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

Advancement of Action Models through Model Circulation among Robots and Model Builders in the Robot AI Ecosystem

Yu Sakai[1] and Kazuyuki Morioka[2]

Abstract— **Recently, AI-based action models are expected to be applied to the robot systems. This study proposes the robot AI ecosystem for AI-based action models. This extends the platform sharing action models to the model circulation system through sharing demonstration data and model building. Specifically, we will focus on building advanced models through model circulation among the demonstration data of the robot behaviors and model training by the model builders in the ecosystem. Experiments in two scenarios to build the new generation action models through model circulation were performed to demonstrate the advancement of the action models. The results of experiments confirmed that open collaboration among the robots and the model builders in the ecosystem is effective for creating the advanced action models.**

I. Introduction

Autonomous mobile robots have been expected to be used in our daily life for a long time. Currently, the mobile robots can be found in the limitted situations. Also, the mobile robots based on the traditional map-based strict control method have difficulty operating flexibly in dynamic environments.

Some of the mobile robots controlled by AI-based action models created by machine learning have been proposed to achieve flexible navigation recently. We have also focused on autonomous mobile robots based on action models. And, we believe that this AI-based style of autonomous mobile robots will become more common in the future and contribute to the widespread mobile robots. In anticipation of such a future robot society, the authors have proposed a platform for sharing the AI-based action models in autonomous navigation tasks [1]. The propposed AI model sharing platform makes it possible to secure the ownership and value of the shared action models by utilizing the blockchain. Virtual currency payment for using the action model in the platform is also implemented. In other words, the mobile robots using the action models have to pay the virtual currency to the model owners. We assume that the model transaction with payment will be general in the future AI-based robotic society.

In this study, the AI model sharing platform is extended to the ecosystem of robot AI models. The robot AI ecosystem provides not only model sharing but also a model enhancement mechanism based on demonstration data sharing and model updating. This is a model circulation system through the use of the models by the robot, accumulation of robot navigation results as demonstration data and model updating by the model builders.

This paper mainly introduces the proposed robot AI ecosystem. Especially, the model update and enhancement through the model circulation are described in detail. Experimental results of model updating in the robot AI ecosystem are provided to represent the progressive development of the action models.

II. AI-Based Robot Systems

A. AI-based control in the robotics field

Recently, AI-based robot control has been applied to several studies and real-world implementations. For example, AI-based control of multiple UAVs [2] and picking operations with robot arms [3] can be found. In AI-based robotic systems, control outputs are generated from trained AI models. This feature is significantly different from the conventional robot systems based on strict theories and mathematical formulas.

The standard training methods of AI models are forward reinforcement learning or imitation learning. Each of these methods has its own characteristics. For example, the forward reinforcement learning doesn't need to obtain supervised data in advance for training. As AI models with the forward reinforcement learning are trained in simulation environments, the gap between simulation and the real environment must be addressed to transfer the models to the real-world robot application. On the other hand, the imitation learning requires many demonstration data from an actual robot for training. This is challenging because the robotic demonstration data cannot be obtained from the internet, making it difficult to collect enough data. Both methods are common in the viewpoint of orienting end-to-end control from sensor inputs to control outputs. In this paper, the AI model for robot control is called "action model" that generates actions as output from sensor input.

In the field of mobile robots, which is focused on in this study, various AI-based approaches have been studied. Tsuruta et al. achieved monocular camera-based autonomous navigation employing an action model trained by the forward deep reinforcement learning [4]. Yokoyama et al. performed tasks using multiple models obtained through reinforcement learning [5]. They used a four-legged robot equipped with arms. By combining models created for each function, they acheived a total to pick, move and place objects. Imitation learning-based approaches have also been studied. NoMaD[6] and ViNT[7], proposed by a research team at UC Berkeley, enable autonomous navigation to destinations

[1]Meiji Univ, Graduation School of Advanced Mathematical Sciences, Network Design Program, Japan email: cs243010@meiji.ac.jp
[2]Meiji Univ, School of Interdisciplinary Mathematical Sciences, Japan email: morioka@meiji.ac.jp

using only visual input. The action models in these studies are trained using many demonstration data including camera images and control outputs from actual mobile robots.

