2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Fast Action Generation via Knowledge Distillation with Flow Matching for Social Navigation

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

***Abstract*— Mobile robot navigation in dynamic environments that contain pedestrians is one of the key challenges in the development of autonomous mobile service robots. This field, known as social navigation, has seen significant research progress using reinforcement learning approaches. In recent years, numerous diffusion-based reinforcement learning methods capable of generating diverse actions have been proposed. However, compared to conventional reinforcement learning approaches, the diffusion model's slow generation process presents a significant barrier to real-time processing. To address this, we propose a method for knowledge distillation of conditional diffusion models by combining Gaussian Prior with Flow Matching to enable faster action generation in dynamic environments. Experiments using a crowd navigation benchmark in simulation environments demonstrate that a significant reduction of the time required for action generation is possible while maintaining nearly the same performance as teacher models.**

I. INTRODUCTION

In recent years, the adoption of mobile robots has seen a marked increase, driven by labor shortages in a variety of sectors. In order to successfully deploy mobile robots in environments characterized by high pedestrian traffic, such as hospitals and shopping malls, it is imperative to realize methods that facilitate the smooth navigation of robots while ensuring the avoidance of collisions in such crowd environments.

This type of robot navigation research is known as social navigation, and approaches utilizing deep reinforcement learning have been proposed [1]–[3]. Deep reinforcement learning involves operating a model in the target environment while collecting data. To facilitate efficient learning, diverse experiences are necessary, and it is desirable to generate actions with high expected reward pre-configured in the environment. However, conventional action generation methods using simple probability distributions like normal distributions struggle to produce both diverse and high value actions, and require substantial time to gather sufficient data.

In contrast, diffusion models [4], which have recently gained attention for their applications in generation tasks, can generate diverse data by learning an inverse diffusion process that reconstructs noise-corrupted data. When applied to online reinforcement learning, these models have been shown to significantly improve performance [5]. However, this performance comes at the cost of requiring a large number of incremental denoising steps, which results in a significant computational burden. This limitation makes standard diffusion models less suitable for real-time applications, such as robotic control, where fast action sampling is essential at every timestep.

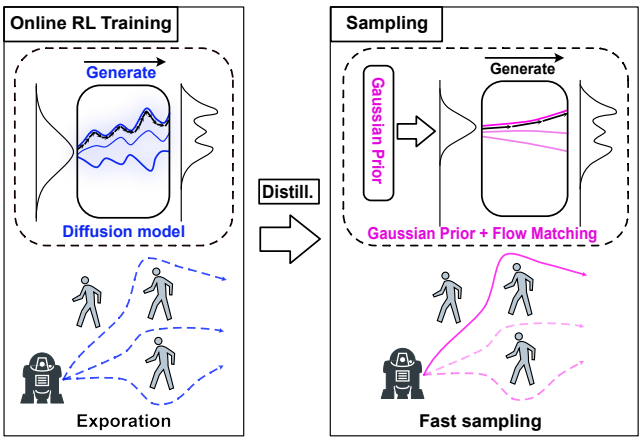


Fig. 1: Fast Action Generation using Flow Matching and a Gaussian Prior. The left part of the diagram illustrates a diffusion model that captures a multimodal distribution over trajectories, enabling diverse and collision-free path exploration. In contrast, the right part shows our proposed approach, where Flow Matching is used to learn a vector field that maps a Gaussian Prior, trained on samples from the teacher model, to the teacher's policy distribution. By generating directly from a Gaussian Prior trained that, our method can skip many intermediate steps, significantly accelerating action sampling.

In this study, we propose a distillation method for diffusion models that leverages Flow Matching to learn a vector field from a Gaussian distribution trained via likelihood-based imitation learning to the teacher policy distribution. This reduces the number of reverse diffusion steps and enables faster action sampling without compromising performance. Fig. 1 illustrates the core components of our proposed approach. It contrasts the conventional multi-step denoising process in diffusion models trained via online reinforcement learning with our accelerated framework, which combines Flow Matching and imitation learning to train a Gaussian Prior. By learning a direct mapping from the Gaussian Prior to the teacher policy distribution, our method significantly

[1]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

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

[†]Authors contributed equally

reduces the number of steps required for action generation. This makes it particularly suitable for real-time robotic control in dynamic, human-populated environments.

