42	11.29 ± 1.26	0.327 ± 0.011	30.32 ± 3.41	68.96 ± 3.03	12.22 ± 0.15	0.022 ± 0.021
QVPO	98.24 ± 0.02	1.68 ± 0.01	9.75 ± 0.97	0.618 ± 0.001	97.44 ± 0.03	2.36 ± 0.03	9.99 ± 0.79	0.606 ± 0.001
QSM	98.28 ± 0.03	1.60 ± 0.03	$\mathbf{8.70 \pm 0.07}$	$\mathbf{0.647 \pm 0.000}$	98.44 ± 0.01	1.32 ± 0.01	$\mathbf{9.05 \pm 0.05}$	$\mathbf{0.640 \pm 0.000}$
QSM-A (w/ annealing)	$\mathbf{98.68 \pm 0.00}$	$\mathbf{0.84 \pm 0.00}$	9.24 ± 0.23	0.637 ± 0.000	$\mathbf{98.88 \pm 0.00}$	$\mathbf{0.60 \pm 0.00}$	9.37 ± 0.20	0.636 ± 0.000

TABLE II

NUMERICAL COMPARISON IN CIRCLE CROSSING SCENARIO WITH FIVE SOCIAL FORCE PEDESTRIANS

	Visible				Invisible			
Method	Success [%] ↑	Collision [%] ↓	Exec. time [s] ↓	Return ↑	Success [%] ↑	Collision [%] ↓	Exec. time [s] ↓	Return ↑
ORCA (for reference)	90.80 ± 0.00	9.20 ± 0.00	9.79 ± 0.00	0.544 ± 0.000	19.00 ± 0.00	81.00 ± 0.00	10.82 ± 0.00	-0.088 ± 0.000
BC	63.4 ± 1.30	36.6 ± 1.30	10.16 ± 0.73	0.303 ± 0.007	0.28 ± 0.00	99.72 ± 0.00	14.62 ± 75.05	-0.222 ± 0.000
AWR	86.92 ± 0.57	13.04 ± 0.58	11.52 ± 2.18	0.483 ± 0.007	41.68 ± 11.13	58.28 ± 11.16	11.37 ± 2.00	0.131 ± 0.076
CAWR	63.32 ± 1.94	34.20 ± 2.25	12.28 ± 11.09	0.278 ± 0.005	6.80 ± 0.90	92.48 ± 1.12	21.17 ± 70.54	-0.172 ± 0.006
QVWR	58.04 ± 0.89	36.36 ± 1.91	12.46 ± 7.28	0.245 ± 0.003	3.60 ± 0.33	96.08 ± 0.38	23.85 ± 70.94	-0.196 ± 0.003
AWAC	80.64 ± 3.24	19.32 ± 3.25	11.47 ± 7.14	0.431 ± 0.022	42.68 ± 6.67	57.32 ± 6.67	15.53 ± 70.50	0.127 ± 0.048
DIPO	73.52 ± 0.41	25.00 ± 0.24	12.06 ± 1.30	0.362 ± 0.001	6.88 ± 0.63	92.32 ± 0.54	12.80 ± 0.72	-0.154 ± 0.004
QVPO	99.88 ± 0.00	0.04 ± 0.00	9.47 ± 1.08	0.642 ± 0.001	96.72 ± 0.08	3.20 ± 0.08	9.76 ± 0.99	0.607 ± 0.002
QSM	99.60 ± 0.00	0.40 ± 0.00	$\mathbf{8.40 \pm 0.03}$	$\mathbf{0.669 \pm 0.000}$	98.72 ± 0.00	1.20 ± 0.00	$\mathbf{8.62 \pm 0.07}$	$\mathbf{0.654 \pm 0.000}$
QSM-A (w/ annealing)	$\mathbf{99.92 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	8.72 ± 0.38	0.663 ± 0.000	$\mathbf{98.84 \pm 0.00}$	$\mathbf{0.92 \pm 0.00}$	8.91 ± 0.37	0.649 ± 0.000