B. Previous projects for AI model sharing

We have developed an intelligence-sharing platform for sharing action models among many mobile robots distributed in the city environments [1]. Figure 1 is an overview of the intelligence-sharing platform that we proposed. This platform enables the mobile robots to share action models and to select appropriate models according to the situations while performing autonomous navigation tasks. The model transaction in the platform is based on blockchain technology. The action model owners are managed strictly in the blockchain-based platform. Then, the model owners can receive payment from mobile robots that use the models. We developed this system as the simulation including many mobile robots in the city environment. The results indicated that the economic platform for sharing action models can achieve mutual benefits for both robots and model owners.

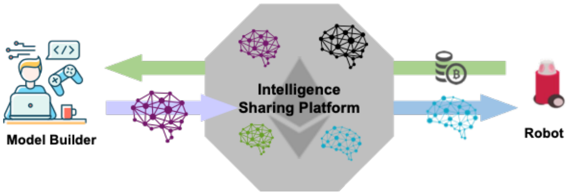

Fig. 1. Overview of the intelligence sharing platform

C. Robot AI ecosystem

Model sharing leads to advanced capabilities for model improvements through the updating processes. Related studies have been conducted to improve the models by collecting data and re-training. Müller et al. present OpenBot-Fleet, open-source system for collective robot learning [8]. The system gathers real-world data for a cloud-based policy improvement loop through teleoperation of the simple mobile robots by users. The system was validated at scale by successfully learning a general navigation policy. Dagdanov et al. propose a "self-improving" framework to enhance RL driving safety [9]. Their system uses black-box verification to find rare failure scenarios and iteratively re-trains the agent on these weaknesses, which significantly reduces collisions in simulation.

These advanced projects share the goal of improving robot action models through data circulation. This convergence of goals implies that our intelligence-sharing platform, introduced earlier, can be designed not only as a technical framework but also as an ecosystem. Then, this paper proposes the robot AI ecosystem including model and data circulation processes. Our ecosystem is different from the aforementioned systems in several key ways: it allows any robot and model builder to participate, it manages the value of both models and demonstration data, and it facilitates transactions for them. By implementing these mechanisms, we aim to establish a sustainable and collaborative marketplace for AI action models.

Figure 2 shows the overview of the robot AI ecosystem. Mobile robots in the ecosystem obtain appropriate action models obtained from the platform according to the situations. As shown in the driving trajectory at the bottom of Figure 2, the robot records demonstration data including the pairs of sensor information as the input and speed commands as the outputs while executing autonomous navigation tasks to the destination. Then, the robot stores demonstration data in the platform and the robot becomes the owner of the stored data. Next, model builders can utilize the stored demonstration data to develop new models with higher performance based on imitation learning. These new models are shared back onto the platform with the ownership of the model builders. By this cycle the robots can access the high performance models for task execution.

