IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 May 2022, accepted 5 June 2022, date of publication 23 June 2022, date of current version 1 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185651

Imitation Learning for Nonprehensile Manipulation Through Self-Supervised Learning Considering Motion Speed

YUKI SAIGUSA[1], SHO SAKAINO[2], (Member, IEEE), AND TOSHIAKI TSUJI[3], (Senior Member, IEEE)

[1]Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan
[2]Department of Intelligent Interactive Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8577, Japan
[3]Graduate School of Science and Engineering, Saitama University, Saitama 338-8570, Japan

Corresponding author: Yuki Saigusa (saigusa.yuki.xp@alumni.tsukuba.ac.jp)

This work was supported in part by the Adaptable and Seamless Technology Transfer Program through Target-driven Research and Development (A-STEP) from the Japan Science and Technology Agency (JST) under Grant JPMJTR20RG, and in part by the Japan Society for the Promotion of Science by a Grant-in-Aid for Scientific Research (B) under Grant 21H01347.

ABSTRACT Robots are expected to replace menial tasks such as housework. Some of these tasks include nonprehensile manipulation performed without grasping objects. Nonprehensile manipulation is very difficult because it requires considering the dynamics of environments and objects. Therefore imitating complex behaviors requires a large number of human demonstrations. In this study, a self-supervised learning that considers dynamics to achieve variable speed for nonprehensile manipulation is proposed. The proposed method collects and fine-tunes only successful action data obtained during autonomous operations. By fine-tuning the successful data, the robot learns the dynamics among itself, its environment, and objects. We experimented with the task of scooping and transporting pancakes using the neural network model trained on 24 human-collected training data. The proposed method significantly improved the success rate from 40.2% to 85.7%, and succeeded the task more than 75% for other objects.

INDEX TERMS Bilateral control, imitation learning, machine learning, motion planning, nonprehensile manipulation, self-supervised learning.

I. INTRODUCTION

Robotic manipulation has been widely studied to automate daily human tasks [1]–[3]. Object manipulation involves grasping an object to stabilize it and unrestrained manipulation without grasping the object. The latter unrestrained manipulation is called nonprehensile manipulation, which allows manipulation beyond the range of motion and manipulation of objects that cannot be grasped [4], [5]. However, nonprehensile manipulation requires a complex dynamic model of the object and the environment, in addition to the relationship between the robot hand and object or tool [6]. Model-based methods that formulate complex dynamics with mathematical models have been studied [7], [8]. In [8], Planning for sliding and moving an object considering friction and for carrying an object with it considering gravity and inertia are very difficult. Fotheremore, these methods assume the use of known object shapes, postures, material, and desired trajectories. Therefore these methods are computationally expensive and difficult to adapt to new objects and environments [9].

Recently, end-to-end learning that directly learns object manipulation skills from sensor data has been focused. In particular, imitation learning is known to be one of the most data-efficient methods. Imitation learning is supervised learning from motion data that are demonstrated by humans [10], [11]; therefore, it learns operation skills that require considering the dynamics of the environment and objects. There are studies of nonprehensile manipulation using imitation learning such as pushing objects [12], [13], ball-in-box [14], and lifting shoes [15]; these approaches learn behaviors from data without explicitly giving physical properties such as friction or shape of the manipulated object and without calculating dynamics. However, the above studies were conducted at a single speed, which is slower than human daily movements. This is because these imitation learning methods predicted

The associate editor coordinating the review of this manuscript and approving it for publication was Larbi Boubchir.

the next response value of the robot and gave it as a command value. In general, there is no ideal control system, and there is a delay between the command and response values. Therefore, conventional imitation learning [12]–[15] requires the operation to be very slow to ensure negligible control delay caused during autonomous motion. In particular, fast and dynamic nonprehensile manipulation is significantly affected by this control delay, and task execution is impossible as well.

Bilateral control-based imitation learning has been proposed as a method that can compensate for this control delay [16], [17]. We indicated that this method can generate variable speed motions that consider the dynamics between the robot and environment [18], [19]. Although this method is expected to enable nonprehensile manipulations at high speed and multiple speeds, it is difficult to learn complex dynamics between objects and the environment from sensor data and requires a significant amount of training data. Because imitation learning requires human demonstration to collect the teacher data, collecting a large amount of teacher data is costly and becomes an obstruction for complex task execution.

In contrast, self-supervised learning has been studied to enhance supervised learning [20], [21]. Self-supervised learning allows for self-learning without generate and annotate the training data by humans. Self-supervised learning has also been studied in the field of robotics, such as the feature extraction of image recognition [22], grasping [23]–[25], and liquid pouring [26]; however, these studies have focused on geometric changes such as the grasping position and shape of the object, and there has been no study focused on speed, which is a temporal change. When the nonprehensile motion speed is changed, the dynamics of the object, as well as the robot itself, can change significantly. Although it is very difficult to represent these complex relationships as functions of various speeds, it is expected to be possible to model them with Neural Networks (NNs) based on a large amount of data collected by self-supervised learning.

In this study, we propose self-supervised learning considering motion speed. The proposed method improves the success rate by performing the task with a trained NN and using only the success data for self-supervised learning. We performed the task of scooping and transporting a pancake illustrated in Fig. 1. The experimental results indicated that the proposed method improved the success rate and the reproducibility of the motion speed as well. Finally, the task success rate was improved from 40.2% to 85.7%, and it achieved higher accuracy of time reproduction than humans. The proposed method does not simply linearly expand the trajectory in response to the speed command but learns the dynamics and generates an appropriate motion depending on the motion speed. Furthermore, the proposed method does not explicitly provide the friction and inertia of the pancake, enabling the NN to learn appropriate actions that take them in consideration from the human demonstration. The fact that NN can generate trajectories based on an understanding of the dynamic characteristics of the environment with dynamics is a great academic novelty, and we can expect integration with conventional studies in the control field in the future because changing commands according to speed is similar to designing controllers according to frequency responses. And proposed method is expected to have a significant social impact in terms of the ease of system implementation and the ability to perform difficult tasks. This study is the first approach in imitation learning that makes dynamic behaviors such as high-speed nonprehensile manipulation possible by performing self-supervised learning using bilateral control-based imitation learning and discusses the limitations and future of this method. The advantages of the proposed method are as follows:

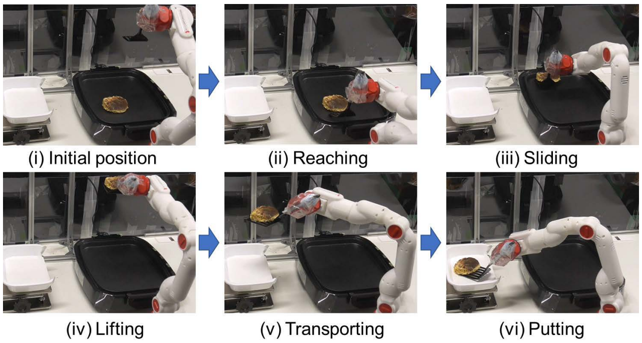


FIGURE 1. Snapshot of scooping and transporting a pancake task.

- Automatically collects speed-labeled training data and enables self-supervised learning
- Learns multiple speeds and positions nonprehensile manipulation from small amounts of human demonstrations
- Is capable of reproducing speeds that exceed human-given training data

The remainder of this paper is organized as follows. Section II presents the related work of this study. Section III explains the robot control system and bilateral control. In Section IV, we introduce bilateral control-based imitation learning. Section V introduces our proposed method of self-supervised learning considering motion speed. Section VI provides the experimental description, results, and discussion. Finally, in Section VII, we conclude this study and discuss future research topics.

II. RELATED WORK

A. NONPREHENSILE MANIPULATION

Nonprehensile manipulation is performed by not directly grasping the object. Although model-based methods have been studied, there is no general methodology, and nonprehensile manipulation is still a difficult task [8], [27]. In particular, it is difficult for model-based methods to adapt to change in the state of contact between the object and environment. Iterative learning control, which performs repetitive motions and corrects modeling errors, has also been reported [28]. However, [28] assumes that the desired trajectory is

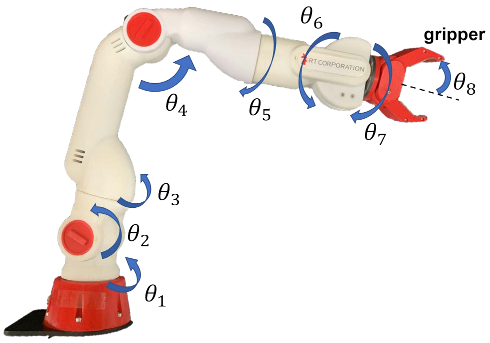


FIGURE 2. Definition of robot's joints (CRANE-X7).

predetermined, which makes it difficult to adapt to changes in the environment.

Recently, model-free methods using machine learning, such as reinforcement learning and imitation learning, have been studied. Reinforcement learning automatically learns behavior based on a reward function through trial and error [29], [30]. Yuan *et al.* performed the task of pushing and replacing objects [31], and Finn *et al*. performed the task of scooping up objects with a spatula [32]. However, reinforcement learning requires a huge number of trials to learn a behavior, and it is difficult to set the reward function [33]. In particular, learning nonprehensile manipulation at multiple speeds is inefficient because different reward functions need to be set up and relearned for each speed. In contrast, imitation learning can be applied with less teacher data because it learns from human motion data. Moreover, it can be easily used by people who are not experts in robotics or machine learning because it does not require reward functions or mathematical models.

B. IMITATION LEARNING

Imitation learning is one of the most efficient methods of end-to-end learning. In the past, several methods using probabilistic models such as hidden Markov models [34] and mixed Gaussian models [35] using NN [36]–[38] and dynamic movement primitives [39], [40] have been studied and proved to be effective. However, the movement of these imitation learnings is slower than humans, owing to control delays that occur only during autonomous operations.

Bilateral control-based imitation learning predicts the next command value from the robot's response value and gives that command value to the robot [16], [17]. This framework of predicting the command value from the response value allows for fast object manipulation considering the control delay. Because this control delay has a significant effect on nonprehensile manipulation, this study employs bilateral control-based imitation learning.

C. SELF-SUPERVISED LEARNING

Self-supervised learning, mainly spread in the field of image recognition, was used as a pre-training for supervised learning in [41], and indirectly improved the success rate of the main task by setting sub-tasks in [42].

Recently, self-supervised learning has also been studied in the field of robotics. Most of the research has been focused on image information, such as object pose estimation [43], [44]. Huang *et al.* applied self-supervised learning to robot motion data for a task of pouring a specified amount of liquid [26]. In [26], self-learning based on the actual amount of liquid poured was employed to enable the robot pour a specified amount of liquid, even if it is a new liquid; however, [26] deals with the control of one degree of freedom (DOF) and does not consider the operation time.