robot moves based on the pseudo-noise-removed action $\hat{\boldsymbol{x}}_0$ inferred in each generation step τ. In addition, using the dot product of the vector from the robot's pre-movement position to $\boldsymbol{p}^w$ and $\hat{\boldsymbol{x}}_0$, and the cosine similarity c of the vectors from the robot's pre- and post-movement positions to $\boldsymbol{p}^w$, case distinctions are made; if the robot approaches static obstacles after moving, a gradual exponential cost is applied, but if the robot crosses over $\boldsymbol{p}^w$ after taking action, a linearly increasing cost with an offset is applied to impose a larger penalty. If the robot neither approaches nor crosses the static obstacles, then the cost is zero. By guiding action generation using the gradient of $\mathcal{J}_w(\hat{\boldsymbol{x}}_0)$, it is possible to avoid both static obstacles and pedestrians.

D. Guidance for Companion Tasks

As a second example of guidance, we describe a case of companion tasks in which the robot aims to reach its destination while accompanying the target pedestrians. During training, our method targets only the avoidance of pedestrians to reach the goal. Therefore, without guidance, it does not generate actions that are specifically designed to accompany pedestrians. The guidance strategy uses the previous position information of the target pedestrian to generate actions that minimize the distance between the robot and previous position of the target pedestrian. This allows action generation in which agents avoid other pedestrians while accompanying the target pedestrian toward their goal. The procedure for executing the guidance is presented in Algorithm 3. In this algorithm, $\mathcal{J}_c(\hat{\boldsymbol{x}}_0)$ is calculated as the distance between the position to which the robot moves based on the pseudo-noise-removed action $\hat{\boldsymbol{x}}_0$ estimated at each generation step τ and the position of the pedestrian from the previous step. The companion task is achieved by guiding the action generation using a gradient of $\mathcal{J}_c(\hat{\boldsymbol{x}}_0)$.

Algorithm 3: Action generation with guidance for companion tasks

Input: Trained score network Ψ_θ, states $\boldsymbol{s}_t = \left[\boldsymbol{s}_t^r, \boldsymbol{s}_t^h\right]$ at time t, current robot position $\boldsymbol{p}_t^r$, previous position of the target pedestrian $\boldsymbol{p}_{t-1}^i$, number of denoising steps T, and noise scheduling parameters α_τ, $\bar{\alpha}_\tau$, β_τ, and $\bar{\beta}_\tau$

$\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

for $\tau = T$ **to** 1 **do**

  Denoising by score and guidance using

  $\hat{\boldsymbol{x}}_0 = \frac{1}{\sqrt{\bar{\alpha}_\tau}} \boldsymbol{x}_T - \frac{\sqrt{1-\bar{\alpha}_\tau}}{\sqrt{\bar{\alpha}_\tau}} \Psi_\theta(\boldsymbol{s}_t, \boldsymbol{x}_\tau, \tau)$

  $\sigma_\tau = \sqrt{\frac{1-\bar{\alpha}_{\tau-1}}{1-\bar{\alpha}_\tau}}\, \beta_\tau$, Compute guidance cost $\mathcal{J}_c(\hat{\boldsymbol{x}}_0)$:

  $\mathcal{J}_c(\hat{\boldsymbol{x}}_0) = ||\boldsymbol{p}_t^i - (\boldsymbol{p}_t^r + \Delta t \hat{\boldsymbol{x}}_0)||^2$

  $\hat{\boldsymbol{x}}_0 = \hat{\boldsymbol{x}}_0 - \sigma_\tau \nabla_{\boldsymbol{x}_\tau} \mathcal{J}_c(\hat{\boldsymbol{x}}_0)$

  $\boldsymbol{\mu}_\tau = \frac{\beta_\tau \sqrt{\bar{\alpha}_{\tau-1}}}{1-\bar{\alpha}_\tau}\, \boldsymbol{x}_0 \; + \; \frac{\sqrt{\alpha_\tau}\left(1-\bar{\alpha}_{\tau-1}\right)}{1-\bar{\alpha}_\tau}\, \boldsymbol{x}_\tau$

  $\boldsymbol{x}_{\tau-1} \sim \mathcal{N}(\boldsymbol{\mu}_\tau, \sigma_\tau \boldsymbol{I})$

end

Execute the action $\boldsymbol{a}_t = \boldsymbol{x}_0$

V. EXPERIMENTS

The experiments were conducted in a simulation environment to evaluate the effectiveness of the proposed method.