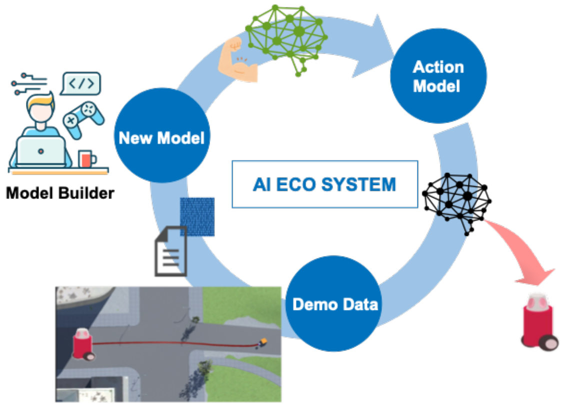

Fig. 2. Overview of the AI ecosystem

III. AI Ecosystem for Mobile Robots

A. System Overview

This study focuses on the concept of model circulation. This chapter describes the robot AI ecosystem for realizing model circulation in the mobile robotics field, especially the data-sharing system. When action models become commonplace, the main challenge will be how to create, maintain, and improve variety of models. The solution to this challenge involves utilizing the new demonstration data generated by robot activities through a model circulation loop formed in the robot AI ecosystem. In this loop, robots provide demonstration data, model builders can utilize the stored demonstration data to develop new generation models with higher performance, and those models are shared back onto the platform. Building this model circulation is the theme of this study, and the specific components will be described later. The AI ecosystem proposed in this study is an extension of the intelligence sharing platform we previously developed [1]. Intelligent sharing platform is extended as the generic data sharing system for storing demonstration data in addition to action models. Robots obtaining action models from this system execute their tasks while recording driving data, and upload the demonstration data to the data sharing system after the tasks are complete. Model builders can select and acquire the demonstration data and action models from this data sharing system, and they can create new models using acquired data. Furthermore, they can upload the new action models to the data sharing system. This allows robots to select the appropriate models from the various generation of action models.

B. Core components of robot AI ecosystem

Figure 3 shows the whole system configuration of the robot AI ecosystem, including the robots and model builders. At first, the data-sharing system that is the core component in the robot AI ecosystem is described.

1) Action Model Repository: This repository accumulates a variety of new action models. The robots download the action models from this repository. Also, model builders can upload new models to the repository. As the result, this repository includes the updated versions of existing models or completely different models created through training by the model builders. The actual action models are stored on a decentralized file system called IPFS, and the data-sharing system stores the hash value of the action model generated from the IPFS [1].
2) Demonstration Data Repository: The driving data while robot navigation is stored as the demonstrtion data in this repository. The demonstration data is also shared and used for creating new action models by model builders. The training of action models is performed in the local computers of model builders in the current system.

C. Model circulation in the robot AI ecosystem

Figure 3 also shows the model circulation among the robots, the model builders and the data-sharing system. **Action model user (robot):** The robot can acquire the appropriate action models from the data-sharing system according to the situation or tasks (A in Figure 3). The robot pays a defined amount of virtual currency every downloading. The obtained action model is deployed to the ROS node of the mobile robot automatially.

The mobile robots perform their tasks for traveling to destinations using downloaded action models. The robots record driving data to destinations. When the each task is complete, this data is uploaded as demonstration data to the data-sharing system (B in Figure 3). Subsequently, the robot is then registered as the owner of the uploaded data. **Action model owner (Model builder):** The action models are generated by model builders. Each model builder is registered as the owner of the corresponding action model. The proposed robot AI ecosystem is a circular system of the action models. The model builders can train new action models based on existing models and demonstration data obtained from the data-sharing system (C and E in Figure 3). The trained new action models are uploaded to the data-sharing system with the model characteristic information such as owner's Ethereum address and the model performance (D in Figure 3). The model owners can receive virtual currency as compensation from user robots according to the use of the action models.

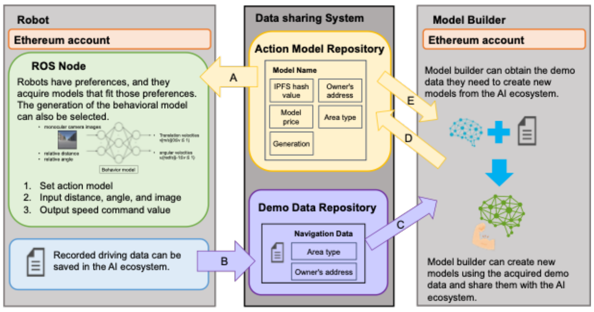

Fig. 3. Model circuration and core components

IV. Action Models For AI-based Navigation

This chapter describes the structure of the action model used in the proposed AI ecosystem. Additionally, we introduce methods for creating action models based on model circulation within the robot AI ecosystem. The action models introduced here are assumed to be utilized for the general differential two-wheeled mobile robots.