The main contributions of this work are as follows:

- A distillation framework for diffusion models using Gaussian Prior and Flow Matching.
- Demonstration of fast action generation without loss of performance.
- Comprehensive evaluation in simulated social navigation tasks.

II. RELATED WORK

A. Social Navigation with Reinforcement Learning

Social navigation methods based on deep Reinforcement Learning (RL) have been actively studied in recent years. Chen et al. [1] designed a reinforcement learning model capable of real-time decision-making in changing environments by considering both destination information and the relative spatial relationships between other pedestrians. Furthermore, Chen et al. [2] proposed a graph-based RL framework that explicitly models pedestrian interactions, enabling robots to navigate smoothly and safely through dynamic crowds. This allows the model to flexibly update its action strategy even in dense crowds, significantly improving its collision avoidance capabilities. Additionally, Zhang et al. [3] introduced a learning approach that accommodates continuous action spaces, eliminating the constraints imposed by discrete action selection to enable smoother, more human-like path generation. These methods prove effective for enhancing adaptability and flexibility in complex crowd environments.Furthermore, Matsumoto et al. [6] achieved both safety and efficiency by flexibly switching between learning-based and rule-based approaches based on the likelihood of the current state.

B. Diffusion Model for Action Generation

Diffusion models have recently emerged as powerful generative models, originally introduced in the context of image synthesis [7]. Owing to their ability to model complex data distributions and generate diverse outputs, diffusion models have been successfully applied in various domains beyond images, including planning and decision-making tasks [8], [9]. More recently, diffusion models have also been adopted in the context of robot navigation and exploration. Sridhar et al. [10] introduced a goal-conditioned masked diffusion policy, NoMaD, that enables long-horizon navigation and exploration in novel environments. Xiong et al. [11] applied diffusion-based trajectory synthesis to quadruped robots in DiPPeST, producing feasible and dynamic motion plans without reliance on explicit planners. Huang et al. [12] further incorporated cost-based guidance into the diffusion process in NaviDiffusor, allowing for efficient visual navigation by biasing sampling toward low-cost trajectories. Tomita et al. [5] proposed COLSON, a robot navigation system that efficiently avoids pedestrians by training diffusion models using online reinforcement learning. This approach demonstrates high success rates even in environments where pedestrian counts and initial configurations significantly differ from the training environment. However, when implementing this system in real-world environments, the time required to sense the environment and generate a single action is significantly slower than the 4 fps of the training environment, indicating the need for speed optimization for real-world deployment. These applications demonstrate the flexibility of diffusion models in handling high-dimensional, multimodal trajectory generation for real-world navigation scenarios.

C. Distillation and Acceleration Techniques

Despite their strong generative capabilities, diffusion models typically require hundreds of iterative denoising steps, which makes them computationally expensive and limits their real-time applicability, particularly in robotics and navigation scenarios within complex crowd environments. To address this limitation, several acceleration and distillation techniques have been proposed.

Distillation-based approaches, such as Salimans et al. [13] introduced Progressive Distillation, reduce generation steps by training a student model to mimic the outputs of a full-step teacher model in significantly fewer iterations. This process is performed recursively, where each student learns from a teacher that uses half as many denoising steps. By repeating this half way strategy, the final student model is able to generate high quality samples in just four steps, making it well suited for real-time applications.

Song et al. [14] introduced Consistency models provide an alternative acceleration mechanism by learning a self-consistent mapping between noisy and clean samples in a few time steps. Rather than iteratively denoising, these models directly learn to produce clean outputs from noisy inputs in a single step, greatly improving generation speed. Recent works have further extended these ideas to action generation tasks. In particular, consistency model applies consistency distillation to visuomotor policies, accelerating generation by training few steps student policy from a multistep diffusion teacher [15].