A. Simulation Environment and Settings

We used the CrowdNav environment used in related studies [4], [5]. In this environment, pedestrians are controlled using ORCA [34]. The models were trained for 100,000 episodes in a visible circle crossing scenario with five pedestrians. In addition, data were collected using ORCA before each training session, and the training began with a dataset of 2000 episodes. The hyperparameter α of QSM and $\hat{\alpha}$ of QSM with annealing (QSM-A) were set to 400. The number of generation steps was set to 100 for each diffusion-based reinforcement learning method.

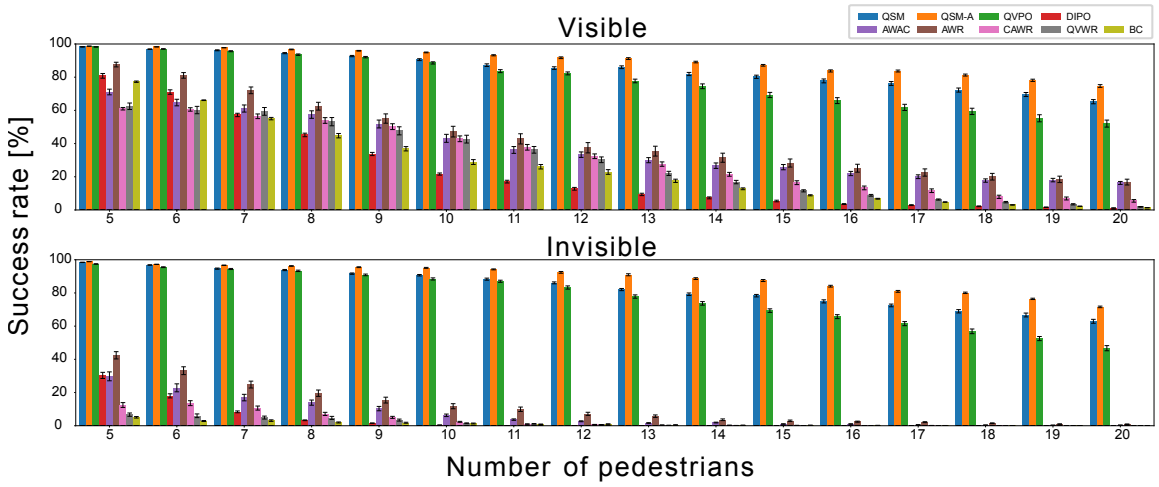

Fig. 3. Comparison of success rate while changing the number of pedestrians. The upper graph shows the results for the visible setting, while the lower graph shows the results for the invisible setting.

B. Performance Evaluation

The performance of each method was evaluated in a circle crossing scenario with pedestrians controlled by ORCA and another scenario with pedestrians controlled by the social force model. In each scenario, the methods were evaluated under two conditions: one where the pedestrians were aware of the robot's position (visible) and the other where they were unaware (invisible). For comparison, the proposed method was evaluated against BC, CAWR [12], which tackles social navigation with continuous action spaces; and related reinforcement learning methods, such as AWR [35], QVWR [36], and AWAC [37]. Furthermore, comparisons were made with other diffusion-based reinforcement learning methods, including DIPO [25] and QVPO [27]. For each method, five models were trained using different random seeds, and their average performance adopted as the evaluation result. The evaluation metrics included success rate, collision rate, average execution time, and average return. Each test scenario comprised 500 episodes.

The results are listed in Tables I and II. The $\pm$ symbol in each table indicates the standard deviation computed from the results of training runs using five independent random seeds. The tables demonstrate that QSM-based methods achieved superior performance in both ORCA and social force model [38] scenarios with visible and invisible settings. Whereas QSM demonstrated a slightly faster average execution time and superior performance from the aspect of return in comparison with QSM and QSM-A, QSM-A consistently achieved the highest success rate and lowest collision rate. Tables I and II demonstrate that methods other than QSM-based approaches showed significant performance degradation in an invisible setting. These results suggest that the QSM-based method outperformed the other approaches, maintaining high performance even when pedestrian behavior patterns change, such as in an invisible setting or when using the social force model for pedestrian simulation.