A. Action model for mobile robot navigation

Figure 4 shows the structure of the action model used in this study. This model has three types of inputs and two types of outputs. This model structure is the same as the method proposed by Tsuruta et al [4]. The three types of input states are as follows:

1) camera image (84 $\times$ 84 pixels) attached to the robot
2) relative distance between the robot and the destination
3) relative angle between the robot and the destination

Similarly, the two types of output actions are as follows:

1) linear velocity v[m/s] ($0 \leqq v \leqq 1$)
2) angular velocity ω[rad/s]($-1 \leqq \omega \leqq 1$)

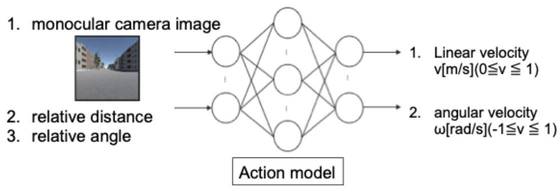

Fig. 4. Structure of the action model

There are two major advantages to autonomous navigation methods based on this model.

1) The robot configuration can be simplified. Currently, the popular system configurations of the mobile robots include multiple expensive sensors such as 3D-LiDARs. In contrast, this method achieves a simple robot configuration with only a monocular camera and rough localization system using the enough trained models.
2) The robots can perform fleixible navigation by switching appropriate action models according to the environments or situations.

B. Building action models in robotic AI ecosystem

We discuss the relationship between these action models and the AI ecosystem. The proposed system assumes that the action models are circulated in the robotic AI ecosystem as shown in Figure 3. In particular, model building within the framework of an AI ecosystem is described.

There are two ways for model builders to create action models within the AI ecosystem.

1) The model builder trains an action model through forward deep reinforcement learning in the simulator environment.
2) The model builder obtains navigation demonstration data from the AI ecosystem and updates the existing models using imitation learning and fine-tuning.

The action models are created using either of these methods and registered with the data-sharing system.

Next, we will explain how to train the above action models in detail. For forward reinforcement learning, we used the approximate policy optimization (PPO) [10] algorithm. The agents take actions with trial and error in the target environments, and obtain rewards for the actions in forward reinforcement learning. Through this process, the action models are trained. As the result, the agents can take the optimal actions according to the states.

In contrast, a combination of imitation learning and fine-tuning to update the model is especially important for model circulation in the robotic AI ecosystem. Generally, imitation learning uses state-action pairs as the ground-truth data from experts. By using demonstration data to imitate the behavior of experts, imitation learning can efficiently learn complex tasks. The Generative Adversarial Imitation Learning (GAIL) [11] is known as a learning method that improves on behavior cloning (BC). GAIL is a method that uses two neural networks: "Generator" and "Discriminator". BC is a basic imitation learning method. In this study, GAIL is utilized as the imitation learning method for updating the models.

TABLE I

REWARD SETTINGS

Rewards	Details
+250	Robot reaches the destination
-50	Robot collides with obstacles
± 0.1	Robot approaches or leaves the destination
-0.01	Angle to the destination is ± 90 degrees or more
-0.1	Angle to the destination is ± 150 degrees or more
-0.001	Each time the robot takes one step

In GAIL, the Discriminator outputs the probability that the inputs, including states and actions of the action model, are either the output generated by the Generator or the training data. Also, the Discriminator is trained to be able to distinguish differences between the output generated by the Generator and the training data. And the Generator is trained by PPO as same as the general forward deep reinforcement learning. In the training, the probability output from the Discriminator is treated as the reward. This means that the Generator is trained so that the Discriminator mistakenly recognizes the Generator's output as training data. In addition, this study utilizes the learning system extending GAIL. This learning system is capable of obtaining rewards from the environment similar to the general deep reinforcement learning, not only from the GAIL Discriminator. This means that the actions generated by the Generator are evaluated by both the Discriminator and the environment. This feature makes the action model possible to output appropriate actions even for states that were not included in the demonstration data, while referering the demonstration data. The reward setting for creating the model is shown in the Table I. In this study, all trainings were performed for 500,000 steps based on this reward setting. Finally, the Generator trained in the process described above is stored in the data sharing system as the action model by the imitation learning. In the next chapter, we will explain the details of the model update experiments in the proposed robotic AI ecosystem.