Regarding dynamic object manipulation such as nonprehensile manipulation at any speed, it is necessary to consider the interaction with the environment and the dynamics of the manipulated object according to the motion speed. There has been no study on self-supervised learning to learn dynamic behaviors using a time metric. In this study, we propose a self-supervised learning method using task completion time to achieve nonprehensile manipulation at any speed. The proposed method allows the NN to learn the dynamics of the entire task through the metric of speed. In the field of control engineering, it is possible to understand the dynamics of a linear system by drawing a Bode diagram from input-output data at multiple frequencies. Given that the NN input and output data of nonprehensile operations performed at multiple speeds are similar while using Bode diagrams, it will be possible to learn the dynamics of the entire task. Our proposal is different from conventional methods in that it simply changes the parameter of the amount of liquid to motion speed, because it is capable of learning the dynamics of the environment and manipulated objects, and it is possible to generate nonlinear motion in accordance with speed.

III. CONTROL SYSTEM

A. MANIPULATER

We utilized two CRANE-X7 manipulators manufactured by RT Corporation, as illustrated in Fig. 2. This robot has a mechanism of 7-DOF joints and a 1-DOF gripper, and the angle of each joint can be measured. The angles corresponding to each joint $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7$, and gripper θ_8 are defined as per Fig. 2.

B. CONTROLLER

The manipulator control system comprised position and force controllers. The position controller comprised a proportional and differential controller, whereas the force controller comprised a proportional controller. The control system is depicted in Fig. 3. In the figure, $\theta, \dot{\theta}$, and τ represent the angle, angular velocity, and torque of each joint, respectively, and the superscripts *cmd*, *res*, *ref*, and *dis* represent the command, response, reference, and disturbance values, respectively. The caret $\hat{\bigcirc}$ indicates the estimated values. The joint angle of each joint was obtained by the optical encoder and angular velocity was calculated by its pseudo-differential.

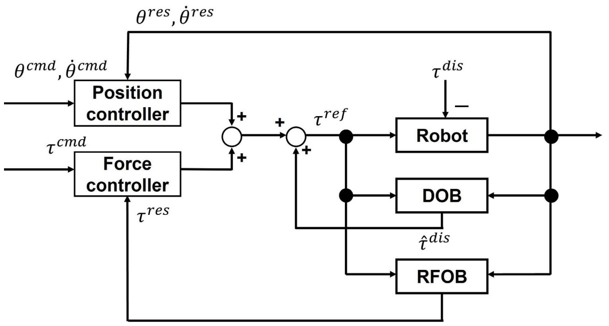


FIGURE 3. Block diagram of controller.

The disturbance torque τ^{dis} was calculated using a disturbance observer (DOB) [45] and the torque response value τ^{res} was calculated using a reaction force observer (RFOB) [46]. Specifically, a force sensor was not utilized in this study. The details of RFOB are presented in Section III-D.

C. FOUR-CHANNEL BILATERAL CONTROL

Bilateral control is a teleoperation system that utilizes two robots: a leader and a follower. Bilateral control synchronizes the positions of the two robots and presents the reaction force caused by the contact of the follower with the environment to the leader [47]. Using this technique, the human operating the leader can execute tasks as if they were directly controlling the follower. In particular, four-channel bilateral control, which has position and force controllers in both the leader and follower, is the best method for imitation learning using force information [17]. Therefore, four-channel bilateral control was implemented in this study. The block diagram of four-channel bilateral control is showed in Fig. 4. The angle and torque targets for four-channel bilateral control are defined as follows:

$$\boldsymbol{\theta}_l^{res} - \boldsymbol{\theta}_f^{res} = \mathbf{0}, \tag{1}$$

$$\boldsymbol{\tau}_l^{res} + \boldsymbol{\tau}_f^{res} = \mathbf{0}. \tag{2}$$

where $\boldsymbol{\theta}$ is the angle vector, $\boldsymbol{\tau}$ is the torque vector, and the subscripts l and f represent the leader and follower, respectively. Because this study utilized a robot with 8-DOF, including the gripper, the angle response value vector $\boldsymbol{\theta}^{res}$ is $\boldsymbol{\theta}^{res} = [\theta_1^{res}, \theta_2^{res}, \cdots, \theta_8^{res}]^T$. The torque response value vector $\boldsymbol{\tau}^{res}$ is $\boldsymbol{\tau}^{res} = [\tau_1^{res}, \tau_2^{res}, \cdots, \tau_8^{res}]^T$. The torque reference value vectors of the controller that satisfies equations (1) and (2) are represented by the following equations:

$$\begin{aligned}\boldsymbol{\tau}_l^{ref} &= -\frac{\boldsymbol{J}}{2}(\boldsymbol{K_p} + \boldsymbol{K_d}s)(\boldsymbol{\theta}_l^{res} - \boldsymbol{\theta}_f^{res})\\ &\quad - \frac{1}{2}\boldsymbol{K_f}(\boldsymbol{\tau}_f^{res} + \boldsymbol{\tau}_l^{res}),\end{aligned} \tag{3}$$

$$\begin{aligned}\boldsymbol{\tau}_f^{ref} &= \frac{\boldsymbol{J}}{2}(\boldsymbol{K_p} + \boldsymbol{K_d}s)(\boldsymbol{\theta}_l^{res} - \boldsymbol{\theta}_f^{res})\\ &\quad - \frac{1}{2}\boldsymbol{K_f}(\boldsymbol{\tau}_f^{res} + \boldsymbol{\tau}_l^{res}).\end{aligned} \tag{4}$$

where $\boldsymbol{K_p}$ and $\boldsymbol{K_d}$ represent the diagonal gain matrix of proportional and differential control of the position, $\boldsymbol{K_f}$ represents the diagonal proportional control gain matrix of force, and $\boldsymbol{J}$ is the inertia matrix and s is the Laplace operator. These matrix subscripts represent each joint, and the interference of the axes was neglected.

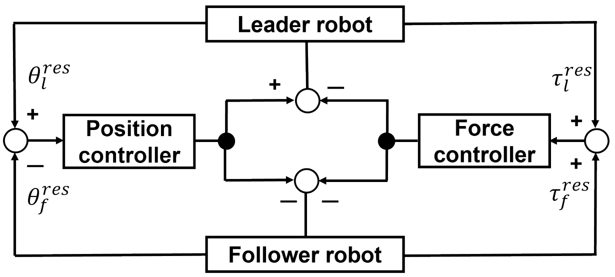


FIGURE 4. Block diagram of four-channel bilateral control.

TABLE 1. Gain values for the robot controller.

	Parameter	Value
K_{p1}	Joint 1's Position feedback gain	256
K_{p2}	Joint 2's Position feedback gain	196
K_{p3}	Joint 3's Position feedback gain	961
K_{p4}	Joint 4's Position feedback gain	225
K_{p5}	Joint 5's Position feedback gain	289
K_{p6}	Joint 6's Position feedback gain	324
K_{p7}	Joint 7's Position feedback gain	144
K_{p8}	Gripper's Position feedback gain	324
K_{d1}	Joint 1's Velocity feedback gain	32.0
K_{d2}	Joint 2's Velocity feedback gain	28.0
K_{d3}	Joint 3's Velocity feedback gain	62.0
K_{d4}	Joint 4's Velocity feedback gain	30.0
K_{d5}	Joint 5's Velocity feedback gain	34.0
K_{d6}	Joint 6's Velocity feedback gain	36.0
K_{d7}	Joint 7's Velocity feedback gain	24.0
K_{d8}	Gripper's Velocity feedback gain	36.0
K_{f1}	Joint 1's Force feedback gain	0.70
K_{f2}	Joint 2's Force feedback gain	0.70
K_{f3}	Joint 3's Force feedback gain	1.00
K_{f4}	Joint 4's Force feedback gain	0.75
K_{f5}	Joint 5's Force feedback gain	0.80
K_{f6}	Joint 6's Force feedback gain	1.00
K_{f7}	Joint 7's Force feedback gain	0.80
K_{f8}	Gripper's Force feedback gain	1.00

$$\boldsymbol{K_p} = diag[K_{p1}, K_{p2}, \cdots, K_{p8}], \tag{5}$$

$$\boldsymbol{K_d} = diag[K_{d1}, K_{d2}, \cdots, K_{d8}], \tag{6}$$

$$\boldsymbol{K_f} = diag[K_{f1}, K_{f2}, \cdots, K_{f8}], \tag{7}$$

$$\boldsymbol{J} = diag[J_1, J_2, \cdots, J_8]. \tag{8}$$

The gain values adopted are listed in Table 1. The control gains were determined by trial and error to be the maximum value within the range in which the robot does not vibrate. We used the same value for both robots. In this study, θ_1 and θ_2 were controlled with a cutoff frequency of 15.0 rad/s, and the other joints with a cutoff frequency of 20.0 rad/s.

D. ROBOT DYNAMICS AND SENSORLESS REACTION FORCE MEASUREMENT

In this section, we describe the dynamics of the robot and torque sensorless measurement of reaction force using RFOB. In this study, the dynamics of the robot is represented

TABLE 2. Identified system parameters for the robot controller.