Fig. 3 shows a comparison of success rates with an increasing number of pedestrians in the circle crossing scenario under both visible and invisible settings. As shown in the figure, QSM-A consistently outperformed the other approaches for all the pedestrian numbers. The performance gap with QVPO was small for five pedestrians but widened as the number of pedestrians increased. This trend may be attributed to QVPO's reliance on weighted regression, which limits generalization to unseen scenarios. Compared with other Gaussian policy-based methods, such as AWR, CAWR, QVWR, and AWAC, the proposed QSM-based approach demonstrated significantly superior performance. This indicates that the QSM-based method outperformed Gaussian policy-based approaches, maintaining high performance even when the number of pedestrians changed in the environment.

C. Evaluation of Guidance for Static Obstacle Avoidance

The effectiveness of the guidance method for static obstacle avoidance was evaluated in an environment with walls and three pedestrians approaching from the front.

Fig. 4 shows a qualitative comparison. The results showed that when the robot initially intended to move left to avoid pedestrians, resulting in a potential collision with the wall, shifting to the right by guidance allowed it to avoid both pedestrians and the wall.

Table III lists a quantitative comparison of cases with and without guidance. In the table, Col-P represents the collision rate with pedestrians, and Col-W represents the collision rate with the walls. The results show that the proposed guidance method significantly reduced collisions with walls and improved the success rates. Although the collision rate with pedestrians increased compared to the case without guidance, this likely occurred because the robot collided with the walls before encountering pedestrians in the case without guidance. This phenomenon can be attributed to the slight increase in the execution time performance observed in the absence of guidance compared with the presence of guidance. These results demonstrate that the proposed method can guide action generation to allow pedestrian avoidance and navigation around static obstacles, which were not considered during the training phase.

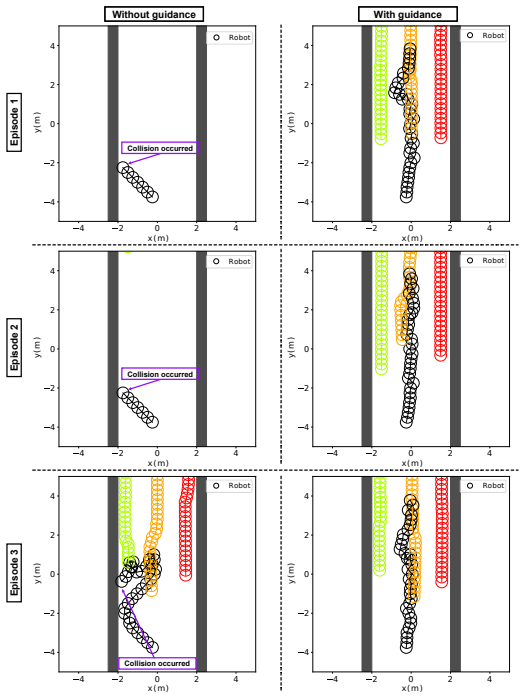

Fig. 4. Comparison between with and without guidance for static obstacle avoidance. Colored circles represent pedestrian trajectories, and the black rectangles represent walls.