C. Demonstration data for model training

The demonstration data used in the current AI ecosystem is recorded during the robot navigation using the action models in the simulator environment. The demonstration data is composed of the pairs of the model inputs and the outputs while a series of the navigation. More precisely, each pair consists of the inputs (a camera image, and the relative distance and angle to the destination) and the outputs (linear velocity and angular velocity). As shown in Figure 5, a single demonstration data file stored in the data-sharing system contains multiple datasets based on the results in the several similar navigation tasks. In this study, each demonstration data file contains datasets obtained from approximately 25 navigation results.

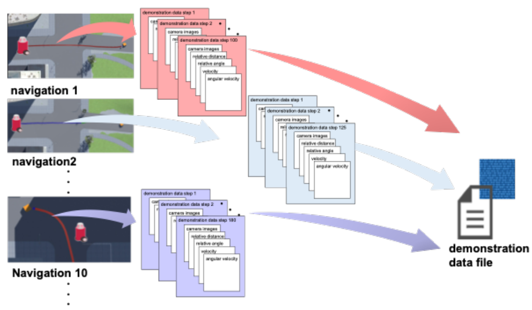

Fig. 5. Structure of the demonstration data file

V. EXPERIMENT: MODEL UPDATE USING DEMO DATA

This chapter outlines the process of creating next generation models through the model circulation in the AI ecosystem. We perform the experiments to update the action models in the following two scenarios:

A. Scenario 1
   Increasing navigation functions of the models.
   Initial model: only single direction movement
   Updated model: multiple direction movement
B. Scenario 2
   Expanding applicable areas of the models.
   Initial model: only specific area navigation
   Updated model: multiple area navigation

In the experiments, the robots shown in Figure 3 perform their behaviors using the action models stored in the data sharing system. The robots are controlled by ROS-based software, similar to the actual robots. These behaviors and training sessions are performed in a virtual city environment (Figure 6), which was developed using Unity. As shown in Figure 6, the city environment is divided into three distinct areas.

Fig. 6. Three Areas of Virtual Environment

A. Scenario 1

The goal of this scenario is to evolve from a simple model to increasingly complex models. The simple model has a function of only single direction movement. After the model circulation, the new models have functions of multiple direction movements. This scenario corresponds to C and D in Figure 3. In scenario 1, experiments on the model updating were conducted in the low building area in Figure 6. For model updating, model builders can download existing demonstration data already stored on a data sharing platform. Model builders can also use the simulator as model training environments.

At first, an initial action model trained only for straight-line movement is prepared by a model builder. We define this model as the first-generation model in the scenario 1. This initial model is generated using PPO as described in Section IV.B(1), not imitation learning. In the model training, the destination was placed in front of the robot initial position, repeating trial and error to the same destination in the simulator. The performance of this first-generation model is shown in Table II and Figure 7 that provide the results of left turn, straight and right turn. A characteristic of the first-generation model can be found as its high accuracy in straight-line travel. On the other hand, it doesn't have the ability to turn left or right. These features are as expected for the initial model. The first-generation model built in this training is stored in the data sharing system (D in Figure 3).

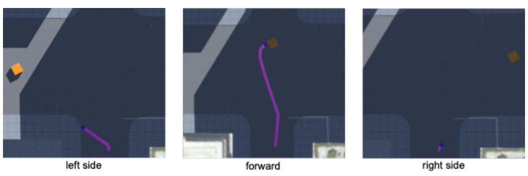

Fig. 7. Navigation path of the first-generation model

The second-generation model is created through a combination of imitation learning and fine-tuning, which utilizes both the demonstration data and the first-generation model. This means that the parameters of the first-generation model are re-trained by imitation learning utilizing demonstration data stored in the data-sharing system. The demonstration data used for this experiment includes sets of right turn movements obtained previously by the other existing model. The robot smoothly turned right during each navigation in this demonstration data. A model builder trains the second-generation model by downloading the initial model and the demonstration data from the data sharing system (C and E in Figure 3), and re-training by the imitation learning described in the above chapter.