	Parameter	Value
J_1	Joint 1's inertia [N·m]	0.012
J_2	Joint 2's inertia [N·m]	0.113
J_3	Joint 3's inertia [N·m]	0.012
J_4	Joint 4's inertia [N·m]	0.040
J_5	Joint 5's inertia [N·m]	0.006
J_6	Joint 6's inertia [N·m]	0.007
J_7	Joint 7's inertia [N·m]	0.006
J_8	Gripper's inertia [N·m]	0.007
D_1	Joint 1's friction compensation cofficient [N·m]	0.050
D_2	Joint 3's friction compensation cofficient [N·m]	0.242
D_3	Joint 5's friction compensation cofficient [N·m]	0.040
D_4	Joint 6's friction compensation cofficient [N·m]	0.039
D_5	Joint 7's friction compensation cofficient [N·m]	0.050
D_6	Gripper's friction compensation cofficient [N·m]	0.021
M_{l1}	Leader's joint 2 gravity compensation coefficient 1 [N·m]	2.094
M_{l2}	Leader's joint 2 gravity compensation coefficient 2 [N·m]	1.151
M_{l3}	Leader's joint 4 gravity compensation coefficient 1 [N·m]	1.183
M_{f1}	Follower's joint 2 gravity compensation coefficient 1 [N·m]	2.294
M_{f2}	Follower's joint 2 gravity compensation coefficient 2 [N·m]	1.451
M_{f3}	Follower's joint 4 gravity compensation coefficient 1 [N·m]	1.483

as follows:

$$J_1\ddot{\theta}_1^{res} = \tau_1^{ref} - \tau_1^{res} - D_1\dot{\theta}_1^{res}, \tag{9}$$

$$J_2\ddot{\theta}_2^{res} = \tau_2^{ref} - \tau_2^{res} - M_1\sin(\theta_2^{res}) + M_2\sin(\theta_2^{res} + \theta_4^{res}), \tag{10}$$

$$J_3\ddot{\theta}_3^{res} = \tau_3^{ref} - \tau_3^{res} - D_2\dot{\theta}_3^{res}, \tag{11}$$

$$J_4\ddot{\theta}_4^{res} = \tau_4^{ref} - \tau_4^{res} + M_3\sin(\theta_2^{res} + \theta_4^{res}), \tag{12}$$

$$J_5\ddot{\theta}_5^{res} = \tau_5^{ref} - \tau_5^{res} - D_3\dot{\theta}_5^{res}, \tag{13}$$

$$J_6\ddot{\theta}_6^{res} = \tau_6^{ref} - \tau_6^{res} - D_4\dot{\theta}_6^{res}, \tag{14}$$

$$J_7\ddot{\theta}_7^{res} = \tau_7^{ref} - \tau_7^{res} - D_5\dot{\theta}_7^{res}, \tag{15}$$

$$J_8\ddot{\theta}_8^{res} = \tau_8^{ref} - \tau_8^{res} - D_6\dot{\theta}_8^{res}. \tag{16}$$

where D and M are the friction and gravity compensation coefficients, respectively, and the respective subscripts are adopted to identify each coefficient. The off-diagonal terms in the inertia matrix were considered as negligible. The subscripts of the other parameters correspond to the numbers of the joints. Based on (9)–(16), the physical parameters were identified. Using maximum length null sequence signal input, measured angular velocity and angular acceleration during free motion of each joint. Then, assuming that $\boldsymbol{\tau}^{res} = \mathbf{0}$, each parameter was determined by the least squares method. The output of DOB, calculated from the torque reference and response values of acceleration, is represented as follows:

$$\hat{\boldsymbol{\tau}}^{dis} = \boldsymbol{\tau}^{ref} - \boldsymbol{\tau}^{res} - \boldsymbol{J}\ddot{\boldsymbol{\theta}}^{res}, \tag{17}$$

Using the identified parameters and (9)–(16), the reaction force was calculated from RFOB without force sensor as expressed in the following equation.

$$\tau_1^{res} = \tau_1^{dis} - D_1\dot{\theta}_1^{res}, \tag{18}$$

$$\tau_2^{res} = \tau_2^{dis} - M_1\sin(\theta_2^{res}) + M_2\sin(\theta_2^{res} + \theta_4^{res}), \tag{19}$$

$$\tau_3^{res} = \tau_3^{dis} - D_2\dot{\theta}_3^{res}, \tag{20}$$

$$\tau_4^{res} = \tau_4^{dis} + M_3\sin(\theta_2^{res} + \theta_4^{res}), \tag{21}$$

$$\tau_5^{res} = \tau_5^{dis} - D_3\dot{\theta}_5^{res}, \tag{22}$$

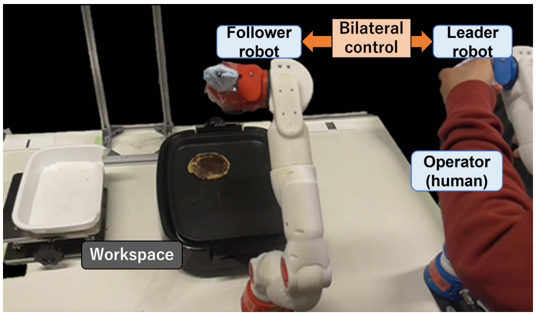


FIGURE 5. Collection of training data using bilateral control.

$$\tau_6^{res} = \tau_6^{dis} - D_4\dot{\theta}_6^{res}, \tag{23}$$

$$\tau_7^{res} = \tau_7^{dis} - D_5\dot{\theta}_7^{res}, \tag{24}$$

$$\tau_8^{res} = \tau_8^{dis} - D_6\dot{\theta}_8^{res}. \tag{25}$$

The physical parameters of the robot are listed in Table 2. In this study, the turner was fixed on the follower, thus the gravity compensation value of the follower was increased.

IV. BILATERAL CONTROL-BASED IMITATION LEARNING

Bilateral control-based imitation learning can imitate the subtle force of humans and manipulate objects faster than other imitation learning methods. In this section, the flow of bilateral control-based imitation learning is divided into three phases.

A. COLLECTING TRAINING DATA

In bilateral control-based imitation learning, two robots were utilized only when collecting training data. We experimented with nonprehensile manipulation by scooping up a pancake on a hot plate, and thereafter transporting and placing it on an adjacent tray at multiple speeds. Therefore, operator demonstrated nonprehensile manipulation at multiple speeds using bilateral control. The operator executed the task by controlling the leader and remotely controlling the follower that was in the workspace, as illustrated in Fig. 5. Because the robot in Fig. 2 had a redundant DOF, θ_3 was fixed with position control, and the robot was controlled at a frequency of 500 Hz, which was the fastest control cycle for this robot.

B. TRAINING THE NN MODEL

In bilateral control-based imitation learning, using NN as learning model. The basic structure of the NN for bilateral control-based imitation learning is displayed in Fig. 6.

In bilateral control-based imitation learning, the model is constructed with the response value of the follower as the input and that of the leader as the output. This configuration makes it possible to reproduce the same system in autonomous operation as during training data collection, but without the human and leader robot. The difference between bilateral control during collecting training data and executing a task is illustrated in Fig. 7. Recurrent Neural Network (RNN) [12], [16]–[19], [26], [38] and Convolutional

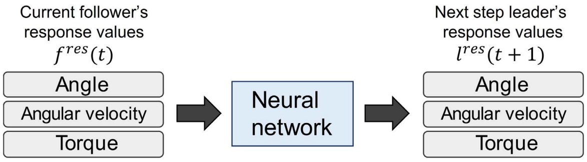


FIGURE 6. Basic structure of learning model in bilateral control-based imitation learning.

Neural Network (CNN) [12], [13], [15], [36], [37] are mainly used in the research of robot motion generation using NNs. RNN was suitable for learning time series information, while CNN was suitable for learning image information. In this study, we used RNN because we did not use image information for training, and among RNN, we used Long Short-Term Memory (LSTM), which can handle long time-series data.

C. EXECUTING A TASK

During autonomous operation, the trained NN replaced the human and the leader robot. In this study, the inference of NN was done every 20 ms, and the control cycle of the robot was 2 ms, which was the same as that when collecting the training data.

V. SELF-SUPERVISED LEARNING CONSIDERING SPEED

Our self-supervised learning fine-tunes the NN using data of autonomous operations conducted in Section IV-C. Therefore, this method is regarded as the fourth stage following those described in Section IV: collecting training data, training the NN model, and executing a task. The process from bilateral control-based imitation learning to self-supervised learning of the proposal is illustrated in Fig. 8. When a task is executed using an NN that has not been overtrained, the behavior in each trial is different as though it is similar. In our method, trying the task multiple times and using only the successful motions within the variance for learning makes it possible to learn only the good behaviors from the minute differences. Repeating this process at multiple speeds will enable progressive learning of the dynamics of the entire task and improve the success rate.

To reuse the motion data during autonomous operation, it is necessary to determine the success or failure of the task, as well as annotating the generated behavior. In the following, we explain the automated annotation method, the method of determining success or failure, and the method of creating a dataset for relearning.

A. DETERMINE THE TASK COMPLETION TIME

In this study, the task was to scoop up a pancake on a hot plate and place it on an adjacent tray. Therefore, the time when the torque response value of θ_2, triggered by the movement to place the pancake on the tray was observed above a threshold, was defined as the task completion time. The angle response value of θ_1 and torque response value of θ_2 during task execution are illustrated in Fig. 9. The red line in Fig. 9 (b) is 1.2 N·m, which was set as the threshold in this study, and t_f is the task completion time. The sliding and putting phases are distinguished using a threshold based on the angular response value of θ_1 in Fig. 9 (a). Based on the accurately labeled speed, the NN corrects the error of the task completion time.

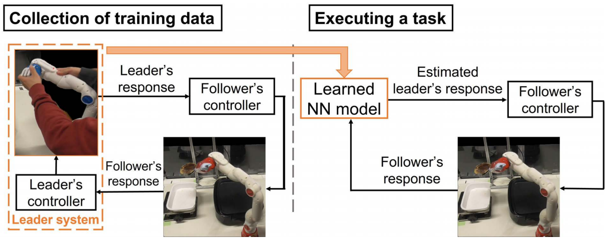


FIGURE 7. Difference in bilateral control when collecting training data and executing a task.

B. JUDGING THE SUCCESS OR FAILURE OF A TASK

The image on the tray was acquired at the end of the task, and after binarization using HSV color space, the area of the yellow object was calculated to determine its success or failure. An example of the binarization is illustrated in Fig. 10 (b). As the threshold for binarization using HSV color space, pixels in the range of 15 < h < 35.5, 100 < s, and v < 180 were set as white. Five hundred pixels or more of the binarized image were considered successful.

C. GENERATE POSITION COMMAND

To demonstrate that the proposed method can be employed with self-supervised learning focused on spatial information, we considered the initial position of the pancake. The image of the workspace at the start was obtained, image was binarized by color as in Section V-B, coordinates of the center on the pixel were calculated, and these were adopted as the position coordinates (x,y) of the pancake. The calculated position coordinates were adopted as position command values. Fig. 10 illustrates the coordinates were calculated.

D. CREATING A DATA SET FOR RETRAINING

In self-supervised learning, there is no difference between a method that repeatedly collects and trains a small amount of data and a method that collects and trains a large amount of data at once [26]. Therefore, we assume that only a significant amount of data was collected and trained at a time. In short, we performed several actions with the same model and added the obtained action data to the original training dataset to create a new dataset.