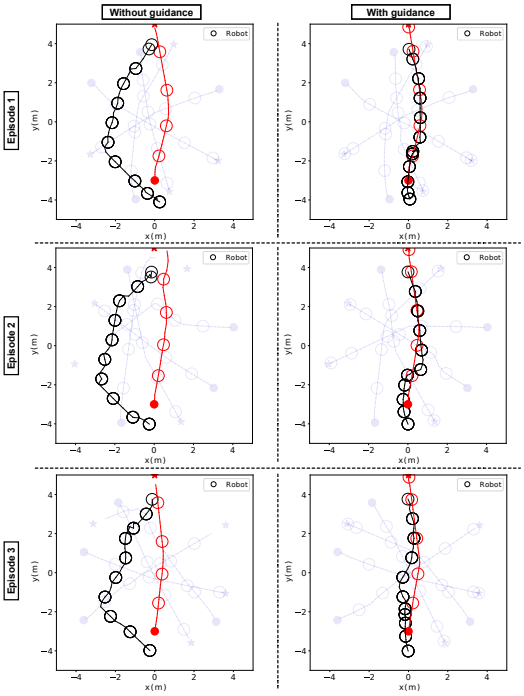

Fig. 5. Comparison between with and without guidance for companion tasks. The red circles indicate target pedestrians' trajectories, and blue trajectories indicate other pedestrians.

TABLE III

NUMERICAL COMPARISON IN ENVIRONMENTS WITH STATIC OBSTACLES

Method	Success [%]↑	Col-P [%]↓	Col-W [%]↓	Exec. time [s]↓	Return↑
w/o guidance	0.6	**0.00**	99.4	**8.17**	−0.226
w/ guidance	**97.2**	2.8	**0.0**	8.46	**0.640**

TABLE IV

NUMERICAL COMPARISON FOR COMPANION TASKS

Method	Success [%]↑	Collision [%]↓	Exec. time [s]↓	Return↑	FD [m]↓
w/o guidance	**96.4**	**3.6**	**9.09**	**0.619**	2.07
w/ guidance	92.0	7.4	10.48	0.549	**1.20**

D. Evaluation of Guidance for Companion Tasks

The guidance method for companion tasks was evaluated in a scenario involving five pedestrians. The assessment focused on two key aspects: a qualitative evaluation of the robot's behavioral changes in response to the guidance in directing the target pedestrian and a performance evaluation to ensure that the introduction of this guidance mechanism did not result in significant performance degradation.

Fig. 5 shows a qualitative comparison. These results demonstrate that, when guidance was applied, the robot followed pedestrian movements more closely than when guidance was not applied, confirming that the guidance mechanism successfully generated actions that accompany pedestrians.

The results of the quantitative evaluation are listed in Table IV. In the table, FD denotes the discrete Fréchet distance between the robot and the target pedestrian. These results demonstrates that although the overall performance decreased when using the guidance mechanism, the success rates remained above 90%, and the Fréchet distance was reduced by approximately 40%.

Overall, the proposed guidance method can modify the robot's behavior without causing significant performance degradation, even for tasks that contain objectives that are not considered during training.

VI. REAL-WORLD DEMONSTRATION

A real-world demonstration was conducted using the developed robot system across three scenarios. This demonstration verified the applicability of the proposed method to actual robotic systems. The system was based on a Mecanum rover and equipped with a 2D-LiDAR (UST-20LX) and an onboard computer (Jetson AGX Orin) for processing. The software system was built using ROS 2 Humble, and pedestrian detection was performed using LFE-Peaks [39]. In addition, localization was realized using adaptive Monte Carlo localization (AMCL). To process the pedestrian detection and proposed method, we used a computer equipped with an AMD Ryzen 9 7900 CPU and NVIDIA GeForce RTX 4090 GPU to distribute the computational load.

A. Circle Crossing Scenario

We verified whether the robot could successfully execute the circle crossing scenario used in the simulation evaluation. Fig. 6 shows the actual demonstration of the robot navigating through pedestrians while reaching its destination. This confirms that the proposed method can be effectively applied, even when operating a physical robot in a real-world environment.

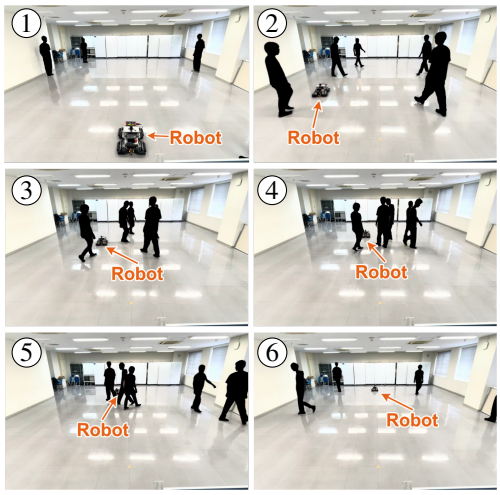

Fig. 6. Scenes of the real-world demonstration using the proposed method in the circle crossing scenario.