The performance of the trained second-generation model is also shown in Table II and Figure 8. The results indicate the second-generation model allows the mobile robot to move to its destinations regardless of its lateral positions. However, the trajectories tend to be linear to the destinations. This feature is not desirable for smooth navigation, although the functions of models are extended from the first-generation model. The second-generation model built is also stored in the data sharing system (D in Figure 3) for the next use.

Fig. 8. Navigation path of the second-generation model

Following a similar process to the second-generation model, the third-generation model is also trained using the second-generation model and the other demonstration data. The demonstration data in this training is composed of left-turn movements to destinations on the left side. Similar to the demonstration data of the right turns, the robot smoothly turned left during each navigation in this demonstration data. The performance of the third-generation model is shown in Table II and Figure 9. The third-generation model can travel to any destination, regardless of its directions. In addition, when turning right or left, the robot using this model can perform obvious smooth turning movements. These results indicate that the action models are extended through model circulation in the proposed robot AI ecosystem.

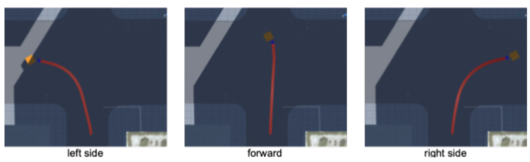

Fig. 9. Navigation path of the third-generation model

TABLE II
NAVIGATION PERFORMANCE OF ACTION MODELS IN EACH GENERATION

model generation	left	straight	right
first-generation	0%(0/50)	100%(50/50)	0%(0/50)
second-generation	100%(50/50)	100%(50/50)	100%(50/50)
third-generation	100%(50/50)	100%(50/50)	100%(50/50)

B. Scenario 2

The purpose of the scenario 2 experiment is to obtain an action model capable of navigation in multiple areas through model circulation. This scenario corresponds to all of flows A, B, C, D, and E in Figure 3. The action models corresponding to navigation in the respective areas in Figure 6 are prepared through training based on deep reinforcement learning. In detail, the model is trained so that the robot can travel smoothly to any destination in each target area through curriculum learning and random settings of start and goal pairs. These models are treated as first-generation models in the second scenario.

The robots perform the navigation tasks in each area using the corresponding first-generation model (A in Figure 3). The demonstration data in each area is created by the state-action pairs while the robot navigation tasks. In the two building areas, the robots travel to destinations along roads avoiding collisions with obstacles like buildings. In the park area, the robots travel to along the park passage avoiding grass areas. The start and goal positions of the navigation in each area are randomely provided. The second-generation model is created through re-training based on demonstration data and the first-generation model. As the result of re-training, the second-generation model acquires the ability to navigate two areas: the area initially trained and the area recorded in the demonstraton data used. The second-generation models are obtained by re-training the first-generation model using the imitation learning. This training process includes curriculum learning and random settings of start and goal pairs similar to the training of the first-generation models.

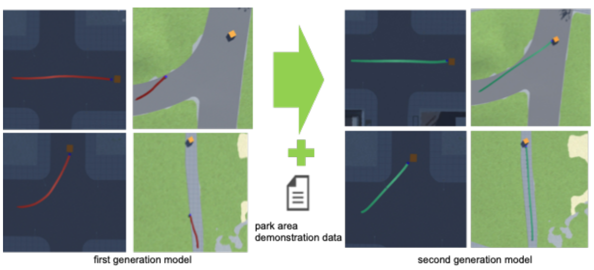

Fig. 10. Performance comparison between trained models

Figure 10 shows an example of the model performance comparison between the first-generation model and the second-generation model. The trajectories in the figure are navigation results in the park area and the high building area. The left side of the figure shows the trajectories of the robot navigation using the first-generation model for the high building area. The first-generation model was able to reach destinations in the high building area, but was unable to reach destinations in the park area. These features are as expected for the first-generation model of the high building area. In contrast, the right side of Figure 10 shows the trajectories of the robot navigation the second-generation model. The second-generation model is capable of reaching its destination even in the park areas. This is because the second-generation model obtained the navigation ability of the new areas through re-training using demonstration data. Naturally, the second-generation model is also capable of navigating areas that it trained on initially. The results indicate that the model circulation can enhance existing models through re-training using demonstration data.