In parallel, Liu et al. [16] introduced Rectified Flow, which builds upon the Flow Matching framework [17]. Unlike score-based diffusion models that learn a reverse-time score function, Rectified Flow directly models a continuous vector field that transports noise to data through the integration of an ordinary differential equation (ODE). Flow Matching formulates generative modeling as learning a time-dependent velocity field that maps one distribution to another. Rectified Flow builds upon this formulation by directly learning such a vector field for deterministic sampling. To further improve efficiency and accuracy, Rectified Flow introduces a Reflow procedure that iteratively refines the learned vector field by retraining on samples generated from the current flow. This iterative flow matching process uses updated noise–data pairs, gradually aligning the learned trajectories with more efficient transport paths from source to target distributions. It has been observed to produce increasingly straight transport trajectories, which reduces discretization error and enables

accurate generation with fewer integration steps. This reduces discretization error during ODE integration and enables high-quality sample generation with significantly fewer steps. Furthermore, in Flow Matching which learns vector fields between different distributions it is possible to employ the desired distribution as the prior, while Rectified Flow is particularly useful for tasks involving transformations between datasets, such as converting images of cats to dogs.

In this study, we focus on the aspect of Flow Matching that learns vector fields between distributions. We develop a method that learns the transformation from a gaussian distribution trained via likelihood-based imitation learning to the distribution of the teacher model. This enables faster action generation while maintaining performance equivalent to that of the teacher model in a diffusion model-based reinforcement learning framework for social navigation.

III. PRELIMINARIES

A. Problem settings

This study considers a task where a robot navigates through an environment containing multiple pedestrians in the X-Y plane while avoiding collisions. We define the reward, state, and action as follows:

- State: The state consists of position and velocity data for both pedestrians and the robot. Each pedestrian and robot observation is represented by a vector $(p_x^i, p_y^i, v_x^i, v_y^i)$, where for i-th pedestrian or robot, (p_x^i, p_y^i) denotes the position.
- Action: Assuming a holonomic omnidirectional robot, we use a 2D vector (v_x, v_y) consisting of the x-axis input velocity v_x and y-axis input velocity v_y for the robot's movement in 2D space.
- Reward: The reward function for this study is specified in Equation (1). Here, d_t represents the minimum distance between the robot and surrounding pedestrians, $\boldsymbol{p}_t^r$ denotes the robot's position at time t, and $\boldsymbol{p}_g$ indicates the robot's target position.

$$R(\boldsymbol{s}_t) = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \boldsymbol{p}_t^r = \boldsymbol{p}_g \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

B. Diffusion models formulation

Diffusion models consist of two key processes: a forward diffusion process and a reverse diffusion process. In the forward process, a sample $\boldsymbol{x}_0$ drawn from the target data distribution is gradually corrupted by Gaussian noise over a series of discrete time steps, ultimately transforming the sample into pure noise $\boldsymbol{x}_T$. This process is defined as:

$$\boldsymbol{x}_\tau = \sqrt{\alpha_\tau}\boldsymbol{x}_{\tau-1} + \sqrt{1-\alpha_\tau}\boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad (2)$$

where α_τ is a predefined noise schedule that determines the variance at each time step τ, and $\tau \in \{1, 2, \ldots, T\}$ denotes a discrete time index. In the reverse process, a neural network $\boldsymbol{\epsilon}_\theta$ is trained to denoise the sample by predicting the added noise at each time step. The generation process involves starting from pure noise $\boldsymbol{x}_T$ and iteratively applying the denoising steps to obtain a clean sample $\boldsymbol{x}_0$:

$$\boldsymbol{x}_{\tau-1} = \frac{\boldsymbol{x}_\tau - \sqrt{1-\alpha_\tau}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_\tau, \tau)}{\sqrt{\alpha_\tau}}. \quad (3)$$

The model is trained to minimize the difference between the predicted noise and the true noise used during the forward process, commonly via a simple mean squared error loss:

$$L(\theta) = \mathbb{E}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\alpha\tau}\boldsymbol{x}_0 + \sqrt{1-\alpha_\tau}\boldsymbol{\epsilon}, \tau)\right\|^2\right]. \quad (4)$$

Through this training objective, the model learns a time-dependent denoising function that enables the generation of high-quality samples from random noise.