B. Corridor Scenario

We experimentally verified whether the guidance method for static obstacle avoidance could be successfully implemented on an actual robot in a corridor environment. The width of the corridor was approximately 2 m, and one pedestrian in the environment had to be avoided. Fig. 7 shows the results without and with guidance. Without guidance, the robot collided with the wall. By contrast, with guidance, the robot successfully avoided both pedestrians and static obstacles. These results confirm that the proposed guidance method for static obstacle avoidance can effectively guide action generation not only for pedestrians but also for static obstacles when deploying a physical robot in real-world environments.

C. Companion Scenario

We evaluated whether the guidance mechanism for companion tasks could successfully generate the appropriate actions in a real-world companion scenario. Two pedestrians were placed in the environment: one was the target pedestrian and the other was the pedestrian to be avoided. Fig. 8 shows the demonstration results. When operating without guidance, the robot simply avoided the pedestrians, whereas with guidance, it successfully generated a behavior that accompanied the target pedestrian. Notably, in the second result with guidance, the pedestrian moved significantly toward the right side of the environment, and the corresponding robot exhibited substantial changes in its behavior. These results confirm that our guidance mechanism for companion tasks can successfully direct behavior generation on a actual robot in a real-world environment, thereby effectively guiding the robot to accompany specific pedestrians.

VII. CONCLUSIONS

In this study, we used diffusion-based reinforcement learning with the QSM-based method for social navigation and showed that it outperformed other methods. In addition, we demonstrated that, guidance enables the incorporation of

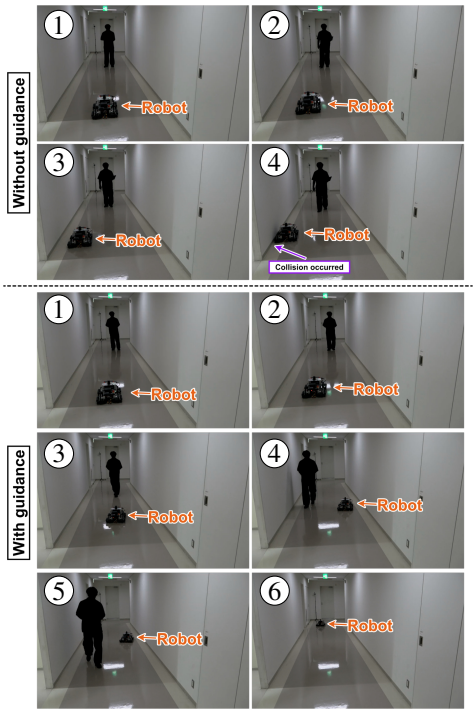

Fig. 7. Scenes of the real-world demonstration using the proposed method in the corridor scenario.

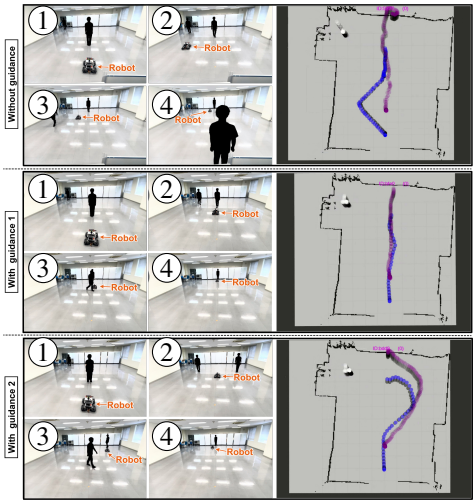

Fig. 8. Scenes of the real-world demonstration using the proposed method in the companion scenario. In the images on the right of each row, the blue markers indicate trajectories of the robot, and the purple markers indicate trajectories of the target pedestrian.

conditions not considered during training without additional training.

In future work, we plan to extend the applicability of the proposed method to a wider range of scenarios and conduct more detailed real-world experiments to further improve its performance.