VI. EXPERIMENT & EVALUATION

We examined the task of scooping and transporting a pancake, and this task has different difficulties in each phase. In the reaching and sliding phases, the robot needed to learn sufficient spatial information, because the robot needed to reach an appropriate position and then slide it to the edge of the hot plate. In the lifting phase, the turner needed to be pressed against the edge of the hot plate with a proper force and

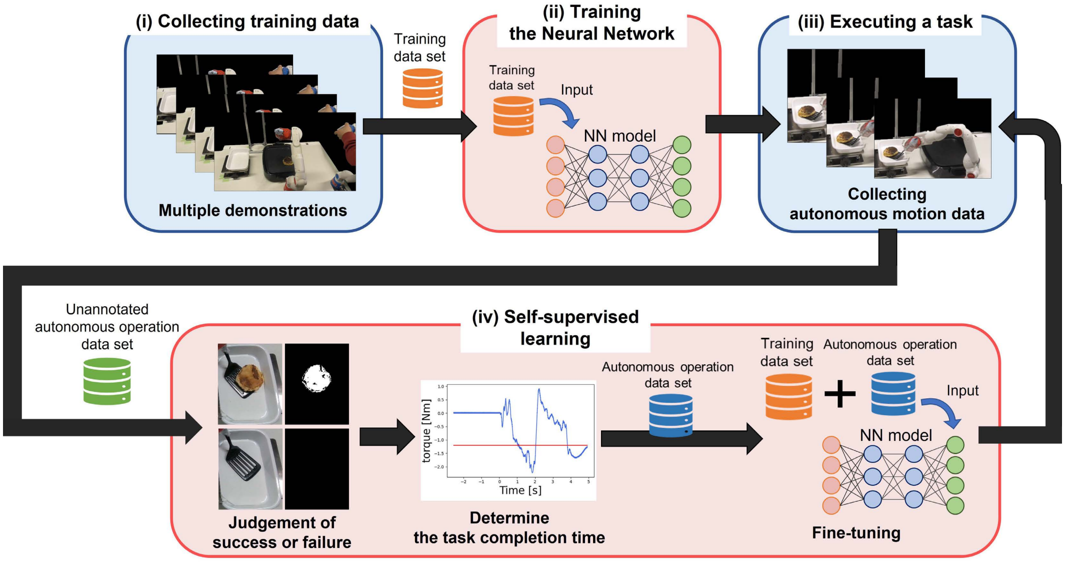
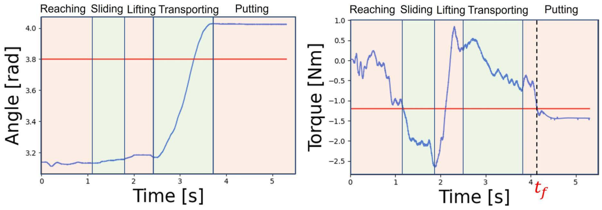

(i) Collecting training data
Multiple demonstrations
Training data set
(ii) Training the Neural Network
Training data set
Input
NN model
(iii) Executing a task
Collecting autonomous motion data
Unannotated autonomous operation data set
(iv) Self-supervised learning
Judgement of success or failure
torque [Nm]
Time [s]
Determine the task completion time
Autonomous operation data set
Training data set
Autonomous operation data set
Input
NN model
Fine-tuning

FIGURE 8. Flowchart of the proposed method. (i)–(iii) are bilateral control-based imitation learning, and (iv) is the proposed method. (iv) includes success/failure determination, teacher label generation, and fine-tuning.



(a) Angle response value of θ_1

(b) Torque response value of θ_2

FIGURE 9. Definition of task completion time.

lifted at the right angle and angular velocity. If the pancake is lifted more than necessary, the pancake will be thrown away; however, if it is not lifted sufficiently, the pancake will slip off the turner. In the transporting phase, it was necessary to move the pancake in an appropriate trajectory that considers inertia and friction. In particular, at high speeds, if the turner is not properly tilted, the pancake will fall in the opposite direction.

We inspected whether repeating the loop of self-supervised learning and executing a task twice, as illustrated in Fig. 8, would improve the success rate for failed positions and speeds, as well as the reproducibility of task completion times. Moreover, as an additional experiment, we conducted tests on untrained objects to investigate whether generalization performance could be obtained.

A. DESIGN & SETUP

The experimental environment is illustrated in Fig. 11 (a). The camera was an Intel RealSense D415. The turner was fixed to the robot hand. In the human-collected training data condition, the pancakes were located in four locations: lower left, upper left, lower right, and upper right, as illustrated in Fig. 11 (b). In addition, the task completion times were 4, 8, and 12 s, and the mass of the pancake was 30 g and 90 g. The pancakes were uniform in size, illustrated in Fig. 11 (c). In all these combinations, we collected the teacher data twice each. Thus, the number of teacher data was 48 (4 [position] × 3 [completion time] × 2 [mass] × 2 [trial]). Among these, 24 data were used as training data, and the remaining 24 data were used as validation data. The collected teacher data was downsampled to 50 Hz data with [48], and the number of training data was made 10 times larger.

Autoregressive learning is less affected by covariate shifts and can be more efficient; therefore, the S2SM model of [49] was implemented. The NN utilized in this study is illustrated in Fig. 12. It comprises a LSTM 8 layers and follows a fully connected layer. The number of layers of NNs was set to a same number with the reference [19], which achieved variable speed motion generation with contact with the environment. However, the number of units was set to 200 to enable learning of more complex expressions and to avoid a significant increase in computation time, since this study learned a 7-DOF joint, whereas the 3-DOF joint in [19] was learned. Max-min normalization was applied to all 24 dimensions of the input information, and mean squared error was adopted as the loss function. In addition, we utilized Adam [50] as the optimizer, and mini-batch training was applied to a batch of 100 data.

B. PRELIMINARY EXPERIMENT

1) DESCRIPTION

As a preliminary experiment, we performed the task with the original model learned using 24 training data. The test was performed thrice for each combination of all 9 positions illustrated in Fig. 11 (b) and 11 speeds (3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 s). Therefore, the total number of tests was 594 (9 [position] × 11 [completion time] × 2 [mass] × 3 [trial]).

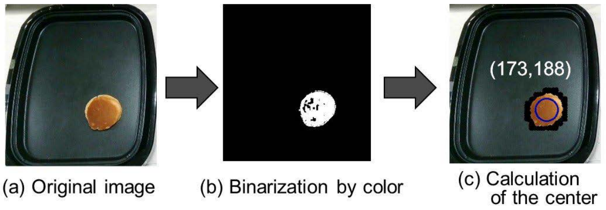


FIGURE 10. Determination of the center coordinates of the pancake.

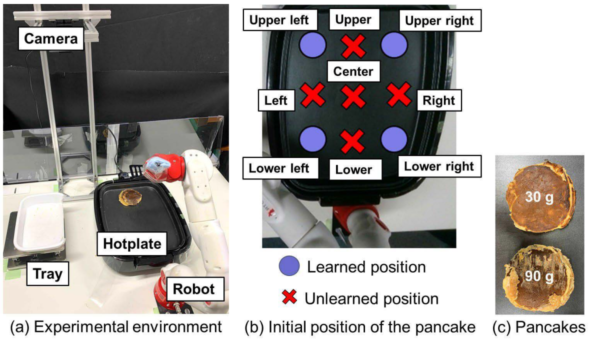


FIGURE 11. Experiment setup.

The number of training times of the NN was determined regarding the validation loss, and 8000 times was adopted. This model is referred to as the original model. The loss graph for the original model is illustrated in Fig. 13 (a)

2) RESULT

The task success rates for the original model are presented in Table 3. Because no significant difference was observed in the results depending on the mass of the pancake, a summary table is presented here. The individual results are provided in the appendix. The success rate of the original model was 40.2% and the task success rates at untrained positions were remarkably low. We determined that successes and failures depend on the task completion time command even if the position command was the same. Most of the failures of the original model were that it operated in a different position from what was commanded. Furthermore, the pancake slipped off when it started to be transported at high speed, and was thrown away during the lifting phase.

Fig. 14 illustrates the reproducibility of completion time. In this figure, the black line is the ideal line, and only successful trials are plotted. Therefore, the position command of Upper with a task completion time of 4 s is not adopted to calculate the mean and variance. From Fig. 14 (a), we can observe that the unlearned speeds of 5 to 6 s and 10 to 11 s are far from the commanded values, and the variance is also very large. The task completion time entirely tended to be slower than the command time.

C. SELF-SUPERVISED LEARNING(1ST TIME)

1) DESCRIPTION

To evaluate the effectiveness of self-supervised learning, we conducted an experiment using the autonomous action

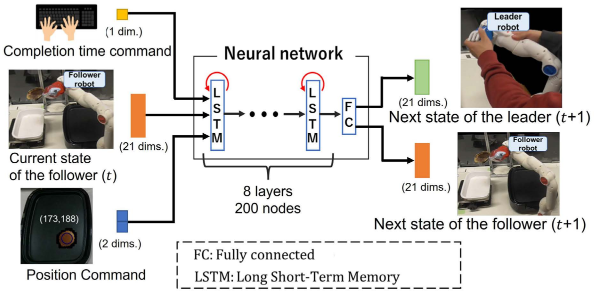


FIGURE 12. Structure of the NN implemented in this study.

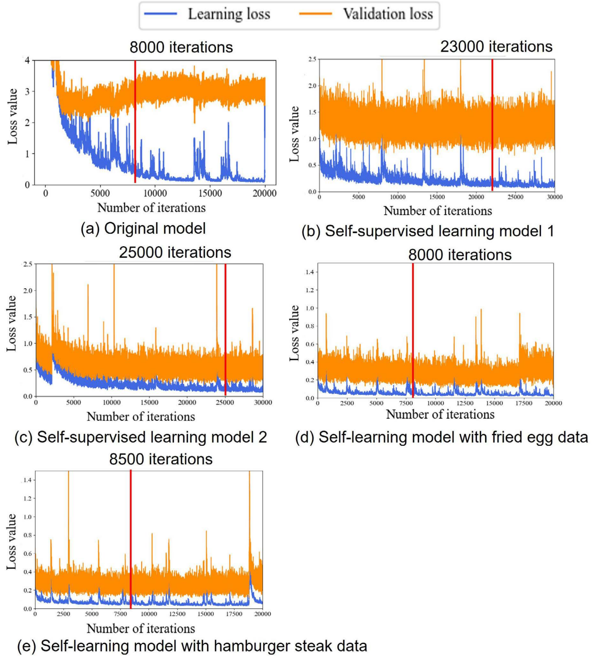


FIGURE 13. Training loss and validation loss for each neural network.

data from Section VI-B. For all patterns that were successful at least once during the trials in Section VI-B, we collected additional data so that the success data would be thrice each. Therefore, the total number of autonomous motion data collected was 300. Among these, 200 data and 100 data were used as training and validation data, respectively. Therefore, the new training dataset comprised 224 data and the validation dataset comprised 124 data. The number of trials was 594, the same as in Section VI-B. The number of training times of the NN was determined considering the validation loss, and 23000 times was adopted. This model is referred to as the self-supervised learning model 1. The loss graph for the self-supervised learning model 1 is illustrated in Fig. 13 (b)