IV. APPROACH

A. Pre-trained diffusion model

We employ Q-Score Matching (QSM) [18] as our policy model, which iteratively optimizes the diffusion model's score function to approximate the action value function. We utilized a pre-trained model from 100,000 episodes of online reinforcement learning [5]. The policy model is trained using pedestrian positions relative to the robot's location and the goal position as the state $\boldsymbol{s}_t$, with the objective of maximizing the reward $R(\boldsymbol{s}_t)$. The loss function for Q-Score Matching is specified in Equation (5).

$$L_{\text{QSM}} = \mathbb{E}[\|\Psi_\theta(\boldsymbol{s}_t, \boldsymbol{x}_\tau, \tau) - \frac{1}{\beta}\nabla_{\boldsymbol{x}_\tau} Q(\boldsymbol{s}_t, \boldsymbol{x}_\tau)\|^2]. \quad (5)$$

Here, $\boldsymbol{x}_\tau$ represents the action $\boldsymbol{a}_t$ in the simulation environment perturbed by diffusion step τ noise. Through this training approach, the policy probability distribution π_θ follows Equation (6).

$$\pi_\theta(\boldsymbol{s}_t) \propto \exp\left(\frac{1}{\beta}Q(\boldsymbol{s}_t, \boldsymbol{a}_t)\right). \quad (6)$$

In this formulation, $Q(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is trained to estimate the expected reward when taking action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$, and can take various values even for the same state $\boldsymbol{s}_t$. The distribution $\pi_\theta(\boldsymbol{s}_t)$ is proportional to the exponential of the action-value function, enabling representation of multimodal behavior. The parameter β, referred to as the temperature coefficient, controls the distribution's smoothness: increasing β produces a smoother distribution with higher entropy, while decreasing β results in a more peaked distribution with lower entropy.

B. Distillation Using Gaussian Prior and Flow Matching

In this study, we propose a method that utilizes a pre-trained diffusion model through online reinforcement learning, employing both Flow Matching and Gaussian Prior for distillation. First, we collect diverse states in a simulated environment and train each state to maximize the likelihood $\log \boldsymbol{p}_\psi(\boldsymbol{s}, \hat{\boldsymbol{a}})$ of the action $\hat{\boldsymbol{a}}$ sampled from the teacher model with respect to its Gaussian Prior $\boldsymbol{p}_\psi$. Subsequently, Flow model train the vector field as $v = \hat{\boldsymbol{a}} - \boldsymbol{x}_T$. where $\boldsymbol{x}_T$ is data points sampled from Gaussian Prior and $\hat{\boldsymbol{a}}$ is

sample of the diffusion model. This effectively replaces the iterative denoising trajectory with a learned direct transport path from the Gaussian Prior to the target distribution. The distillation procedure is presented in Algorithm 1. Although this method does not offer a theoretical guarantee of faster action generation, it enables practical speedup by learning a direct mapping from the trained Gaussian Prior to the teacher policy distribution. This bypasses the iterative denoising process commonly required in diffusion models. Since the prior is optimized to approximate the output distribution of the teacher, the learned vector field provides an efficient trajectory for action generation with significantly fewer steps.