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] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] K. Matsumoto, Y. Hyodo, and R. Kurazume, "Crowd-Aware Robot Navigation with Switching Between Learning-Based and Rule-Based Methods Using Normalizing Flows," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4823–4830, 2024.

[15] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6840–6851, 2020.

[16] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2022.

[17] K. Nakashima and R. Kurazume, "LiDAR Data Synthesis with Denoising Diffusion Probabilistic Models," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14724–14731, 2024.

[18] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with Diffusion for Flexible Behavior Synthesis," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 9902–9915, 2022.

[19] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal-Conditioned Imitation Learning using score-based Diffusion Policies," in *Proceedings of the Robotics: Science and Systems (RSS)*, 2023.

[20] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion," in *Proceedings of the Robotics: Science and Systems (RSS)*, 2023.

[21] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[22] H. J. Terry Suh, G. Chou, H. Dai, L. Yang, A. Gupta, and R. Tedrake, "Fighting Uncertainty with Gradients: Offline Reinforcement Learning via Diffusion Score Matching," in *Proceedings of the Annual Conference on Robot Learning (CoRL)*, pp. 2878–2904, 2023.

[23] B. Kang, X. Ma, C. Du, T. Pang, and Y. A. N. Shuicheng, "Efficient Diffusion Policies For Offline Reinforcement Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 67195–67212, 2023.

[24] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine, "IDQL: Implicit Q-Learning as an Actor-Critic Method with Diffusion Policies," *CoRR*, vol. abs/2304.10573, 2023.

[25] L. Yang, Z. Huang, F. Lei, Y. Zhong, Y. Yang, C. Fang, S. Wen, B. Zhou, and Z. Lin, "Policy Representation via Diffusion Probability Model for Reinforcement Learning," *CoRR*, vol. abs/2305.13122, 2023.

[26] M. Psenka, A. Escontrela, P. Abbeel, and Y. Ma, "Learning a Diffusion Model Policy from Rewards via Q-Score Matching," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 41163–41182, 2024.

[27] S. Ding, K. Hu, Z. Zhang, K. Ren, W. Zhang, J. Yu, J. Wang, and Y. Shi, "Diffusion-based Reinforcement Learning via Q-weighted Variational Policy Optimization," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 53945–53968, 2024.

[28] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70, 2024.

[29] J. Liu, M. Stamatopoulou, and D. Kanoulas, "DiPPeR: Diffusion-based 2D Path Planner applied on Legged Robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9264–9270, 2024.

[30] M. Stamatopoulou, J. Liu, and D. Kanoulas, "DiPPeST: Diffusion-based path planner for synthesizing trajectories applied on quadruped robots," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7787–7793, 2024.

[31] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is Conditional Generative Modeling all you need for Decision Making?," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[32] D. Rempe, Z. Luo, X. B. Peng, Y. Yuan, K. Kitani, K. Kreis, S. Fidler, and O. Litany, "Trace and Pace: Controllable Pedestrian Animation via Guided Trajectory Diffusion," in *Proceedings of theIEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13756–13766, IEEE, 2023.

[33] Y. Zeng, H. Ren, S. Wang, J. Huang, and H. Cheng, "NaviDiffusor: Cost-Guided Diffusion Model for Visual Navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11994–12001, IEEE, 2025.

[34] 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.

[35] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning," *CoRR*, vol. abs/1910.00177, 2019.

[36] P. Kozakowski, L. Kaiser, H. Michalewski, A. Mohiuddin, and K. Kańska, "Q-Value Weighted Regression: Reinforcement Learning with Limited Data," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2022.

[37] A. Nair, M. Dalal, A. Gupta, and S. Levine, "AWAC: Accelerating Online Reinforcement Learning with Offline Datasets," *CoRR*, vol. abs/2006.09359, 2020.

[38] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review. E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[39] F. Amodeo, N. Pérez-Higueras, L. Merino, and F. Caballero, "FROG: a new people detection dataset for knee-high 2D range finders," *Frontiers in Robotics and AI*, vol. 12, p. 1671673, Oct. 2025.