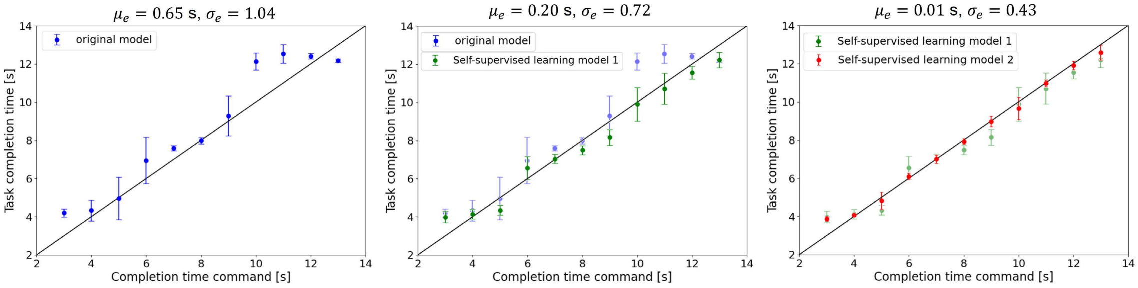
2) RESULT

The task success rates for the self-supervised learning model 1 are listed in Table 4. We determined that the success rate of the self-supervised learning model 1 improved to 81.6%, which was approximately double the success rate of

TABLE 3. Task success rate of the original model.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	0.0(0/6)	50.0(3/6)	0.0(0/6)	**100(6/6)**	0.0(0/6)	0.0(0/6)	**100(6/6)**	0.0(0/6)	0.0(0/6)	27.8(15/54)
4.00	*16.7(1/6)	***100(6/6)**	*50.0(3/6)	*66.7(4/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	0.0(0/6)	0.0(0/6)	59.3(32/54)
5.00	50.0(3/6)	50.0(3/6)	0.0(0/6)	**100(6/6)**	**100(6/6)**	66.7(4/6)	66.7(4/6)	16.7(1/6)	0.0(0/6)	50.0(27/54)
6.00	0.0(0/6)	0.0(0/6)	0.0(0/6)	**100(6/6)**	0.0(0/6)	50.0(3/6)	66.7(4/6)	16.7(1/6)	0.0(0/6)	25.9(14/54)
7.00	66.7(4/6)	0.0(0/6)	0.0(0/6)	**100(6/6)**	0.0(0/6)	0.0(0/6)	0.0(0/6)	66.7(4/6)	0.0(0/6)	25.9(14/54)
8.00	***83.3(5/6)**	*0.0(0/6)	***83.3(5/6)**	***100(6/6)**	0.0(0/6)	0.0(0/6)	66.7(4/6)	16.7(1/6)	0.0(0/6)	38.9(21/54)
9.00	50.0(3/6)	0.0(0/6)	66.7(4/6)	**100(6/6)**	0.0(0/6)	50.0(3/6)	0.0(0/6)	**83.3(5/6)**	0.0(0/6)	38.9(21/54)
10.0	16.7(1/6)	0.0(0/6)	50.0(3/6)	**100(6/6)**	**83.3(5/6)**	16.7(1/6)	66.7(4/6)	**100(6/6)**	0.0(0/6)	48.1(26/54)
11.0	0.0(0/6)	**100(6/6)**	**83.3(5/6)**	**83.3(5/6)**	**83.3(5/6)**	0.0(0/6)	**83.3(5/6)**	50.0(3/6)	33.3(2/6)	57.4(31/54)
12.0	*0.0(0/6)	***100(6/6)**	*0.0(0/6)	*66.7(4/6)	**83.3(5/6)**	0.0(0/6)	66.7(4/6)	66.7(4/6)	0.0(0/6)	42.6(23/54)
13.0	0.0(0/6)	**100(6/6)**	0.0(0/6)	**83.3(5/6)**	0.0(0/6)	0.0(0/6)	0.0(0/6)	66.7(4/6)	0.0(0/6)	27.8(15/54)
Position total	25.8(17/66)	45.5(30/66)	30.3(20/66)	**90.9(60/66)**	40.9(27/66)	25.8(17/66)	56.1(37/66)	43.9(29/66)	3.0(2/66)	40.2(239/594)

*:Learned position and speed commands



(a) Original model (b) Self-supervised learning model 1 (c) Self-supervised learning model 2

FIGURE 14. Comparison of the reproducibility of task completion time using the proposed method. In (b) and (c), the results of the model before fine tuning are plotted.

the original model. With the increase in autonomous motion data, both spatial interpolation and dynamics learning progressed, allowing the robot to complete tasks at task completion times and positions that were unsuccessful in the original model. The major improvement in the spatial direction was the achievement of a task success rate of 56.1 % in the upper, where the original model had a low success rate. In the temporal direction, the robot could rotate its wrist joint with an appropriate force and angular velocity, which reduced the number of failures when the pancake was dropped during the lifting phase.

In the experimental results, there are only a few conditions in which the original model has a higher success rate than the self-supervised learning model 1. This is thought to be because the number of trials per condition was 6 times, which caused a variance in the success rate. However, the success rate for the overall speed is clearly lower for the faster speeds. This indicates that fast operation is more complex than slow operation. Future training with increased amounts of appropriate autonomous data will enable high speed operation.

For position commands, there was no difference in success rate between positions demonstrated by humans and positions not demonstrated by humans. However, for the task completion time command, the success rate was lower for the speed that was not demonstrated by a human compared to the speed demonstrated by a human. These results indicate that it is difficult to learn the dynamics of the change in speed from autonomous motion data, and it is necessary to collect more autonomous motion data at various speeds for learning.

The reproducibility of the task completion time of the self-supervised learning model 1 is illustrated in Fig. 14 (b). Here, as 2) in Section VI-B, only the results of successful trials are employed to calculate the mean and variance. Whereas the error of the original model was 0.65 s on average and 1.04 in variance, the error of the self-supervised learning model 1 was 0.2 s on average and 0.72 in variance, indicating that the reproducibility of task completion time was greatly improved. In particular, for the task completion time commands of 5, 6 s and 10, 11 s, which had large errors in the original model, the plot points are closer to the commanded values and improvement is remarkable. Although the original model tended to operate slowly in response to commands, this issue has been resolved.

D. SELF-SUPERVISED LEARNING(2ND TIME)

1) DESCRIPTION

To test the case of collecting more autonomous motion data, an experiment was conducted using the data collected in Section VI-C. For each of the 180 patterns that were successful at least once in Section VI-C, we obtained additional data so that there were three successful data. The total number of autonomous action data was 540. Among these, 180 data

TABLE 4. Task success rate of the self-supervised learning model 1.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	0.0(0/6)	50.0(3/6)	**100(6/6)**	**83.3(5/6)**	50.0(3/6)	0.0(0/6)	**100(6/6)**	**100(6/6)**	66.7(4/6)	61.1(33/54)
4.00	∗**100(6/6)**	∗66.7(4/6)	∗**100(6/6)**	∗**100(6/6)**	**100(6/6)**	33.3(2/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**88.9(48/54)**
5.00	**100(6/6)**	66.7(4/6)	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	16.7(1/6)	**100(6/6)**	**83.3(5/6)**	0.0(0/6)	72.2(39/54)
6.00	50.0(3/6)	16.7(1/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	33.3(2/6)	0.0(0/6)	66.7(36/54)
7.00	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	66.7(4/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	16.7(1/6)	**87.0(47/54)**
8.00	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**83.3(5/6)**	**96.3(52/54)**
9.00	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	33.3(2/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**92.6(50/54)**
10.0	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	50.0(3/6)	**100(6/6)**	**100(6/6)**	**94.4(51/54)**
11.0	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	66.7(4/6)	50.0(3/6)	0.0(0/6)	**100(6/6)**	**83.3(5/6)**	77.8(42/54)
12.0	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	33.3(2/6)	**92.6(50/54)**
13.0	0.0(0/6)	**83.3(5/6)**	50.0(3/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	50.0(3/6)	**100(6/6)**	33.3(2/6)	68.5(37/54)
Position total	77.3(51/66)	**80.3(53/66)**	**93.9(62/66)**	**98.5(65/66)**	**81.8(54/66)**	72.7(48/66)	**81.8(54/66)**	**92.4(61/66)**	56.1(37/66)	**81.6(485/594)**

∗:Learned position and speed commands

TABLE 5. Task success rate of the self-supervised learning model 2.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	66.7(4/6)	**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**83.3(5/6)**	33.3(2/6)	**83.3(5/6)**	50.0(3/6)	**100(6/6)**	77.8(42/54)
4.00	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	**83.3(5/6)**	66.7(4/6)	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	**92.6(50/54)**
5.00	33.3(2/6)	**100(6/6)**	0.0(0/6)	66.7(4/6)	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**83.3(5/6)**	74.1(40/54)
6.00	66.7(4/6)	**83.3(5/6)**	66.7(4/6)	**100(6/6)**	50.0(3/6)	**100(6/6)**	16.7(1/6)	**83.3(5/6)**	16.7(1/6)	64.8(35/54)
7.00	66.7(4/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	**94.4(51/54)**
8.00	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	∗**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	33.3(2/6)	66.7(4/6)	**87.0(47/54)**
9.00	**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	66.7(4/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**94.4(51/54)**
10.0	**100(6/6)**	**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	**83.3(5/6)**	66.7(4/6)	**100(6/6)**	**100(6/6)**	**92.6(50/54)**
11.0	**100(6/6)**	**100(6/6)**	33.3(2/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**83.3(5/6)**	**100(6/6)**	**100(6/6)**	**90.7(49/54)**
12.0	∗**83.3(5/6)**	∗**100(6/6)**	∗**83.3(5/6)**	∗**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	66.7(4/6)	**92.6(50/54)**
13.0	33.3(2/6)	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	**100(6/6)**	50.0(3/6)	**100(6/6)**	50.0(3/6)	**81.5(44/54)**
Position total	77.3(51/66)	**97.0(64/66)**	77.3(51/66)	**97.0(64/66)**	**86.4(57/66)**	**89.4(59/66)**	78.8(52/66)	**87.9(58/66)**	**80.3(53/66)**	**85.7(509/594)**

∗:Learned position and speed commands

and 360 data were used as training data and validation data, respectively. Thus, the total number of data for training was 404, and that for validation was 484. The total number of trials was 594, the same as in Sections VI-B and VI-C. The number of training times of the NN was determined considering the validation loss, and 25000 times was used. This model is referred to as the self-supervised learning model 2. The loss graph for the self-supervised learning model 2 is illustrated in Fig. 13 (c).