Algorithm 1 Distillation of Diffusion Model via Flow Matching from Gaussian Prior

Require: Pre-trained diffusion policy π_θ, environment $\mathcal{E}$
Ensure: Teacher $\boldsymbol{\pi}_\theta$, Student $\boldsymbol{v}_\phi$, Gaussian Prior $\boldsymbol{p}_\psi$
Prior training
for each iteration **do**
  Sample state $\boldsymbol{s} \sim \mathcal{E}$
  Generate action $\hat{\boldsymbol{a}} \sim \pi_\theta(\boldsymbol{s})$
  Update $\boldsymbol{p}_\psi$ by minimizing: $\mathbb{E}[-\log \boldsymbol{p}_\psi(\boldsymbol{s}, \hat{\boldsymbol{a}})]$
end for
Flow Training
for each iteration **do**
  Sample state $\boldsymbol{s} \sim \mathcal{E}$
  Sample from Gaussian prior $\boldsymbol{x}_T \sim \boldsymbol{p}_\psi(\boldsymbol{s})$
  Generate action from $\boldsymbol{x}_T$ $\hat{\boldsymbol{a}} \leftarrow \pi_\theta(\boldsymbol{x}_T, \boldsymbol{s})$
  Compute interpolated input: $\boldsymbol{x}_t \leftarrow \frac{T-t}{T}\hat{\boldsymbol{a}} + \frac{t}{T}\boldsymbol{x}_T$
  Compute flow vector: $\boldsymbol{v} \leftarrow \hat{\boldsymbol{a}} - \boldsymbol{x}_T$
  Update $\boldsymbol{v}_\phi$ by minimizing: $\mathbb{E}\left[\|\boldsymbol{v} - \boldsymbol{v}_\phi(\boldsymbol{x}_t, t, \boldsymbol{s})\|^2\right]$
end for

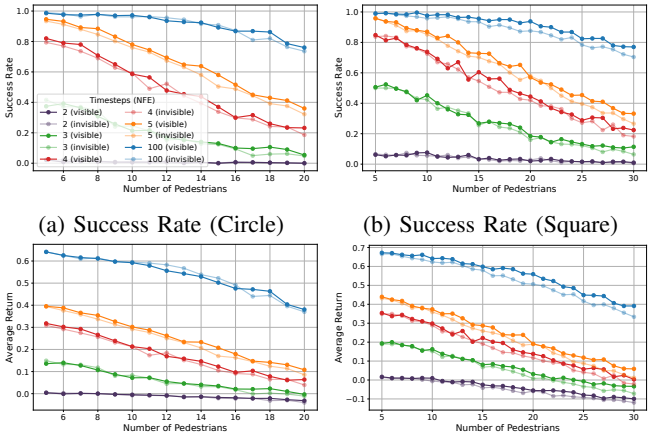
V. Experiment

A. Simulation environment and settings

In the simulation experiments, we utilize the circle crossing scenario from the CrowdNav environment [1], [2]. In this scenario, a robot navigates from an initial position $(x, y) = (0, -4)$ toward a goal location $(x, y) = (0, 4)$. Pedestrians behave according to ORCA [19] and are randomly distributed within a circle of radius $4\mathrm{m}$ at initialization. In this environment, pedestrians are controlled by ORCA [19]. The diffusion models were trained for 100,000 episodes in a visible circle crossing scenario with 5 pedestrians previously. Additionally, data were collected using ORCA before each training session, and training began with a dataset of 2000 episodes. For evaluation, in addition to the circle-crossing scenario, we implement a square-crossing scenario where pedestrians are randomly positioned within a square area of side length $20\mathrm{m}$, as well as invisibility scenarios in which the robot cannot detect pedestrians. The number of pedestrians in the circle-crossing and square-crossing scenarios is varied from 5 to 20 and 5 to 30, respectively. For each distillation method, we train the student models for 100,000 iterations and evaluate their performance with only three denoising steps, examining how performance changes under both visible and invisible pedestrian conditions.

B. Evaluation for the pre-trained model

Compared to our previous approach, the model trained in the simulation environment achieves 4 FPS, while when executed on a real robot using an inference device like Jetson, the maximum and average time from environment sensing to generating a single action are 4.1 s and 2.8 s respectively - significantly slower than the training environment, indicating that performance acceleration is essential for real-world deployment [5]. To address this, we first conduct preliminary experiments to examine how performance changes when reducing the NFE (Number of Function Emissions) of the pretrained diffusion model. Fig. 2 we compare the success rates and average rewards for pedestrian scenarios with 20 and 30 pedestrians when varying the NFE settings to 2, 3, 4, 5, and 100. Note that in the "invisible" configuration where the robot cannot see pedestrians, the results are shown with light-colored lines.