VI. Conclusion and Future Work

This paper introduces the robot AI ecosystem extended from the previously proposed economic platform of action models. In particular, we detailed model updates through model circulation in the ecosystem. The new models can be created through the existing action models and the demonstration data obtained by the actual task executions of the robots, stored in the data sharing system. We conducted model update experiments under this model circulation process in two scenarios. The results confirmed that the model circulation in the ecosystem effectively works for model updating through the autonomous robot behaviors using the action models and the model training by the model builders. Although the current system is the early stage, we believe that a larger scale of open collaboration among the robots and the model builders in the ecosystem will contribute to build and share various types of advanced action models applicable to dynamic situations in the real world.

As the future works, we will build a variety of action models across multiple generations in the robot AI ecosystem. The ecosystem will be extended to add the functions as the economic platform including payments in transactions of model and demonstration data. The AI-based robotic society in the future will be simulated in this system.

References

[1] Y. Sakai, K. Morioka, "Economic Platform for Action Model Sharing Based on Blockchain in Mobile Robot Networks," Proc. of IEEE/SICE SII, pp.1136-1141, 2025.

[2] Yuqing Xie, Chao Yu, Hongzhi Zang, Feng Gao, Wenhao Tang, Jingyi Huang, Jiayu Chen, Botian Xu, Yi Wu, Yu Wang, "Multi-UAV Formation Control with Static and Dynamic Obstacle Avoidance via Reinforcement Learning", arXiv, 2410.18495, 2025.

[3] Levine, Sergey and Vanhoucke, Vincent and Goldberg, Ken, "Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences", PMLR, 2017.

[4] Ryuto Tsuruta, Kazuyuki Morioka, "Autonomous Navigation of a Mobile Robot with a Monocular Camera Using Deep Reinforcement Learning and Semantic Image Segmentation", Proc. of IEEE/SICE SII, pp.1107-1112, 2024.

[5] Naoki Yokoyama, Alex Clegg, Joanne Truong, Eric Undersander, Tsung-Yen Yang, Sergio Arnaud, Sehoon Ha, Dhruv Batra, Akshara Rai. "ASC: Adaptive Skill Coordination for Robotic Mobile Manipulation", IEEE Robotics and Automation Letters, Vol.9, No.1, pp.779-786, 2024

[6] Ajay Sridhar, Dhruv Shah, Catherine Glossop, Sergey Levine, "NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration", arXiv, 2310.07896, 2023.

[7] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, Sergey Levine, "ViNT: A Foundation Model for Visual Navigation", arXiv, 2306.14846, 2023.

[8] Müller, Matthias and Brahmbhatt, Samarth and Deka, Ankur and Leboutet, Quentin and Hafner, David and Koltun, Vladlen, "OpenBot-Fleet: A System for Collective Learning with Real Robots", IEEE International Conference on Robotics and Automation (ICRA), pp.4758-4765, 2024

[9] Dagdanov, Resul and Durmus, Halil and Ure, Nazim Kemal, "Self-Improving Safety Performance of Reinforcement Learning Based Driving with Black-Box Verification Algorithms", IEEE International Conference on Robotics and Automation (ICRA), pp.5631-5637, 2023

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, "Proximal policy optimization algorithms", arXiv preprint, 1707.06347, 2017.

[11] Jonathan Ho, Stefano Ermon, "Generative Adversarial Imitation Learning", arXiv, 1606.03476, 2016.