2) RESULT

The task success rates for the self-supervised learning model 2 are presented in Table 5. The overall task success rate was 85.7%, which is a further improvement from the self-supervised learning model 1. The position commands bottom, lower right, lower and lower left were less successful than the other position commands, and there were many failures to put the turner on top of the pancake. As anticipated in 2) of Section VI-C, the increase in autonomous operation data improved the success rate at high speeds. This result indicates that learning the dynamics changing with speed progressed, and NN generated appropriate behaviors.

The reproducibility of the task completion time of the self-supervised learning model 2 is illustrated in Fig. 14 (c). Here, as 2) in Section VI-B, only the results of successful trials are used to calculate the mean and variance. The self-supervised learning model 2 exhibits very high reproducibility, with a mean error of 0.01 s and variance of 0.43 for the task completion time. In particular, the task completion time commands of 6, 10, and 11 s, for which the variance was large in the self-supervised learning model 1, indicated very small variance. In the self-supervised learning models 1 and 2, the number of operation data used to calculate the variance of 6, 10, and 11 s was 36, 51, 42 and 36, 50, 49, indicating that the variability did not decrease with the increase in the number of successful data. These results indicate that both temporal and spatial learning can be advanced using the proposed method, and variable speed nonprehensile manipulation can be performed considering the dynamics of the environment and object.

E. EVALUATION OF GENERALIZATION PERFORMANCE FOR UNTRAINED OBJECTS

1) DESCRIPTION

Using the self-supervised learning model 2, we evaluated the generalization performance for untrained objects. The untrained objects used in this study were the fried egg and the hamburger-steak shown in the Fig. 15, which weighed 42 g and 89 g, respectively. Three trials each were conducted under the same conditions of position and task completion

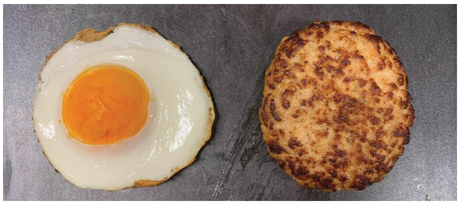


FIGURE 15. Unlearned objects. left;fried egg,right;hamburger-steak.

TABLE 6. Task success rate for untrained objects.

Object	Self-supervised learning model 2	After self-learning
Fried egg	72.1(214/297)	78.8(234/297)
Hamburger steak	73.1(217/297)	73.4(218/297)

time as in Section VI-B. Therefore, 297 trials (9 [position] × 11 [completion time] × 3 [trial]) were conducted for each object. Furthermore, we examined whether fine-tuning the self-supervised learning model 2 with the autonomous action data performed on each object would improve the success rate of the task by learning the physical properties of untrained objects. In this fine-tuning, only the motion data of a specific object was used, not the pancake data.

2) RESULT

As shown on the left side of Table 6, the task success rates for self-supervised learning model 2 were 72.1% for the fried egg and 73.1% for the hamburger-steak. Because the mass, friction of the fried egg or the hamburger-steak are very different from those of a pancake, it is difficult for them to succeed in the task. However, the self-supervised learning model 2, which had been sufficiently trained in nonprehensile manipulation, could successfully perform the task with a high success rate of more than 70% for unlearned objects.

Furthermore, the task success rates of the NN after fine-tuning with the fried egg and hamburger-steak autonomous behavior data is also shown on the right side of Table 6. The numbers of times the NNs were trained was 8000 for the fried egg task and 8500 for the hamburger-steak task, and the loss graphs are shown in Figs. 13 (d), (e). The success rate of the self-learning model with the fried egg was 78.8%, and the success rate of the self-learning model with the hamburger-steak was 73.4%, both of which were slightly better than the results before self-learning. However, the success rate was below 80%, and the improvement in task success rate was less significant when compared to the self-learning of pancakes. Therefore, under the conditions of this study, it was possible to improve the task success rate of nonprehensile manipulation by using the proposed method, though the growth rate was shown to decrease as learning progressed.

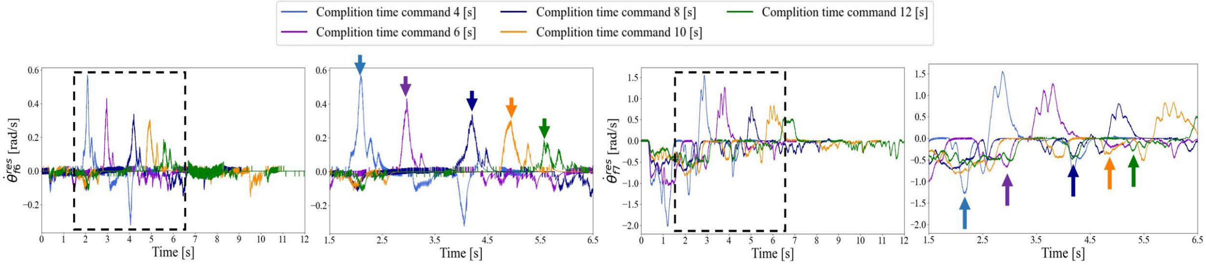
F. DISCUSSION

Fig. 16 illustrates the angular velocity response values of θ_6, θ_7 when the task is performed at multiple speeds using the self-supervised learning model 2. We performed the task in the posture illustrated in Fig. 17. From Fig. 16 (a), it can be confirmed that the wrist joints of θ_6 are rotated rapidly as the task completion time becomes shorter. θ_6 corresponds to the snap of the wrist that moves the turner horizontally, and high-speed rotation of the wrist prevents the pancake from flying away owing to centrifugal force during the transporting phase. It was confirmed that the NN learns the inertia caused by the fast transportation motion. In the angular velocity response values for θ_7 illustrated in Fig. 16 (c), θ_7 corresponds to the snap of the wrist that moves the turner up and down. Fig. 16 (c) illustrates that the angular velocity in the lifting phase changes according to the task completion time, and that the NN generates appropriate motions to prevent the pancake from slipping off and being thrown away in the lifting phase.

From the above data, it is confirmed that the proposed method can learn the environment, pancake, and dynamics between them from the autonomous operation data. Moreover, the results of 2) in Section VI-C indicate that providing various conditions in the spatio-temporal direction of interpolation could learn efficient self-learning. Therefore, we speculate that it is important to train NNs considering both time and space to efficiently learn the dynamic behavior of robots.

The experimental results indicated that the self-supervised learning model 2 was able to reproduce the speed with very high accuracy, with an error of 0.01 s on average and a variance of 0.43. However, the error of 48 datasets collected by humans at 4, 8, and 12 s had a mean of 0.17 s and variance of 0.46. Hence, self-supervised learning model 2 can perform the task with higher accuracy than humans. In conventional imitation learning, all demonstrations have to be taught at approximately the same speed of movement to stabilize learning. In fact, in our conventional method [19], a human demonstrated the task while listening to the sound of a metronome, so that the task execution time would be the same; however, this was very laborious, and it was also impossible to reproduce demonstrations with the same motion speeds. On the contrary, using the proposed method, accurate speed labels can be generated later, and motion speed can be adjusted from the autonomous motion data obtained from the trained NN. Hence, humans are freed from collecting data at the exact rate, and the model itself can be fitted to the exact speed through self-learning.

The proposed method can learn from all the time-series data of successful actions, whereas reinforcement learning learns from rewards determined only by the final states. Therefore, the proposed method is more sample efficient than reinforcement learning. Moreover, because imitation learning limits the range of action by human demonstration, the generated actions are relatively safe and can be subtly adjusted with similar autonomous action data. Reinforcement learning struggles with setting up a reward function that corresponds to task-related parameters, such as arbitrary motion speed, and is prone to overlearning for a specific motion speed.



(a) Angular velocity response value of θ_6 (b) Enlarged view of a part of (a) (c) Angular velocity response value of θ_7 (d) Enlarged view of a part of (c)

FIGURE 16. Comparison of angular velocity response values when executing a task at each speed. The dashed lines in (a) and (c) represent the duration of the transporting and lifting phases, respectively. (b) and (d) focus on the time of the dashed boxes in (a) and (c), respectively.

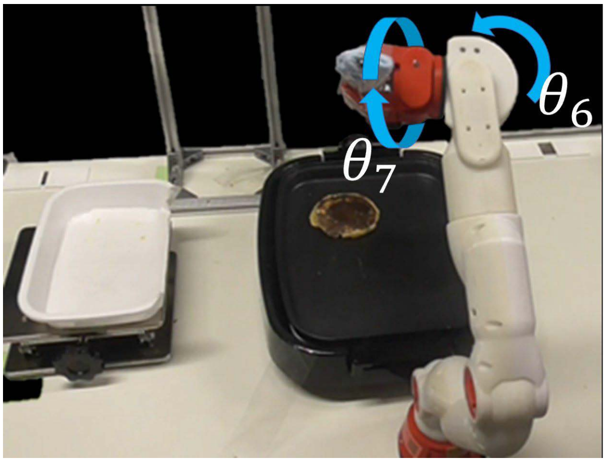


FIGURE 17. State of each joint of θ_6 and θ_7 during task execution.

In the experimental results of Section VI-E, self-learning for unlearned objects did not significantly improve the success rate. We consider that this result is caused by the fact that the self-supervised learning model 2 has already overlearned for the pancake. In particular, in the structure of the NN used in this study, information about the physical properties of the manipulated object was not input. This means that the task is being executed with information such as shape and size unknown. Therefore, when a command with the same position and task completion time was input, the same motion with very little variation was generated without considering the size and shape of the object. Recently, methods using raw images have been studied [51], [52]. These studies add raw images to the NN input and learn to generate appropriate motions in response to changes in the position, and shape of the object. Therefore, we consider that the proposed method will be able to learn motions that consider various sizes, and shape by adding raw image data to the learning process. It may improve the success rate of 80% or more for new objects. However, the input of images makes learning more difficult. In this case, curriculum learning [53], [54], which improves the prediction accuracy of the model by initially learning with data under simple conditions and gradually making the model more difficult, could be introduced to solve the problem. The combination of these methods and the proposed method is expected to be effective for untrained objects, because it considers the shape and size of the manipulated object and avoids overlearning for a specific object.

As mentioned in the introduction, the field of machine learning has focused on geometric problems and has insufficiently considered temporal information such as speed. However, because several physical phenomena have been formulated and understood using differential equations, time information is an important factor in dynamic object manipulation. In the proposed method, the dynamics to be considered in nonprehensile manipulation can be learned by self-learning using a time index. In addition, the proposed method can be widely employed for several tasks using the task completion time as the time index. Therefore, our method is highly compatible with conventional self-supervised learning focusing on geometric problems, and when combined with conventional methods, it is expected to enable more complex and dynamic behaviors, depending on the environment and situation. In recent years, research has been conducted to understand concepts such as color and shape in language and images from the physicality of the robot using self-supervised learning [55]. Further development of the proposed method may lead to the understanding of complex and abstract concepts correlated with speed, such as ‘‘fast or slow,’’ ‘‘strong or weak,’’ and ‘‘heavy or light,’’ through the dynamic body movements of the robot.