(a) Success Rate (Circle) (b) Success Rate (Square)

(c) Average Return (Circle) (d) Average Return (Square)

Fig. 2: Success Rate and Average Return each NFEs number with circle and square crossing.

From the above results, we observe that the diffusion model pretrained using QSM achieves sufficient pedestrian avoidance capability in NFEs-100, demonstrating strong generalization performance across different scenarios and pedestrian counts. However, omitting the NFEs with conventional ODE leads to performance degradation, indicating the difficulty in simultaneously achieving both fast action generation and high success rates.

C. Performance Evaluation

In this section, we conducted comparative experiments between our method and other distillation approaches, detailing the results. Here, both the Consistency model and our proposed method employed a three-step generation process. Fig. 3 shows the movement trajectories of the teacher model and our method versus other approaches. Overall, Fig. 3 demonstrates that our method follows the path most closely resembling the teacher model. Notably, in the second-to-last scenario from the right, the teacher model takes a

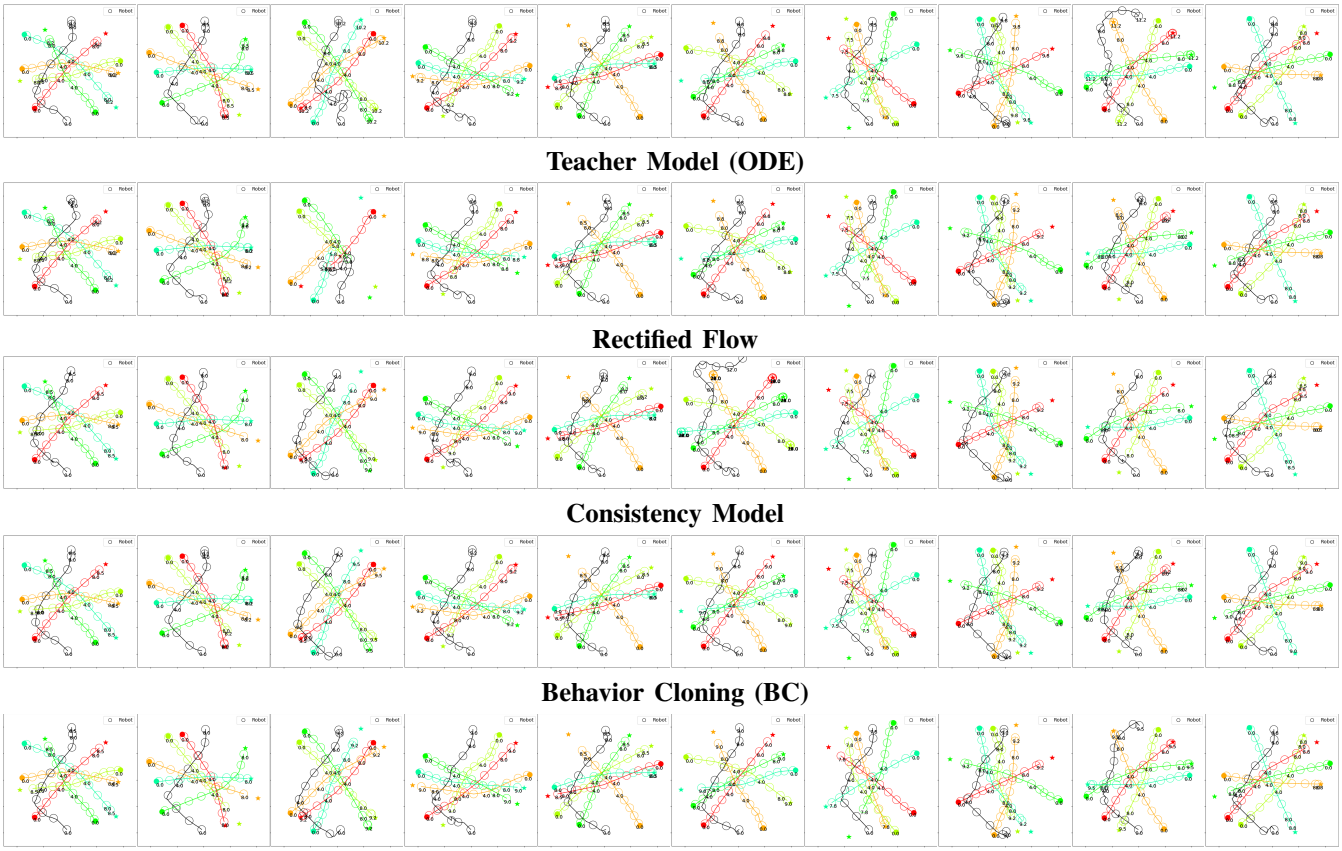
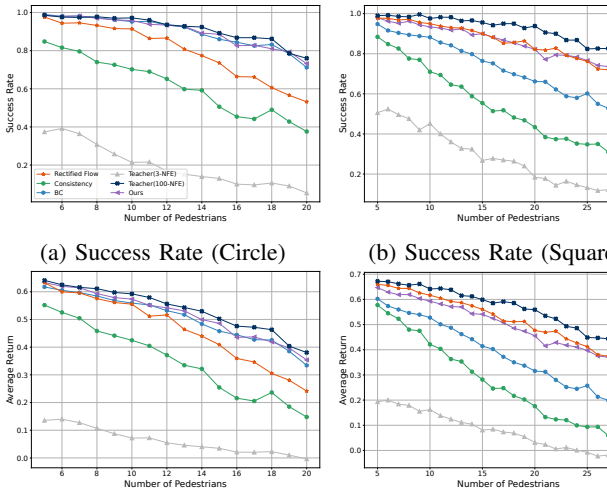


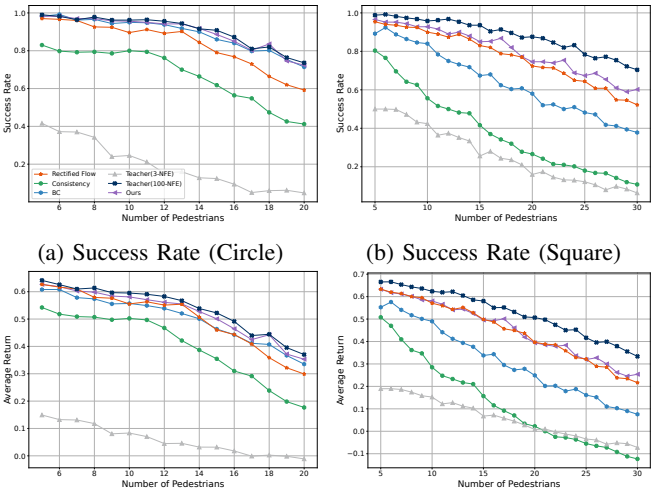
Fig. 3: Comparison of ODE, Rectified Flow, Consistency Model, BC, and Ours evaluations (10 samples each).



(a) Success Rate (Circle) (b) Success Rate (Square)

(c) Average Return (Circle) (d) Average Return (Square)

Fig. 4: Comparison of Success Rate and Average Return for circle and square crossing.



(a) Success Rate (Circle) (b) Success Rate (Square)

(c) Average Return (Circle) (d) Average Return (Square)

Fig. 5: Comparison of Success Rate and Average Return for circle and square crossing with invisible setting.

detour before reaching the goal, a characteristic that only our proposed method successfully captures.