VII. CONCLUSION

We proposed a self-supervised learning method that considers speed. In the proposed method, the NN was fine-tuned using only successful actions among the autonomous action data generated by the trained NN. Using the proposed method was able to improve the success rate in both spatio-temporal conditions. Although nonprehensile manipulation requires considering the dynamics between the environment and object, and it is difficult to perform the task at multiple speeds, the proposed method was feasible with as few as 24 pieces of supervised data. Furthermore, the proposed method was able to complete the task in a more accurate time than the given training data. After self-learning, the NN appropriately altered its force and trajectory according to the task completion time, confirming that it is capable of learning the dynamics between itself, the environment, manipulated objects, and these from automatically generated autonomous motion data.

In the past, self-supervised learning focused on spatial information and did not fully utilize temporal information. The proposed method is compatible with these conventional methods, expanding the possibilities of self-supervised learning and contributing greatly to the understanding of dynamic phenomena in robotic tasks. However, the spatial information in this study is limited to the center position of the pancake at the beginning of the task, and the shape and size of the object were not considered. Therefore, our future work is to integrate the proposed method with a real-time image-based motion generation method [51] and a method that considers the shape and size of multiple objects [52] to expand the tasks that can be performed by the robot in space and time.

APPENDIX

The experimental results for the 30 g and 90 g individual pancakes are presented Tables 7–12. As shown in the Tables 7–12, task success rates for each model did not differ significantly with pancake mass.

TABLE 7. Task success rate of the original model for pancake of 30g.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	0.0(0/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	33.3(9/27)
4.00	33.3(1/3)	**100(3/3)**	**100(3/3)**	33.3(1/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	0.0(0/3)	63.0(17/27)
5.00	0.0(0/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	33.3(1/3)	33.3(1/3)	0.0(0/3)	51.9(14/27)
6.00	0.0(0/3)	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	0.0(0/3)	0.0(0/3)	33.3(9/27)
7.00	33.3(1/3)	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	0.0(0/3)	66.7(2/3)	0.0(0/3)	22.2(6/27)
8.00	66.7(2/3)	0.0(0/3)	66.7(2/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	33.3(1/3)	33.3(1/3)	0.0(0/3)	33.3(9/27)
9.00	66.7(2/3)	0.0(0/3)	33.3(1/3)	**100(3/3)**	0.0(0/3)	66.7(2/3)	0.0(0/3)	66.7(2/3)	0.0(0/3)	37.0(10/27)
10.0	0.0(0/3)	0.0(0/3)	0.0(0/3)	**100(3/3)**	66.7(2/3)	33.3(1/3)	33.3(1/3)	**100(3/3)**	0.0(0/3)	37.0(10/27)
11.0	0.0(0/3)	**100(3/3)**	66.7(2/3)	66.7(2/3)	66.7(2/3)	0.0(0/3)	66.7(2/3)	33.3(1/3)	33.3(1/3)	48.1(13/27)
12.0	0.0(0/3)	**100(3/3)**	0.0(0/3)	33.3(1/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	66.7(2/3)	0.0(0/3)	44.4(12/27)
13.0	0.0(0/3)	**100(3/3)**	0.0(0/3)	66.7(2/3)	0.0(0/3)	0.0(0/3)	0.0(0/3)	66.7(2/3)	0.0(0/3)	25.9(7/27)
Position total	18.2(6/33)	54.5(18/33)	24.2(8/33)	81.8(27/33)	39.4(13/33)	36.4(12/33)	51.5(17/33)	42.4(14/33)	3.0(1/33)	39.1(116/297)

TABLE 8. Task success rate of the original model for pancake of 90g.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	0.0(0/3)	**0.0(0/3)**	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	22.2(6/27)
4.00	0.0(0/3)	**100(3/3)**	**0.0(0/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	0.0(0/3)	55.6(15/27)
5.00	**100(3/3)**	**0.0(0/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	**33.3(1/3)**	**100(3/3)**	0.0(0/3)	0.0(0/3)	48.1(13/27)
6.00	0.0(0/3)	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	**0.0(0/3)**	**33.3(1/3)**	33.3(1/3)	0.0(0/3)	18.5(5/27)
7.00	**100(3/3)**	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	0.0(0/3)	66.7(2/3)	0.0(0/3)	29.6(8/27)
8.00	**100(3/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	0.0(0/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	44.4(12/27)
9.00	33.3(1/3)	0.0(0/3)	**100(3/3)**	**100(3/3)**	0.0(0/3)	33.3(1/3)	0.0(0/3)	**100(3/3)**	0.0(0/3)	40.7(11/27)
10.0	33.3(1/3)	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	0.0(0/3)	59.3(16/27)
11.0	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	**100(3/3)**	66.7(2/3)	33.3(1/3)	66.7(18/27)
12.0	0.0(0/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	**66.7(2/3)**	0.0(0/3)	**33.3(1/3)**	66.7(2/3)	0.0(0/3)	40.7(11/27)
13.0	0.0(0/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	0.0(0/3)	0.0(0/3)	0.0(0/3)	66.7(2/3)	0.0(0/3)	29.6(8/27)
Position total	33.3(11/33)	36.4(12/33)	36.4(12/33)	100(33/33)	42.4(14/33)	15.2(5/33)	60.6(20/33)	45.5(15/33)	3.0(1/33)	41.4(123/297)

TABLE 9. Task success rate of the self-supervised learning model 1 for pancake of 30g.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	0.0(0/3)	**33.3(1/3)**	**100(3/3)**	**66.7(2/3)**	**33.3(1/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	55.6(15/27)
4.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	96.3(26/27)
5.00	**100(3/3)**	**66.7(2/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**33.3(1/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	74.1(20/27)
6.00	33.3(1/3)	**33.3(1/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	0.0(0/3)	70.4(19/27)
7.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	85.2(23/27)
8.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	100(27/27)
9.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	33.3(1/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	92.6(25/27)
10.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	96.3(26/27)
11.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	33.3(1/3)	**0.0(0/3)**	**100(3/3)**	**100(3/3)**	77.8(21/27)
12.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	96.3(26/27)
13.0	0.0(0/3)	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**33.3(1/3)**	**100(3/3)**	**33.3(1/3)**	70.4(19/27)
Position total	75.8(25/33)	84.8(28/33)	93.9(31/33)	97.0(32/33)	81.8(27/33)	75.8(25/33)	81.8(27/33)	97.0(32/33)	60.6(20/33)	83.2(247/297)

TABLE 10. Task success rate of the self-supervised learning model 1 for pancake of 90g.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	0.0(0/3)	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	0.0(0/3)	**100(3/3)**	**100(3/3)**	66.7(2/3)	66.7(18/27)
4.00	**100(3/3)**	**33.3(1/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**0.0(0/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	81.5(22/27)
5.00	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**0.0(0/3)**	**100(3/3)**	66.7(2/3)	0.0(0/3)	70.4(19/27)
6.00	66.7(2/3)	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	0.0(0/3)	63.0(17/27)
7.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	33.3(1/3)	88.9(24/27)
8.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	92.6(25/27)
9.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	33.3(1/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	92.6(25/27)
10.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	33.3(1/3)	**100(3/3)**	**100(3/3)**	92.6(25/27)
11.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	66.7(2/3)	0.0(0/3)	**100(3/3)**	66.7(2/3)	77.8(21/27)
12.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	88.9(24/27)
13.0	0.0(0/3)	**66.7(2/3)**	33.3(1/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	33.3(1/3)	66.7(18/27)
Position total	78.8(26/33)	75.8(25/33)	93.9(31/33)	100(33/33)	81.8(27/33)	69.7(23/33)	81.8(27/33)	87.9(29/33)	51.5(17/33)	80.1(238/297)

TABLE 11. Task success rate of the self-supervised learning model 2 for pancake of 30g.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	66.7(2/3)	**100(3/3)**	66.7(2/3)	**100(3/3)**	85.2(23/27)
4.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	92.6(25/27)
5.00	66.7(2/3)	**100(3/3)**	0.0(0/3)	**33.3(1/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	70.4(19/27)
6.00	33.3(1/3)	66.7(2/3)	33.3(1/3)	**100(3/3)**	**33.3(1/3)**	**100(3/3)**	**33.3(1/3)**	66.7(2/3)	**33.3(1/3)**	55.6(15/27)
7.00	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	96.3(26/27)
8.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	66.7(2/3)	85.2(23/27)
9.00	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	**33.3(1/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	88.9(24/27)
10.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	96.3(26/27)
11.0	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	92.6(25/27)
12.0	66.7(2/3)	**100(3/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	88.9(24/27)
13.0	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	**100(3/3)**	66.7(2/3)	88.9(24/27)
Position total	78.8(26/33)	93.9(31/33)	78.8(26/33)	93.9(31/33)	78.8(26/33)	93.9(31/33)	84.8(28/33)	84.8(28/33)	81.8(27/33)	85.5(254/297)

TABLE 12. Task success rate of the self-supervised learning model 2 for pancake of 90g.

Task completion time command [s]	Position command									Speed total
	Learned				Unlearned					
	Lower left	Upper left	Lower right	Upper right	Center	Left	Lower	Right	Upper	
3.00	66.7(2/3)	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	0.0(0/3)	**66.7(2/3)**	**33.3(1/3)**	**100(3/3)**	70.4(19/27)
4.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	92.6(25/27)
5.00	0.0(0/3)	**100(3/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	77.8(21/27)
6.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**0.0(0/3)**	**100(3/3)**	**0.0(0/3)**	74.1(20/27)
7.00	66.7(2/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	92.6(25/27)
8.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	66.7(2/3)	88.9(24/27)
9.00	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	100(27/27)
10.0	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	**100(3/3)**	**100(3/3)**	**66.7(2/3)**	66.7(2/3)	**100(3/3)**	**100(3/3)**	88.9(24/27)
11.0	**100(3/3)**	**100(3/3)**	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	88.9(24/27)
12.0	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	66.7(2/3)	96.3(26/27)
13.0	0.0(0/3)	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	**100(3/3)**	33.3(1/3)	**100(3/3)**	**33.3(1/3)**	74.1(20/27)
Position total	75.8(25/33)	100(33/33)	75.8(25/33)	100(33/33)	93.9(31/33)	84.8(28/33)	72.7(24/33)	90.9(30/33)	78.8(26/33)	85.9(255/297)

REFERENCES

[1] D. Seita, A. Ganapathi, R. Hoque, M. Hwang, E. Cen, A. K. Tanwani, A. Balakrishna, B. Thananjeyan, J. Ichnowski, N. Jamali, K. Yamane, S. Iba, J. Canny, and K. Goldberg, ''Deep imitation learning of sequential fabric smoothing from an algorithmic supervisor,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 9651–9658.