Fig. 4 shows results for conventional circle-crossing and square-crossing scenarios with pedestrian counts varying from 5 to 20, while Fig. 5 compares circle-crossing and square-crossing scenarios in an invisible configuration where pedestrians are invisible to the robot, again with pedestrian counts varying from 5 to 20 and 5 to 30. The figures also present a comparison of success rates and average return when adjusting the Non-BC NFEs to a fixed value of 3.

First, from Fig. 4, we observe that in the Circle-crossing scenario - the training environment where the robot can see

pedestrians - both the Baseline and our proposed method achieve the highest success rates and average rewards among the models trained on the teacher data. Furthermore, in the Square-crossing scenario (which differs from the training environment), both the Rectified Flow and our proposed method again demonstrate the highest success rates and average rewards among the models. These results indicate that when considering all performance metrics across both circle and square crossing scenarios with pedestrian visibility, our proposed method achieves the closest performance to the teacher model.

Next, analyzing Fig. 5, we find that the proposed method achieves the highest similarity to the teacher model in both success rates and average rewards across all scenarios when the robot is invisible to pedestrians. These results clearly show that our proposed method can maintain performance comparable to the teacher model while significantly accelerating behavior generation, even when operating in scenarios different from the training environment.

VI. Real-world demonstration

A real-world demonstration was conducted using a developed robot system. The system is based on a Mecanum rover and is equipped with a 2D-LiDAR (UST-20LX) and onboard computer (Jetson AGX Orin) for processing. The software system was built using ROS2 Humble, and pedestrian detection was performed using DR-SPAAM [20] Additionally, localization was realized using adaptive Monte Carlo localization (AMCL).

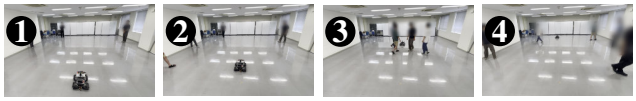


Fig. 6: Scenes of the real-world demonstration using the proposed method in the circle crossing scenario.

The demonstration is shown in Fig. 6. The result shows that the proposed method is capable of controlling the real robot. Regarding action generation time, the conventional method, which generates actions over 100 steps, required an average of 0.28 s with a maximum of 0.41 s on the robot system. In contrast, the proposed method achieved an average of 0.011 s and a maximum of 0.026 s, making it significantly faster and sufficiently fast even for the 0.25 s interval used in the training environment.

VII. CONCLUSIONS

This study achieves faster action generation by employing Flow Matching and Gaussian Prior for knowledge distillation on pretrained diffusion models. This combination successfully reduces the number of generation steps while maintaining both success rate and average reward compared to alternative methods. Future work will extend the applicability of our proposed method to broader scenarios and conduct detailed real-world experiments to further enhance its performance.

References

[1] 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.

[2] 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.

[3] 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.

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6840–6851, 2020.

[5] Y. Tomita, K. Matsumoto, Y. Hyodo, and R. Kurazume, "COLSON: controllable learning-based social navigation via diffusion-based reinforcement learning," *CoRR*, vol. abs/2503.13934, 2025.

[6] 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.

[7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[8] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proceedings of the International Conference on Machine Learnin (ICML)*, 2022.

[9] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[10] 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.

[11] Z. Xiong, T. Liu, Z. He, and F. Gao, "Dippest: Diffusion-based path planner for synthesizing trajectories applied on quadruped robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7787–7793, 2023.

[12] X. Huang, L. Zhang, C. Finn, and S. Levine, "Navidiffusor: Cost-guided diffusion model for visual navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11994–12001, 2025.

[13] T. Salimans and J. Ho, "Progressive distillation for fast sampling of diffusion models," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[14] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *Proceedings of the International Conference on Machine Learnin (ICML)*, pp. 32211–32252, 2023.

[15] M. Sarch, A. Jain, Y. Lu, Y. Chebotar, K. Hausman, and S. Levine, "Consistency policy: Accelerated visuomotor policies via consistency distillation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[16] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[17] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[18] 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 Learnin (ICML)*, pp. 41163–41182, 2024.

[19] 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.

[20] 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.