[2] P. Ruppel and J. Zhang, ''Learning object manipulation with dexterous hand-arm systems from human demonstration,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5417–5424.

[3] N. Saito, D. Wang, T. Ogata, H. Mori, and S. Sugano, ''Wiping 3D-objects using deep learning model based on image/force/joint information,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10152–10157.

[4] W. C. Agboh and M. R. Dogar, ''Pushing fast and slow: Task-adaptive planning for non-prehensile manipulation under uncertainty,'' Jan. 2019, *arXiv:1805.03005*.

[5] K. M. Lynch and M. T. Mason, ''Dynamic nonprehensile manipulation: Controllability, planning, and experiments,'' *Int. J. Robot. Res.*, vol. 18, no. 1, pp. 64–92, Jan. 1999.

[6] F. Ruggiero, V. Lippiello, and B. Siciliano, ''Nonprehensile dynamic manipulation: A survey,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1711–1718, Jul. 2018.

[7] G. Bätz, A. Yaqub, H. Wu, K. Kühnlenz, D. Wollherr, and M. Buss, ''Dynamic manipulation: Nonprehensile ball catching,'' in *Proc. 18th Medit. Conf. Control Autom. (MED)*, Jun. 2010, pp. 365–370.

[8] M. Erdmann, ''An exploration of nonprehensile two-palm manipulation,'' *Int. J. Robot. Res.*, vol. 17, no. 5, pp. 485–503, May 1998.

[9] B. Aceituno, A. Rodriguez, S. Tulsiani, A. Gupta, and M. Mukadam, ''A differentiable recipe for learning visual non-prehensile planar manipulation,'' Nov. 2021, *arXiv:2111.05318*.

[10] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, ''Recent advances in robot learning from demonstration,'' *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 3, pp. 297–330, Dec. 2020.

[11] B. Fang, S. Jia, D. Guo, M. Xu, S. Wen, and F. Sun, "Survey of imitation learning for robotic manipulation," *Int. J. Intell. Robot. Appl.*, vol. 3, no. 4, pp. 362–369, Sep. 2019.

[12] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 3758–3765.

[13] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, "One-shot visual imitation learning via meta-learning," Sep. 2017, *arXiv:1709.04905*.

[14] L. Rozo, P. Jimenez, and C. Torras, "A robot learning from demonstration framework to perform force-based manipulation tasks," *J. Intell. Service Robot., Special Issue Artif. Intell. Techn. Robot., Sens., Represent. Action*, vol. 6, pp. 33–51, Jan. 2013.

[15] T. Tosun, E. Mitchell, B. Eisner, J. Huh, B. Lee, D. Lee, V. Isler, H. S. Seung, and D. Lee, "Pixels to plans: Learning non-prehensile manipulation by imitating a planner," Apr. 2019, *arXiv:1904.03260*.

[16] T. Adachi, K. Fujimoto, S. Sakaino, and T. Tsuji, "Imitation learning for object manipulation based on position/force information using bilateral control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3648–3653.

[17] A. Sasagawa, K. Fujimoto, S. Sakaino, and T. Tsuji, "Imitation learning based on bilateral control for human–robot cooperation," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6169–6176, Oct. 2020.

[18] S. Sakaino, K. Fujimoto, Y. Saigusa, and T. Tsuji, "Imitation learning for variable speed contact motion for operation up to control bandwidth," *IEEE Open J. Ind. Electron. Soc.*, vol. 3, pp. 116–127, 2022.

[19] Y. Saigusa, A. Sasagawa, S. Sakaino, and T. Tsuji, "Imitation learning for variable speed motion generation over multiple actions," in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2021, pp. 1–6.

[20] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[21] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," Oct. 2020, *arXiv:2011.00362*.

[22] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8943–8950.

[23] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," Sep. 2015, *arXiv:1509.06825*.

[24] L. Berscheid, P. Meißner, and T. Kröger, "Self-supervised learning for precise pick-and-place without object model," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4828–4835, Jul. 2020.

[25] K. Takahashi, W. Ko, A. Ummadisingu, and S.-I. Maeda, "Uncertainty-aware self-supervised target-mass grasping of granular foods," May 2021, *arXiv:2105.12946*.

[26] Y. Huang, J. Wilches, and Y. Sun, "Robot gaining accurate pouring skills through self-supervised learning and generalization," Nov. 2020, *arXiv:2011.10150*.

[27] F. Ruggiero, A. Petit, D. Serra, A. C. Satici, J. Cacace, A. Donaire, F. Ficuciello, L. R. Buonocore, G. A. Fontanelli, V. Lippiello, L. Villani, and B. Siciliano, "Nonprehensile manipulation of deformable objects: Achievements and perspectives from the robotic dynamic manipulation project," *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, pp. 83–92, Sep. 2018.

[28] H. Miyashita, T. Yamawaki, and M. Yashima, "Robotic juggling by iterative learning control using optimization," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Feb. 2009, pp. 1427–1432.

[29] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," May 2018, *arXiv:1805.00909*.

[30] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.*, vol. 37, nos. 4–5, pp. 421–436, 2017.

[31] W. Yuan, J. A. Stork, D. Kragic, M. Y. Wang, and K. Hang, "Rearrangement with nonprehensile manipulation using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 270–277.

[32] C. Finn, X. Yu Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," May 2016, *arXiv:1509.06113*.

[33] T. Tanaka, T. Kaneko, M. Sekine, V. Tangkaratt, and M. Sugiyama, "Simultaneous planning for item picking and placing by deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 9705–9711.

[34] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," *Int. J. Humanoid Robot.*, vol. 5, no. 2, pp. 183–202, Jun. 2008.

[35] J. Silverio, Y. Huang, L. Rozo, S. Calinon, and D. G. Caldwell, "Probabilistic learning of torque controllers from kinematic and force constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 6522–6599.

[36] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," Oct. 2017, *arXiv:1710.04615*.

[37] P.-C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata, "Repeatable folding task by humanoid robot worker using deep learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 397–403, Apr. 2016.

[38] K. Kawaharazuka, K. Tsuzuki, M. Onitsuka, Y. Asano, K. Okada, K. Kawasaki, and M. Inaba, "Object recognition, dynamic contact simulation, detection, and control of the flexible musculoskeletal hand using a recurrent neural network with parametric bias," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4580–4587, Jul. 2020.

[39] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, "A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1193–1203, Jun. 2018.

[40] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 763–768.

[41] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2016, pp. 69–84.

[42] V. Guizilini, R. Ambruş, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2485–2493.

[43] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6D object pose estimation for robot manipulation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3665–3671.

[44] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-contrastive networks: Self-supervised learning from video," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1134–1141.

[45] K. Ohnishi, M. Shibata, and T. Murakami, "Motion control for advanced mechatronics," *IEEE/ASME Trans. Mechatronics*, vol. 1, no. 1, pp. 56–67, Mar. 1996.

[46] T. Murakami, F. Yu, and K. Ohnishi, "Torque sensorless control in multidegree-of-freedom manipulator," *IEEE Trans. Ind. Electron.*, vol. 40, no. 2, pp. 259–265, Apr. 1993.

[47] S. Sakaino, T. Sato, and K. Ohnishi, "Multi-DOF micro-macro bilateral controller using oblique coordinate control," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 446–454, Aug. 2011.

[48] R. Rahmatizadeh, P. Abolghasemi, A. Behal, and L. Bölöni, "From virtual demonstration to real-world manipulation using LSTM and MDN," Mar. 2016, *arXiv:1603.03833*.

[49] A. Sasagawa, S. Sakaino, and T. Tsuji, "Motion generation using bilateral control-based imitation learning with autoregressive learning," *IEEE Access*, vol. 9, pp. 20508–20520, 2021.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*.

[51] K. Suzuki, M. Kanamura, Y. Suga, H. Mori, and T. Ogata, "In-air knotting of rope using dual-arm robot based on deep learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 6724–6731.

[52] N. Saito, T. Ogata, S. Funabashi, H. Mori, and S. Sugano, "How to select and use tools: Active perception of target objects using multimodal deep learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2517–2524, Apr. 2021.

[53] Y. Bengio, J. Louradour, R. Collobert, and J. Schoukens, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 41–48.

[54] K. Kutsuzawa, S. Sakaino, and T. Tsuji, "Sequence-to-sequence model for trajectory planning of nonprehensile manipulation including contact model," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3606–3613, Oct. 2018.

[55] J. Zhong, T. Ogata, A. Cangelosi, and C. Yang, "Disentanglement in conceptual space during sensorimotor interaction," *Cognit. Comput. Syst.*, vol. 1, no. 4, pp. 103–112, Nov. 2019.

YUKI SAIGUSA received the B.E. degree in electrical and electronic system engineering from Saitama University, Saitama, Japan, in 2020, and the M.E. degree from the Graduate School of Science and Technology, and the degree programs in intelligent and mechanical interaction systems from the University of Tsukuba, Japan. His research interests include motion control, robotics, and machine learning.

SHO SAKAINO (Member, IEEE) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Yokohama, Japan, in 2006, 2008, and 2011, respectively. He was an Assistant Professor at Saitama University, from 2011 to 2019. Since 2019, he has been an Associate Professor at the University of Tsukuba. His research interests include mechatronics, motion control, robotics, and haptics. He received the IEEJ Industry Application Society Distinguished Transaction Paper Award, in 2011 and 2020. He also received the RSJ Advanced Robotics Excellent Paper Award, in 2020.

TOSHIAKI TSUJI (Senior Member, IEEE) received the B.E. degree in system design engineering and the M.E. and Ph.D. degrees in integrated design engineering from Keio University, Yokohama, Japan, in 2001, 2003, and 2006, respectively. He was a Research Associate with the Department of Mechanical Engineering, Tokyo University of Science, from 2006 to 2007. He is currently an Associate Professor with the Department of Electrical and Electronic Systems, Saitama University, Saitama, Japan. His research interests include motion control, haptics, and rehabilitation robots. He received the FANUC FA and Robot Foundation Original Paper Award, in 2007 and 2008, respectively. He also received the RSJ Advanced Robotics Excellent Paper Award and the IEEJ Industry Application Society Distinguished Transaction Paper Award, in 2020.